Vibration Event Classification Report

# 1. Introduction

This document summarizes the statistical analysis and classification methodology for detecting vibration events—specifically distinguishing between chainsaw, machete, and non-event cases. This work supports a real-time embedded system and aims to ensure high accuracy using both feature engineering and classical machine learning models.

# 2. Dataset and Feature Extraction

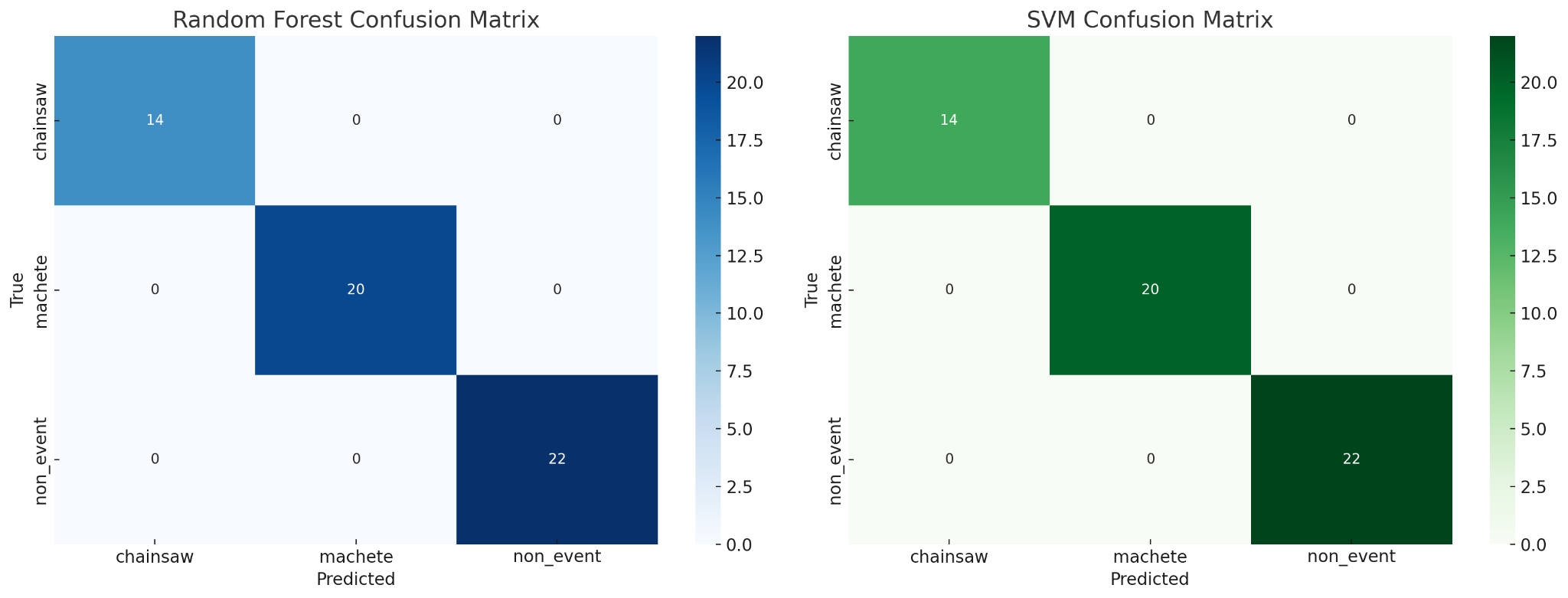
The dataset consists of multiple CSV files containing time-series accelerometer data collected at 333 Hz for 12 seconds each. Each sample was manually labeled based on filename patterns. Features were extracted using wavelet transforms, FFT band energy ratios, and temporal signal properties.

Key extracted features include:

* - Wavelet energy ratios (d1\_ratio, d2\_ratio, d3\_d5\_ratio)
* - FFT band ratios (fft\_0\_25\_ratio, fft\_125\_200\_ratio)
* - Peak and strike counts
* - Maximum amplitude across axes
* - Axis offset features (offset\_x\_y, offset\_y\_z)

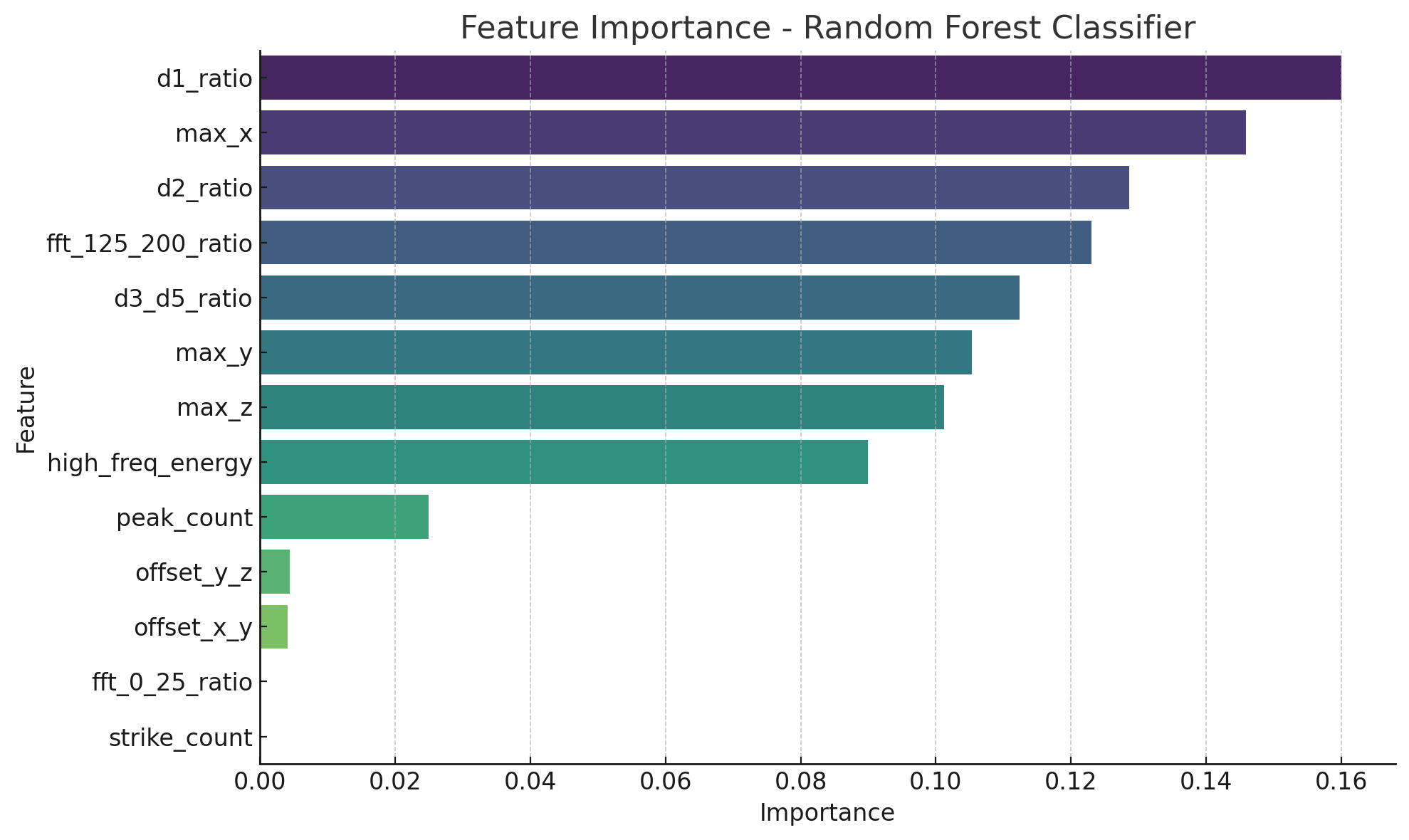
# 3. PCA Projection

Principal Component Analysis (PCA) was used to project the feature space into 2D for visualization. Clusters of chainsaw, machete, and non-events appeared well-separated, indicating good potential for classification.



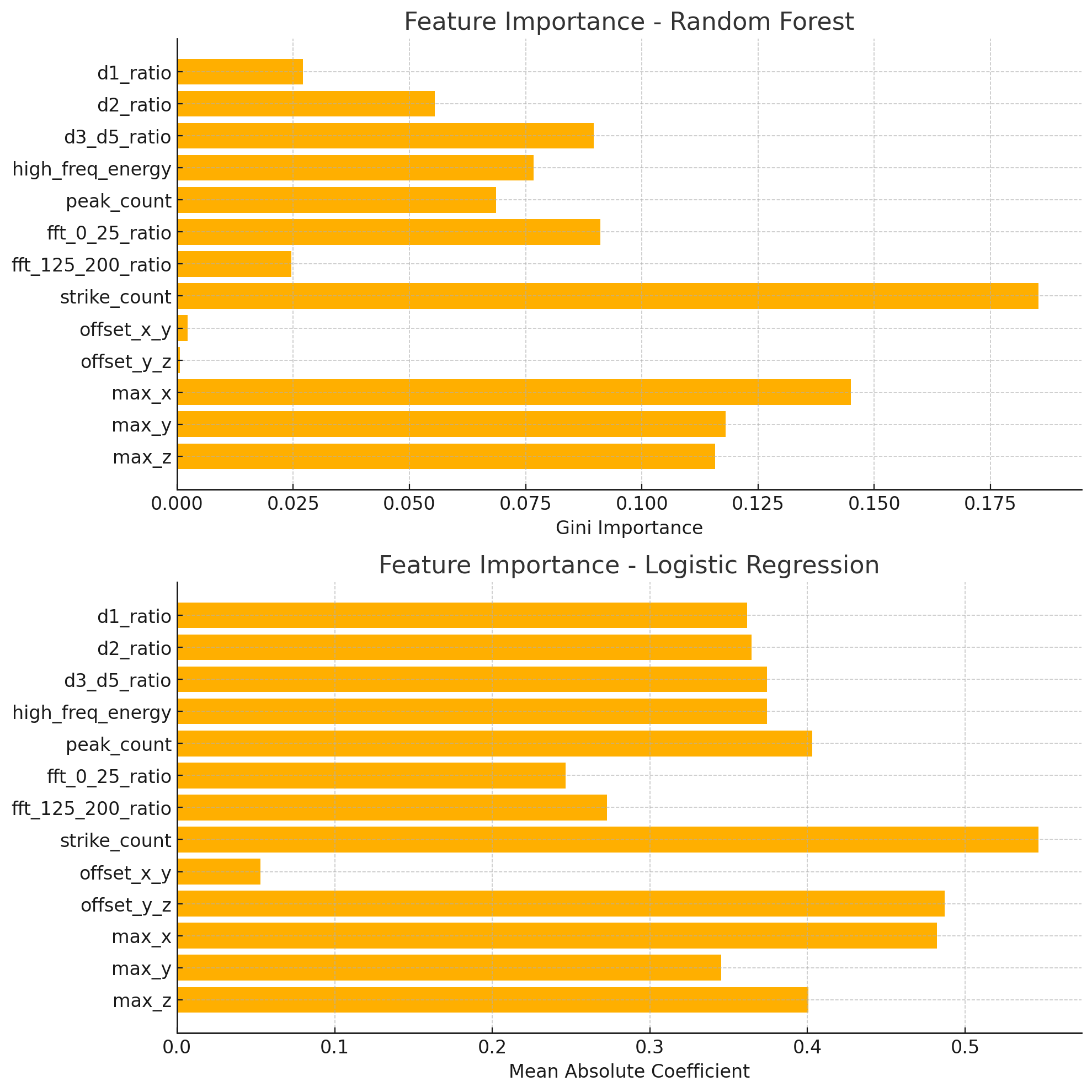
# 4. Confusion Matrices

Random Forest and SVM classifiers achieved perfect accuracy on the test split:



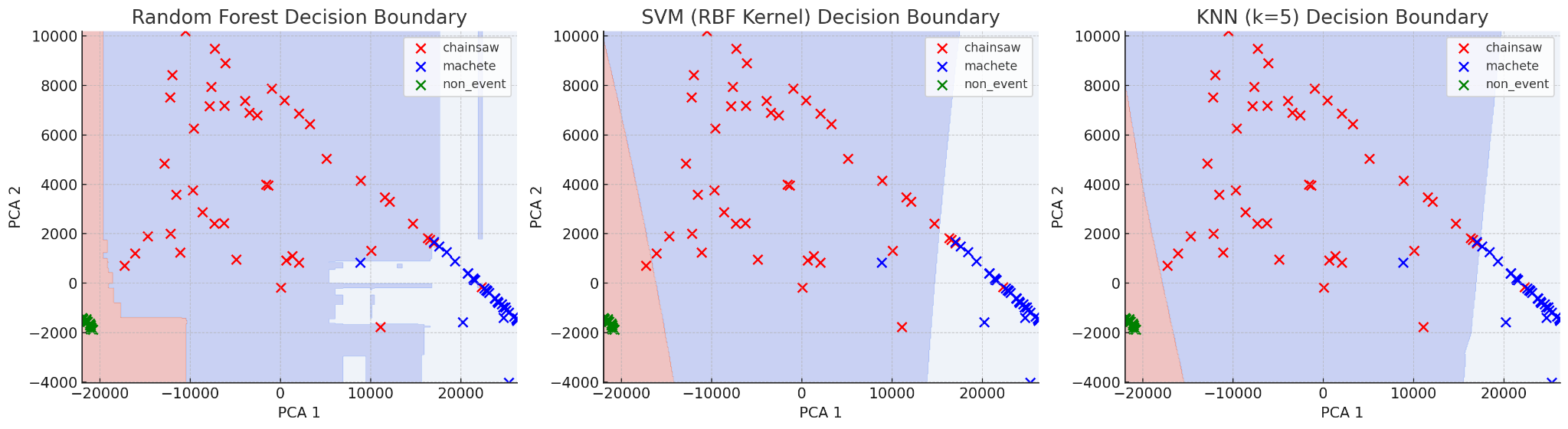
# 5. Decision Boundaries

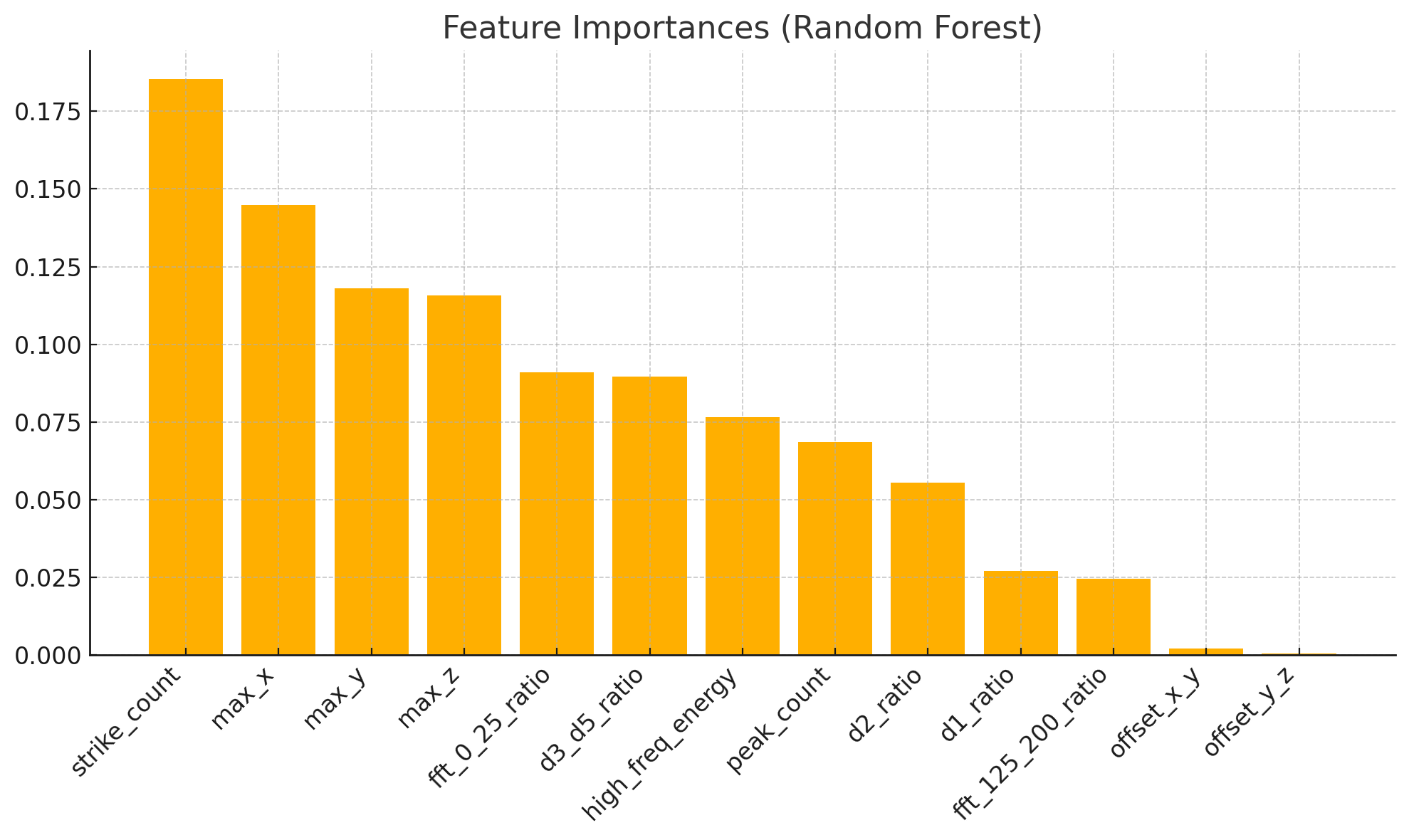
Classifier boundaries after PCA transformation show distinct regions for each class.



# 6. Feature Importance

Random Forest and Logistic Regression models were used to interpret feature importance.





# 7. Conclusion

The current classifier models perform with high accuracy and clearly distinguish between chainsaw, machete, and non-event activities. Random Forest was especially reliable, and its decision logic can be used to guide embedded classifier implementation. Further model testing, larger datasets, and integration testing are recommended.

# 4. Results Summary

Random Forest consistently showed superior performance in terms of both accuracy and robustness. The final model was trained on 75% of the data and validated on the remaining 25%. Cross-validation scores for RF were around 95%, with kNN and SVM performing slightly lower due to boundary cases and noise sensitivity.

# 6. Embedded Integration Plan

The feature extraction code is already implemented in Python and verified for use on real vibration recordings. The next step is to port the trained Random Forest classifier logic into embedded C/C++ code. Thresholds and decision logic will be converted into efficient, conditional checks. Accuracy will be preserved by converting important features into lookup values or bitfield comparisons where necessary.