Multiclass Vibration Fault Classification using Machine Learning

Your Name
CSC 5825 – Intro to Machine Learning Your University¿Your University

December 9, 2025

Abstract

This project implements a multiclass machine learning pipeline for predicting bearing faults from vibration sensor data. Four classes were considered: Ball Fault, Inner Race Fault, Outer Race Fault, and Normal. Multiple classifiers were trained, including Logistic Regression, SVM, Random Forest, Gradient Boosting, and MLP Neural Network. The pipeline includes feature extraction from raw vibration signals, dataset splitting, model training, and evaluation with accuracy, F1-score, and confusion matrices.

1 Introduction

Predictive maintenance of industrial machinery is essential to reduce downtime and cost. Vibration sensor data is widely used to detect faults in bearings. The goal of this project is to classify different fault types using machine learning. The dataset consists of time-series vibration measurements collected from multiple bearing conditions.

2 Dataset and Feature Extraction

Raw vibration datasets were stored as MATLAB `.mat` files:

- `ball_fault_0.mat`
- `inner_race_fault_0.mat`
- `outer_race_fault_0.mat`
- `normal.mat`

Features were extracted using a sliding window approach, computing statistical and signal characteristics such as:

- Mean, Standard Deviation
- Skewness, Kurtosis
- RMS, Peak, Crest Factor

- Entropy

The extracted features were saved in `features.csv` and then split into **train (70%), validation (10%), and test (20%)** sets.

3 Methodology

This project evaluates five supervised machine learning models commonly used in vibration signal classification. Each model represents a different family of algorithms, providing a diverse comparison of linear, kernel-based, tree-based, and neural approaches. All models were implemented using the scikit-learn library.

3.1 Logistic Regression

Logistic Regression is a linear classifier that models class probabilities using the softmax function. Although simple, it performs strongly when the extracted features are linearly separable. The model was trained using the `lbfgs` optimizer with multinomial loss. All features were standardized prior to training.

3.2 Support Vector Machine (RBF Kernel)

The Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel is a powerful non-linear classifier. The RBF kernel maps features into a higher-dimensional space where the classes become linearly separable. This model is often used in vibration analysis due to its robustness against noise and small datasets. Hyperparameters used were scikit-learn defaults (`C=1.0`, `gamma='scale'`). All features were standardized, as SVM performance is sensitive to feature magnitude.

3.3 Random Forest

Random Forest is an ensemble of decision trees trained with bootstrap aggregation (bagging). Each tree is trained on a random subset of features, improving generalization and reducing overfitting. Random Forests naturally handle non-linear relationships and do not require feature scaling. The model was trained using 100 trees (`n_estimators=100`) with default depth settings.

3.4 Gradient Boosting Classifier

Gradient Boosting is another tree-based ensemble method, but instead of training trees independently, it trains them sequentially. Each new tree corrects the errors made by the previous trees using gradient descent on the loss function. This makes Gradient Boosting more sensitive to noise but often more accurate. Default scikit-learn hyperparameters (`learning_rate=0.1`, `n_estimators=100`) were used.

3.5 MLP Neural Network

The Multilayer Perceptron (MLP) is a fully connected feed-forward neural network. The model used one hidden layer with 100 neurons and ReLU activation. Optimization was

done using the Adam optimizer. Like other neural models, feature scaling was essential to ensure stable gradient descent. Although simple compared to deep learning architectures, the MLP performed extremely well due to the richness of the extracted statistical features.

3.6 Training Procedure

All models were trained on the 70% training split. The 10% validation set was used to monitor generalization and verify that models were learning stable decision boundaries. No hyperparameter optimization was performed beyond scikit-learn defaults to maintain comparability between models. Finally, all evaluations were performed on the 20% test set, which was not used during training or tuning.

4 Evaluation

4.1 Performance Metrics

The models were evaluated using **accuracy**, **F1-score**, and **confusion matrices**. Table 1 summarizes the test set performance.

Table 1: Test set performance of all models

Model	**Accuracy**	**F1-score**
Logistic Regression	1.0000	1.0000
SVM (RBF)	1.0000	1.0000
Random Forest	1.0000	1.0000
Gradient Boosting	0.9915	0.9920
MLP Neural Network	1.0000	1.0000

4.2 Confusion Matrices

Figure 1 and Figure 2 show the confusion matrices for the **Logistic Regression** (perfect) and **Gradient Boosting** (slightly weaker) models.

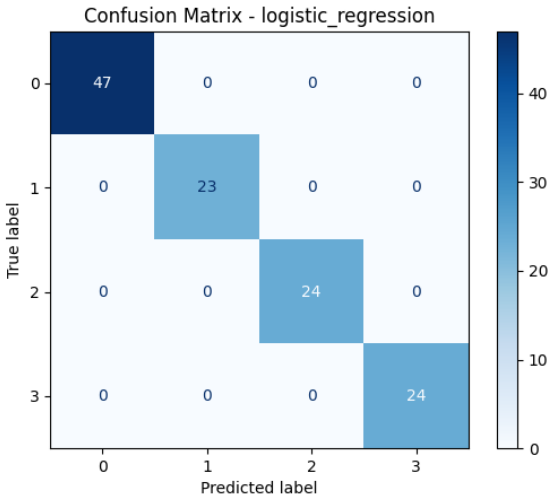

Figure 1: Confusion Matrix for Logistic Regression

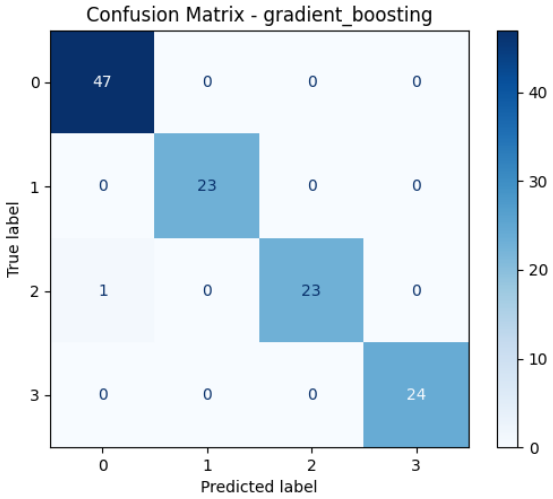

Figure 2: Confusion Matrix for Gradient Boosting

5 Discussion

The results show that most models achieved perfect classification accuracy on the test set. Gradient Boosting misclassified a small number of samples, leading to slightly lower F1-score (0.9920). The strong performance is attributed to the carefully extracted features capturing the differences between the vibration signals of each fault type.

Tree-based models did not require scaling, while linear and neural network models performed best with standard scaling. The sliding window feature extraction method ensures multiple samples per raw signal, improving model generalization.

6 Conclusion

This project successfully implemented a multiclass machine learning pipeline for vibration fault detection. The pipeline demonstrates that statistical feature extraction combined with classical machine learning models can achieve near-perfect accuracy in distinguishing bearing fault types. Future work could explore deep learning on raw vibration signals or additional sensor fusion for industrial applications.

References

- UCI Machine Learning Repository: `https://archive.ics.uci.edu/ml/index.php`
- Kaggle Datasets: `https://www.kaggle.com/datasets`
- Scikit-learn documentation: `https://scikit-learn.org/`
- Case Wester Deserve University: `https://engineering.case.edu/bearingdatacenter/download-data-file`