18-Aug-24

PORTFOLIO

WEEK 3 - DOCUMENTATION

Tran Duc Anh Dang | 103995439
STUDIO CLASS: STUDIO 1-3

Abstract

This week 3 portflio report, explores the application of various machine learning models, including Support Vector Machine (SVM), Random Forest, SGDClassifier, and MLPClassifier, to classify sensor data from meat processing activities. The data was preprocessed, and features were engineered to enhance model performance. Multiple models were trained and evaluated, with the SVM model with hyperparameter tuning selected as the final model due to its simplicity, interpretability, and computational efficiency. The models achieved near-perfect accuracy, though potential overfitting was identified, suggesting the need for further validation. The findings demonstrate the effectiveness of machine learning in accurately classifying complex datasets.

You can find the requirements, documentation and source code file at:
Requirements: https://github.com/kinqsradio/COS40007-Artificial-Intelligence-for-Engineering/tree/main/week-03-portfolio/requirements
Documentation: https://github.com/kinqsradio/COS40007-Artificial-Intelligence-for-Engineering/tree/main/week-03-portfolio/docs
Code: https://github.com/kinqsradio/COS40007-Artificial-Intelligence-for-Engineering/tree/main/week-03-portfolio/code

Table of Contents

Summary Table of Studio 3: Activity 6 (1 mark)

SVM Model	Train-test split accuracy	Cross-validation accuracy
Original features	0.771429	0.756818
With hyperparameter tuning	0.771429	0.756818
With K-Best features	0.714286	0.759848
With PCA features	0.857143	0.803030

The original features yielded a train-test split accuracy of 0.771429 and a cross-validation accuracy of 0.756818. Hyperparameter tuning did not improve the performance, as the accuracy remained the same. However, applying K-Best feature selection slightly reduced the train-test accuracy to 0.714286 but improved cross-validation accuracy to 0.759848. The most significant improvement was observed with PCA, which increased the train-test accuracy to 0.857143 and the cross-validation accuracy to 0.803030.

Summary of Table of Studio 3: Activity 7 (1 mark)

Model	Train-test split accuracy	Cross-validation accuracy
SVM	0.771429	0.756818
SGD	0.885714	0.818182
RandomForest	0.942857	0.897727
MLP	0.828571	0.819697

SVM model showed moderate performance with a train-test split accuracy of 0.771429 and cross-validation accuracy of 0.756818. SGD classifier performed better, achieving a train-test accuracy of 0.885714 and a cross-validation accuracy of 0.818182. RandomForest classifier outperformed others with a train-test accuracy of 0.942857 and cross-validation accuracy of 0.897727. MLP classifier also performed well, with a train-test accuracy of 0.828571 and cross-validation accuracy of 0.819697.

Data Collection (1 mark)

Data Overview

The dataset for this project comprises acceleration data from sensors placed at various body positions during two types of meat processing activities: boning and slicing. Specific columns corresponding to the L5 and T12 positions were selected for analysis. The data was extracted, labeled, and combined to create a single dataset for further processing.

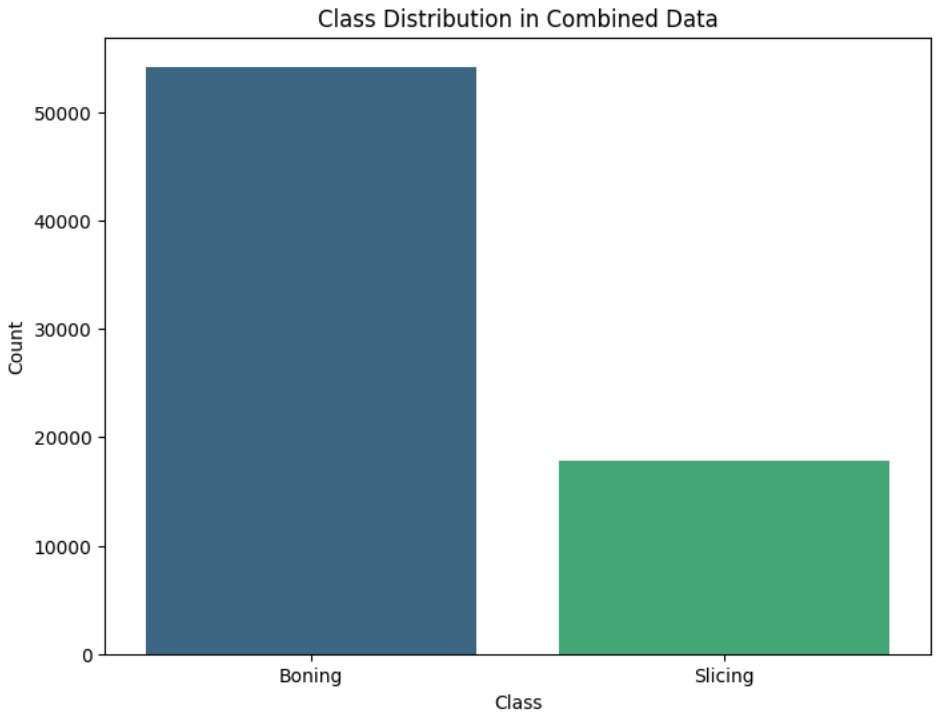
Data Extraction and Preparation

Extracted Columns:

- **Frame**: The frame number.
- **L5 x, L5 y, L5 z**: Acceleration data along the x, y, and z axes from the L5 sensor.
- **T12 x, T12 y, T12 z**: Acceleration data along the x, y, and z axes from the T12 sensor.
- **Class labels** has been added to distinguish between the two activities (0 for boning, 1 for slicing), and the datasets were merged into a single DataFrame.

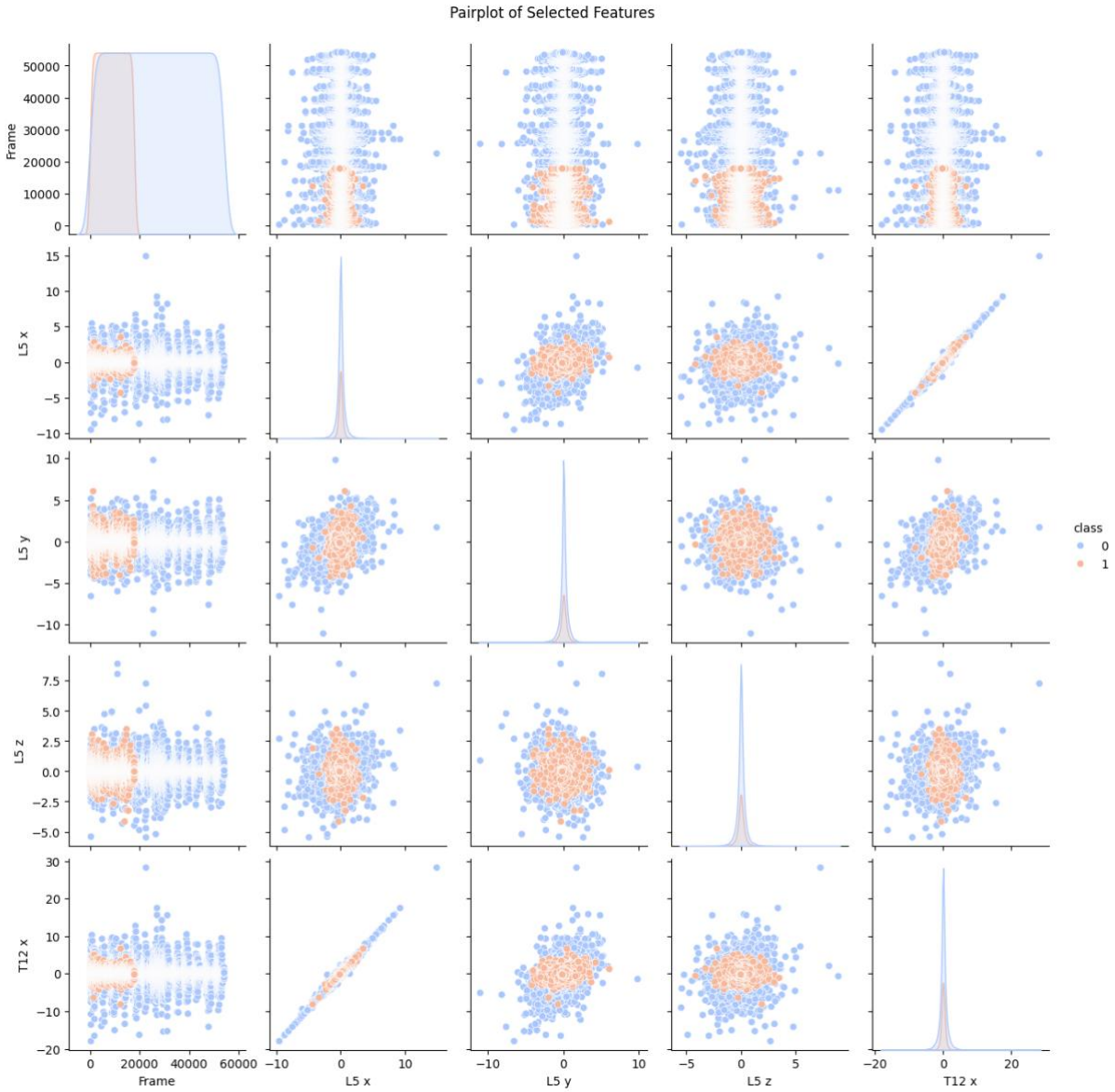
Data Inspection and Visualization

Class Distribution



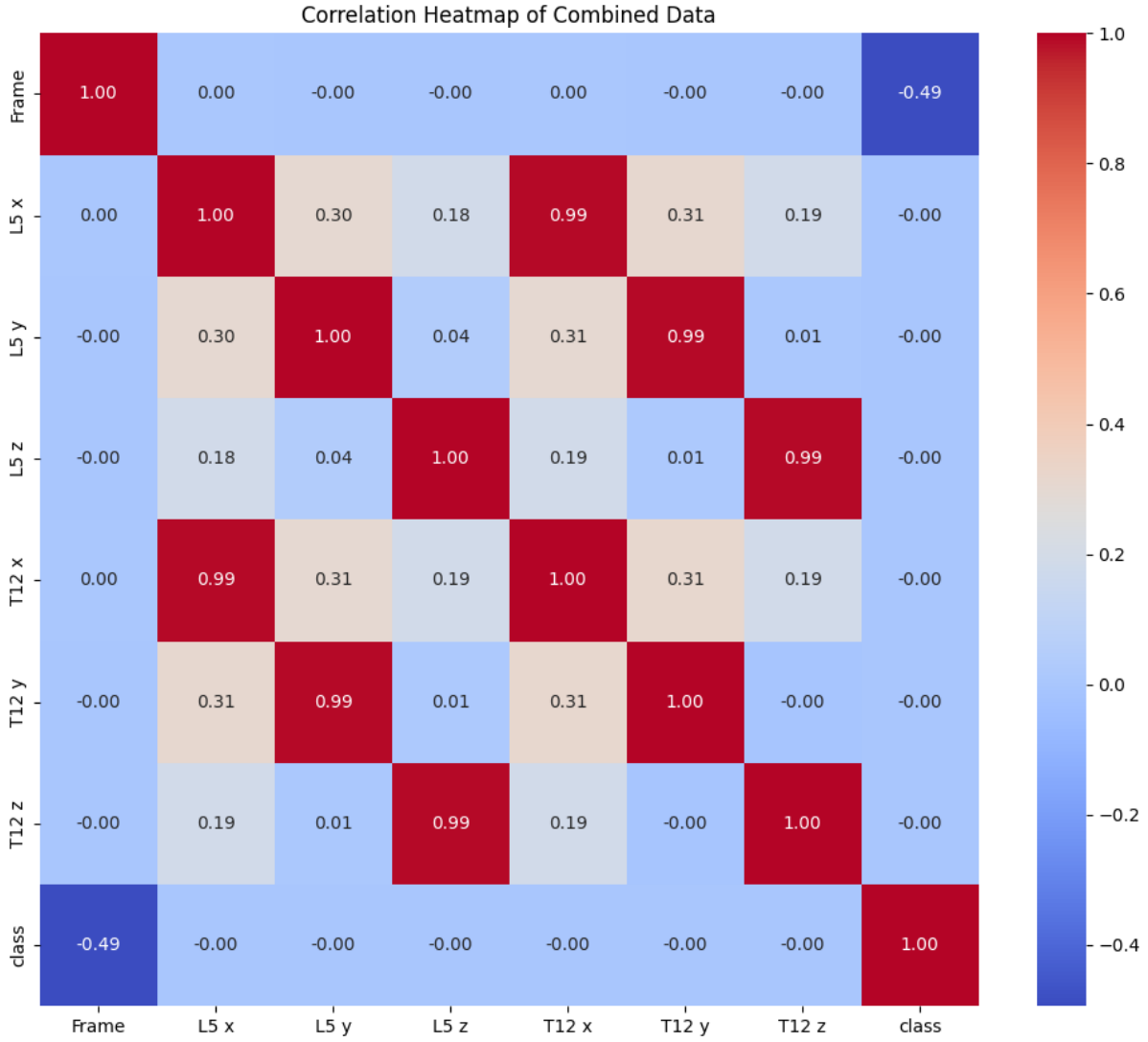
The bar chart shows that the dataset is imbalanced, with more samples labeled as boning than slicing. This imbalance could impact the model's performance and should be considered during model training.

Pairplot of Selected Features



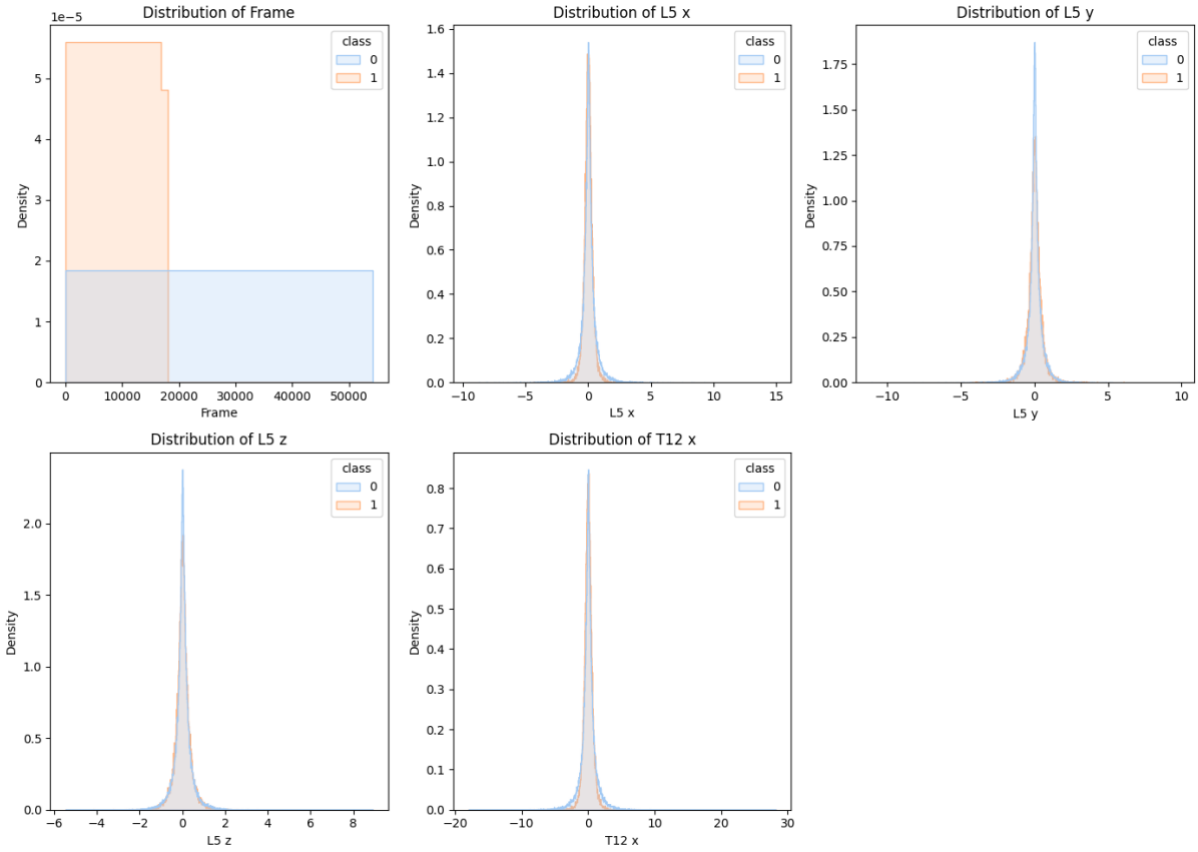
This pairplot illustrates the distribution and potential correlations between the selected features for each class. While some features exhibit distinct patterns between the two classes, others show significant overlap, which may affect the model's ability to distinguish between the two activities.

Correlation Heatmap



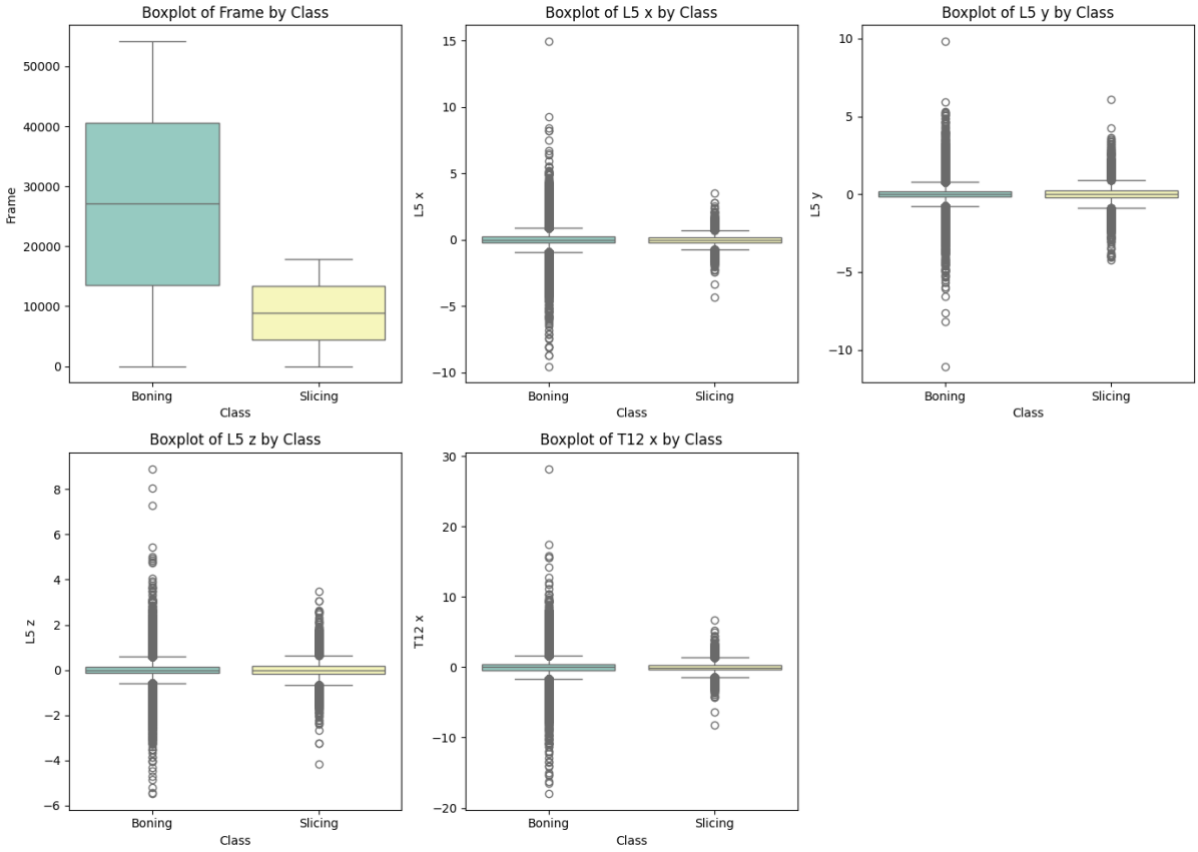
The heatmap reveals strong correlations between certain features, particularly between the L5 and T12 sensors' x-axis acceleration values. These correlations might indicate redundancy, suggesting that dimensionality reduction techniques like PCA could be beneficial.

Feature Distributions by Class



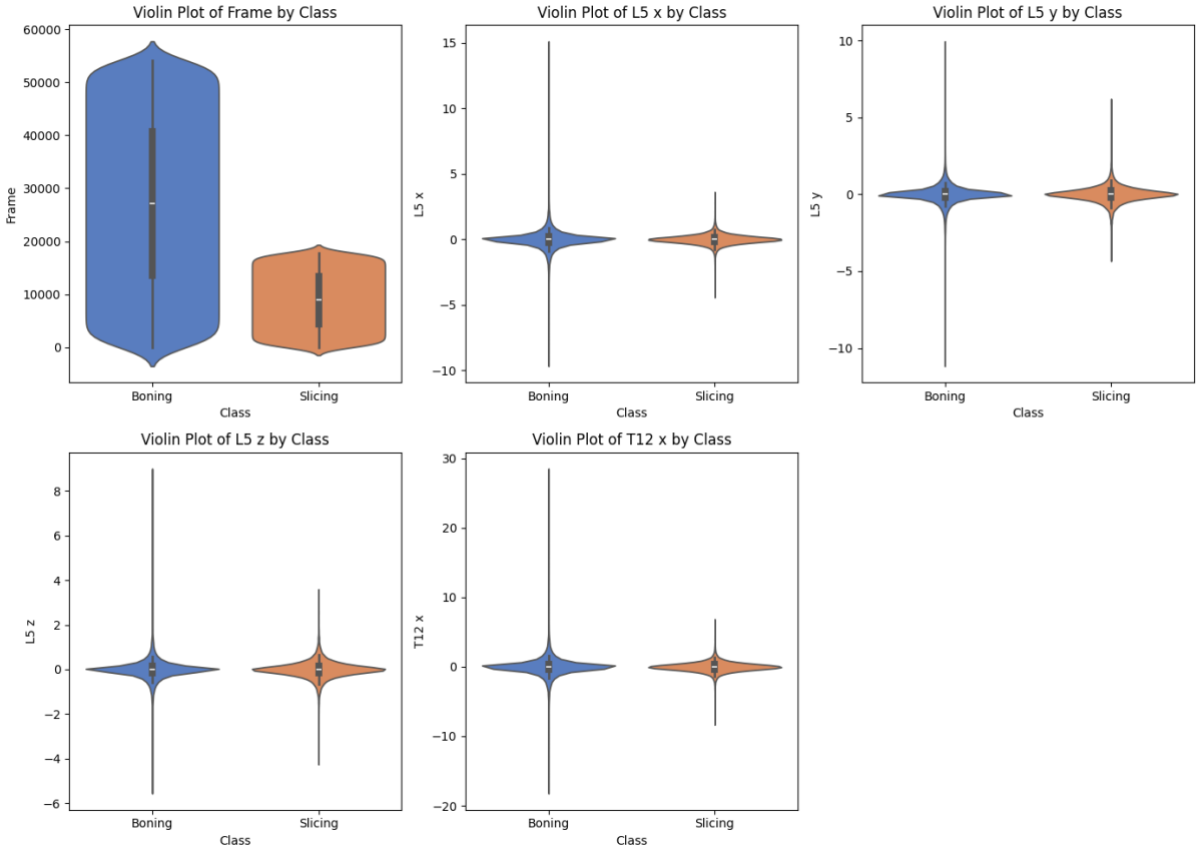
These distribution plots provide insights into how each feature varies between boning and slicing activities. The overlap between the two distributions, particularly in the L5 x and T12 x features, suggests that some features may not contribute strongly to the classification task.

Boxplot of Features by Class



The boxplots show the central tendency and variability of the features for each class. The presence of outliers in certain features, like T12 x, could potentially affect model performance.

Violin Plot of Features by Class



These violin plots highlight the distribution of the data and provide a clear visualization of the data's density across different ranges, offering a more nuanced view of the data distribution for each feature.

Comments on Dataset

The data collection and inspection process has provided valuable insights into the structure and characteristics of the dataset. The visualizations indicate that while there is some distinguishability between the two classes, the overlap and correlation between certain features may present challenges in developing an effective classification model. These insights will guide the subsequent steps in feature engineering and model development.

Create composite columns (1 mark)

The composite columns created include various Root Mean Square (RMS) values and orientation-based calculations like Roll and Pitch. These features are derived from the L5 sensor's x, y, and z axis acceleration data, as follows:

Composite Columns Calculation

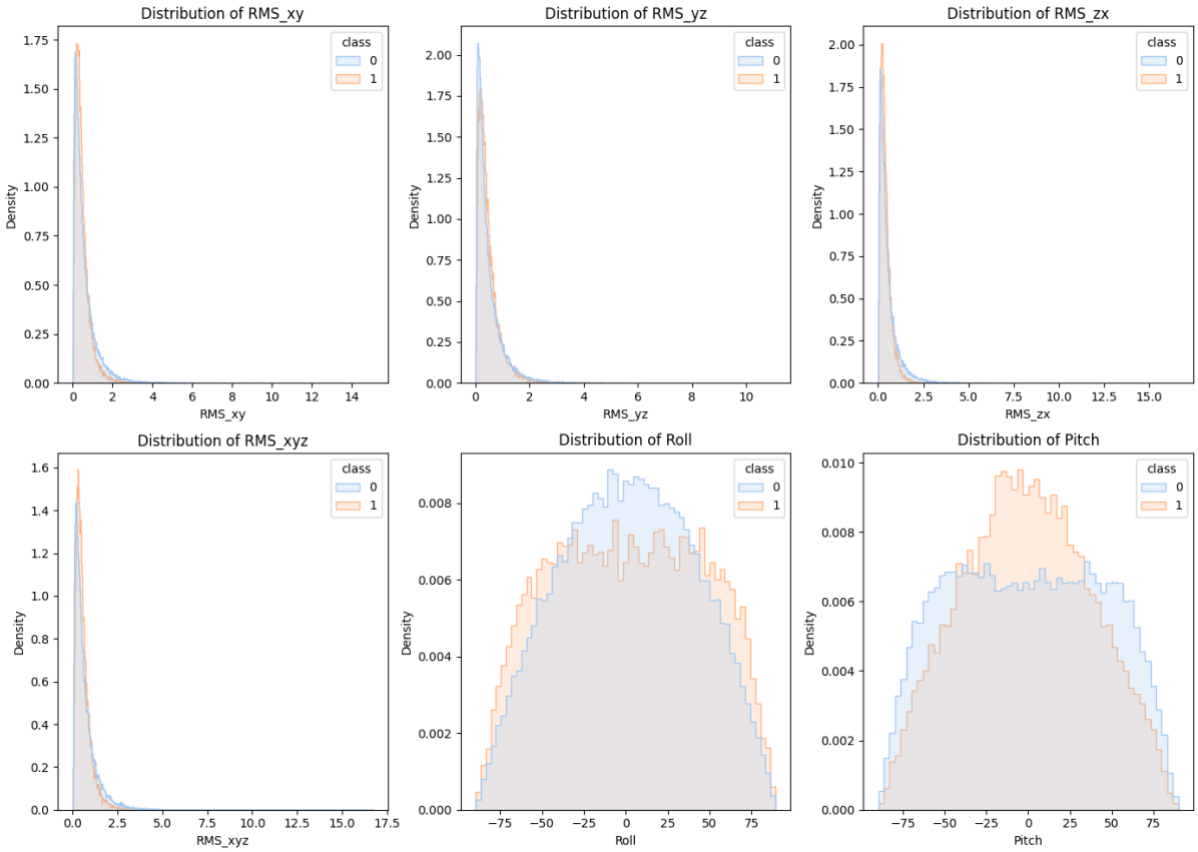
Computed composite columns:

- **RMS_xy**: Root mean square value of x and y.

- **RMS_yz**: Root mean square value of y and z.
- **RMS_zx**: Root mean square value of z and x.
- **RMS_xyz**: Root mean square value of x, y, and z.
- **Roll**: 180/pi x arctan2(accelY, sqrt(accelX^2 + accelZ^2))
- **Pitch**: 180/pi x arctan2(accelY, sqrt(accelY^2 + accelZ^2))

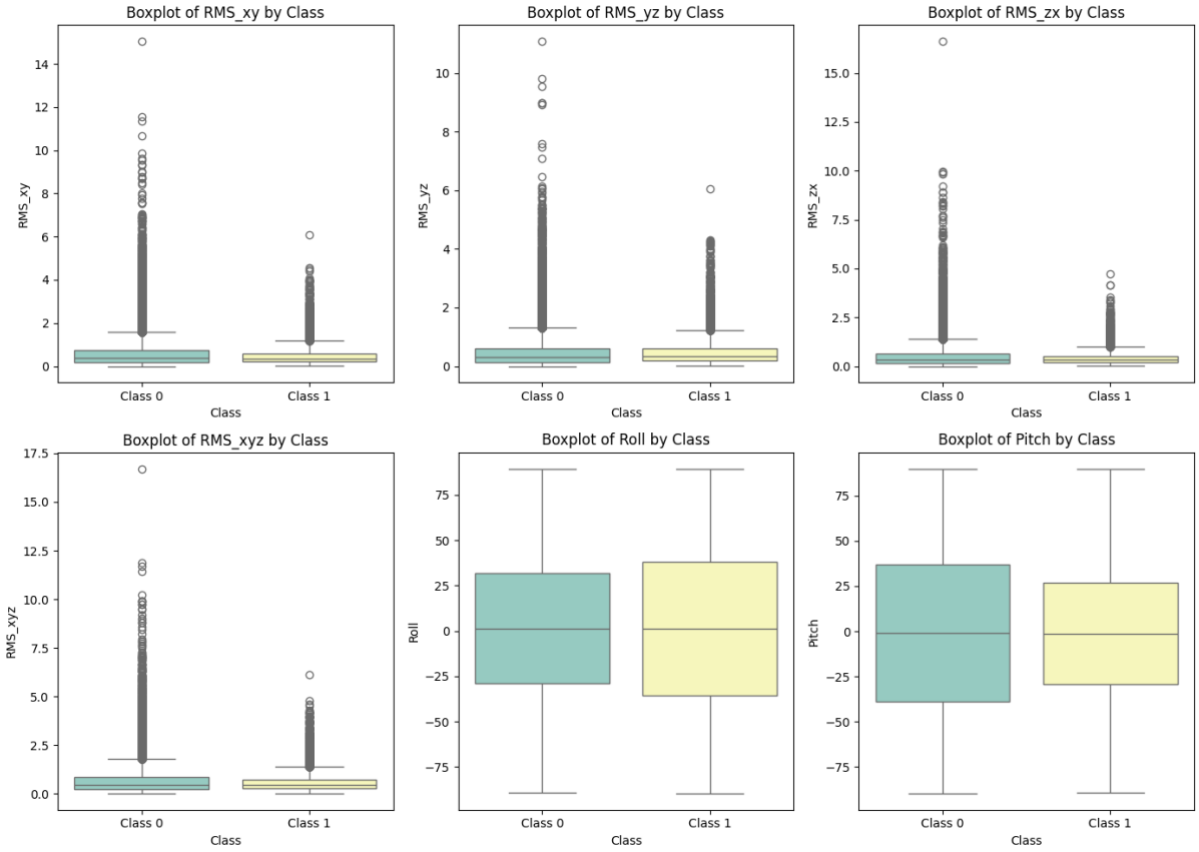
Data Visualization and Analysis

Distribution of Composite Features



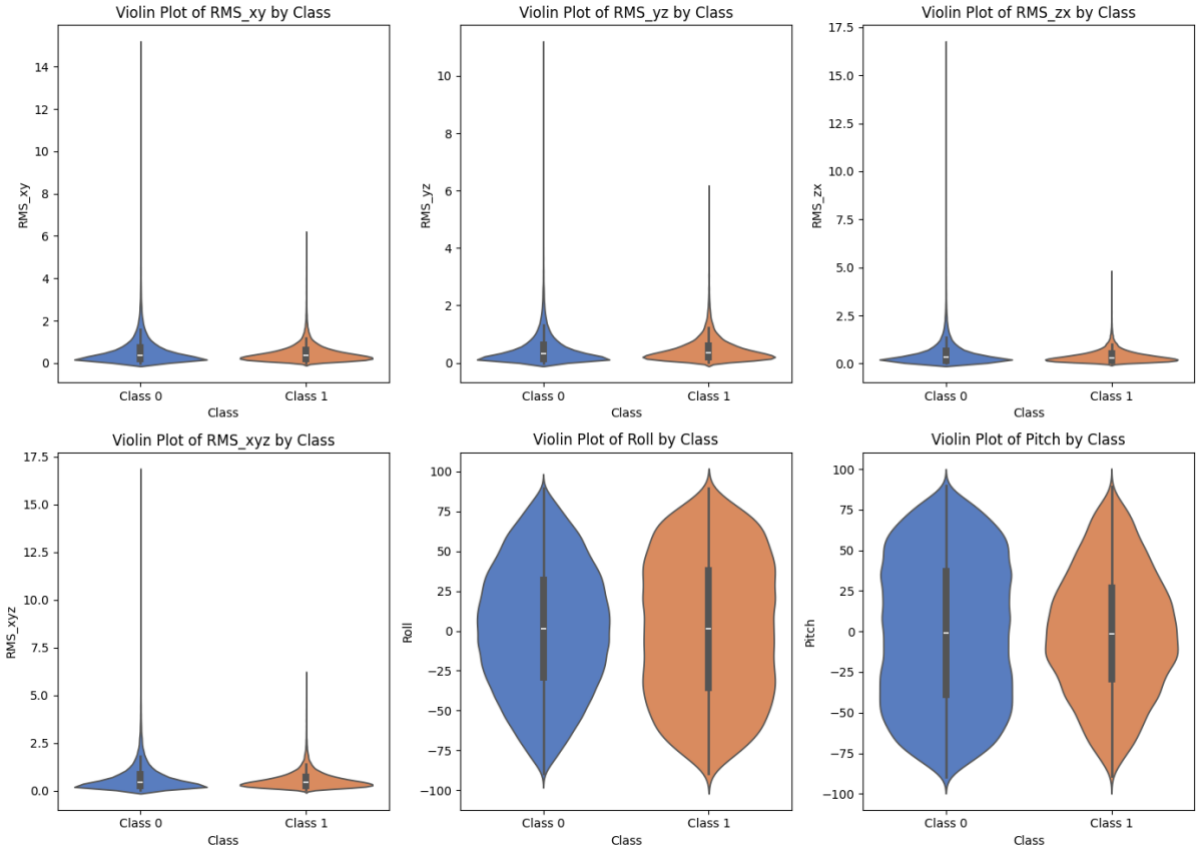
The histograms show the density distribution of each composite feature for both classes. Some features, such as RMS_xy and RMS_yz, show a slight difference in distribution between the two classes, which may provide useful information for classification.

Boxplots of Composite Features by Class



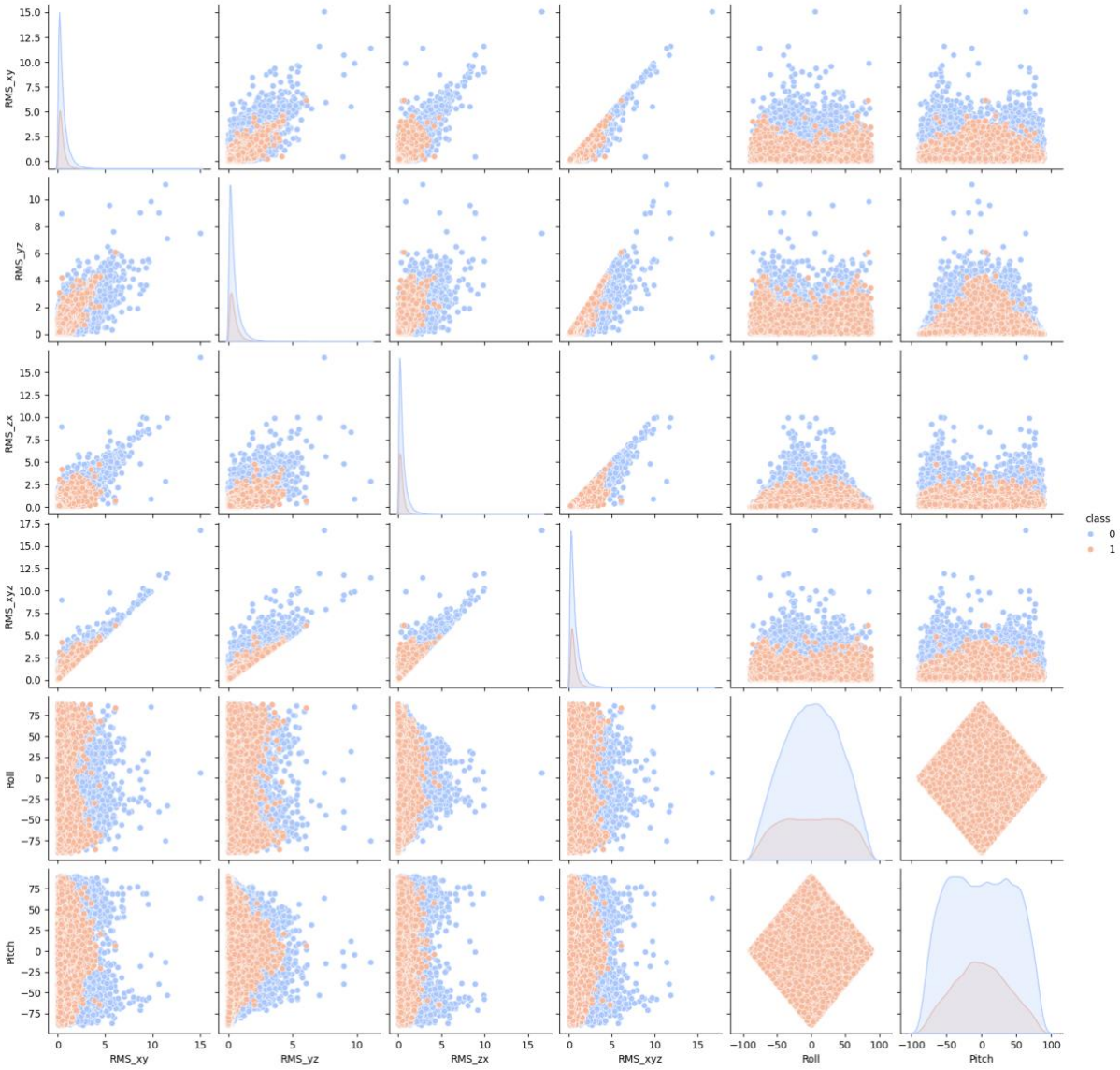
The boxplots illustrate the central tendency and variability of the composite features. Notably, the Roll and Pitch features have relatively symmetrical distributions with fewer outliers, while RMS features show more variance.

Violin Plots of Composite Features by Class



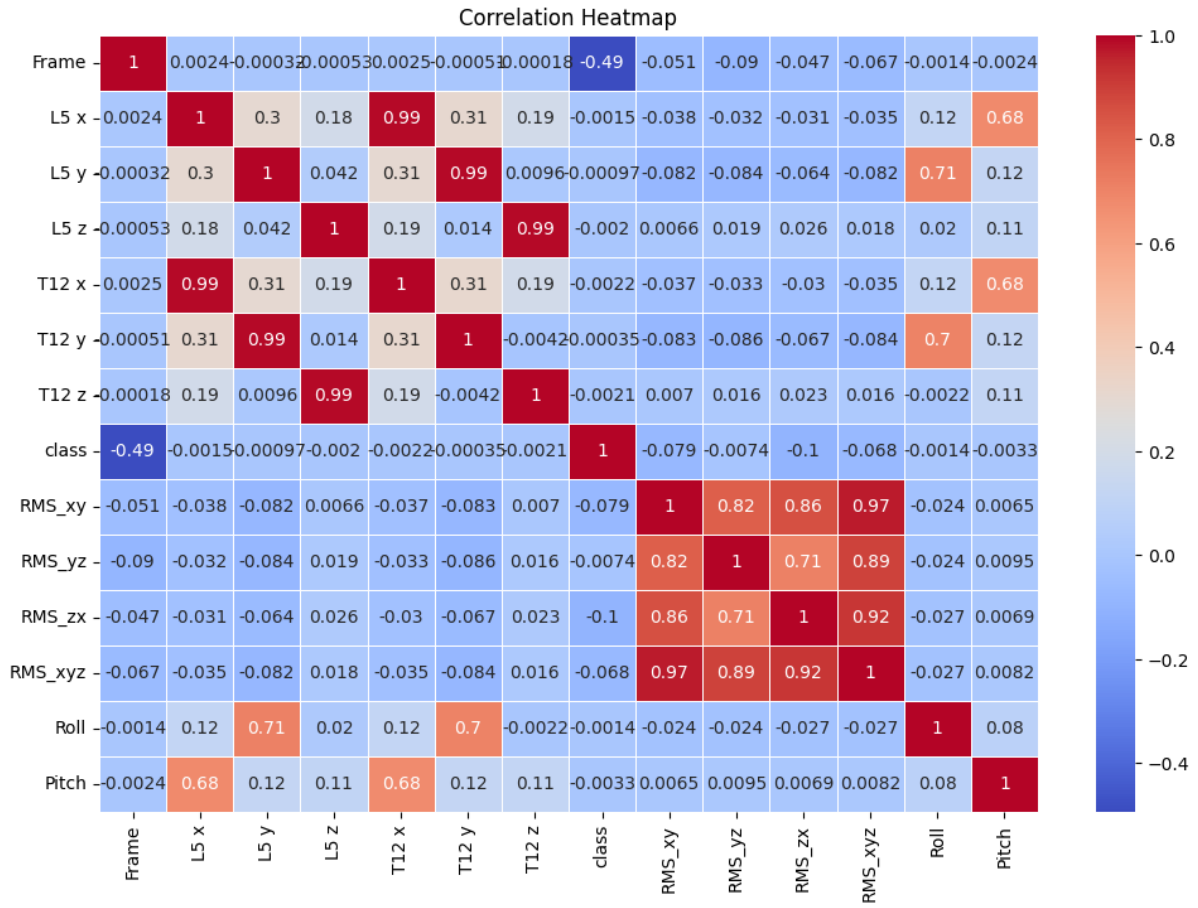
The violin plots provide a detailed view of the distribution's shape, indicating that certain features like Roll and Pitch might have more predictive power due to their distinct distributions between classes.

Pairplot of Composite Features



The pairplot reveals how the composite features relate to each other and to the class labels. Features like RMS_xyz and RMS_zx appear to have some level of separation between the classes, which could be beneficial for the classification task.

Correlation Heatmap of Combined Data



The heatmap shows strong correlations between the RMS features, which might indicate redundancy. However, the Roll and Pitch features show moderate correlations with the RMS features, suggesting they may capture additional information useful for classification.

Comment on Dataset Compsite Columns

The creation of composite columns has introduced new features that provide additional insights into the data, potentially improving the classification model's performance. The visualizations indicate that these features exhibit varying levels of separability between the two classes, which will guide the feature selection process in the subsequent steps.

Data pre-processing (3 marks)

The preprocessing involves grouping the data by minute intervals and computing various statistical features, including mean, standard deviation, minimum, maximum, area under the curve (AUC), and the number of peaks for each feature.

Feature Computation

The function **compute_features** was created to calculate several key statistical properties for each sensor data column:

- **Mean (_mean):** The average value of the data.
- **Standard Deviation (_std):** Measures the amount of variation or dispersion.
- **Minimum (_min):** The smallest value in the data.
- **Maximum (_max):** The largest value in the data.
- **Area Under Curve (AUC, _auc):** Computed using the trapezoidal rule to measure the integral of the data, which can be related to the overall activity.
- **Peaks (_peaks):** The number of peaks found in the data, indicating the frequency of significant movements.

These features were calculated for each column of the sensor data (L5 x, L5 y, L5 z, T12 x, T12 y, T12 z, and composite columns).

Grouping Data by Time Intervals

The data was grouped by each minute, assuming 60 frames per minute. This resulted in a reduced and more manageable dataset where each row represents aggregated data over one minute. The class labels were preserved in the grouping process to ensure that the final dataset could still be used for classification.

Inspection of Processed Data

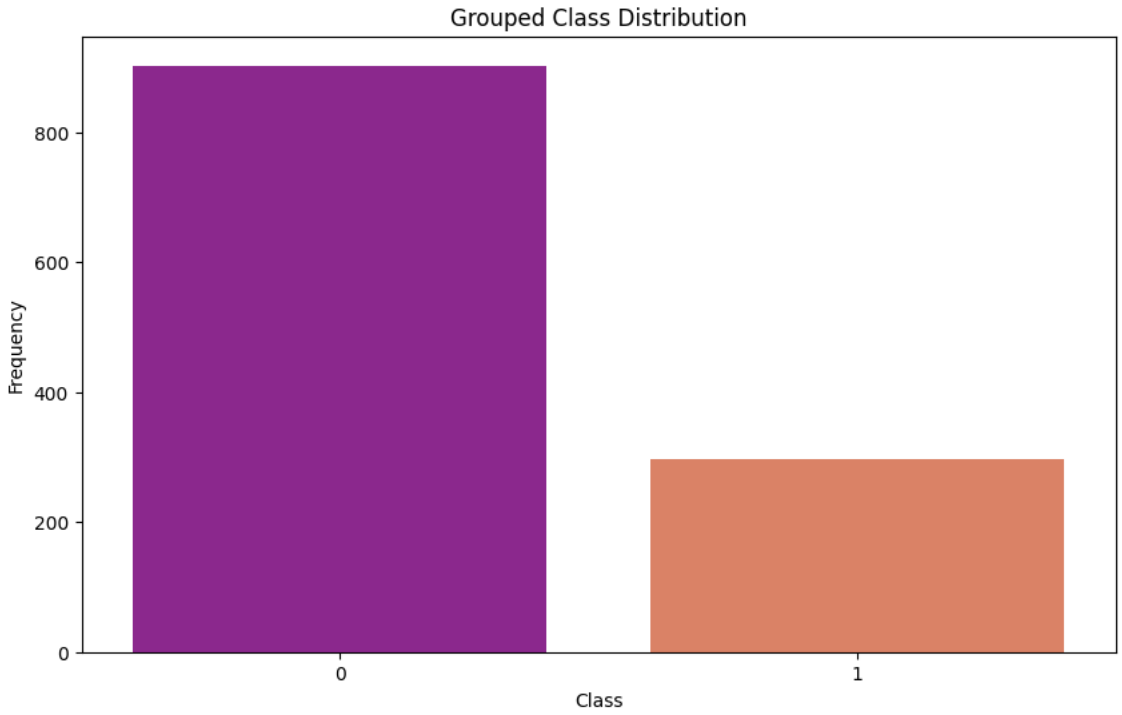
After grouping and feature computation, the processed dataset was inspected to ensure the integrity and correctness of the transformations:

- **Shape:** The processed dataset has 1,201 rows and 74 columns.
- **Columns:** The dataset contains columns for each computed feature, along with the class and Frame columns.
- **Data Types:** All computed features are in the float64 format, suitable for numerical analysis.

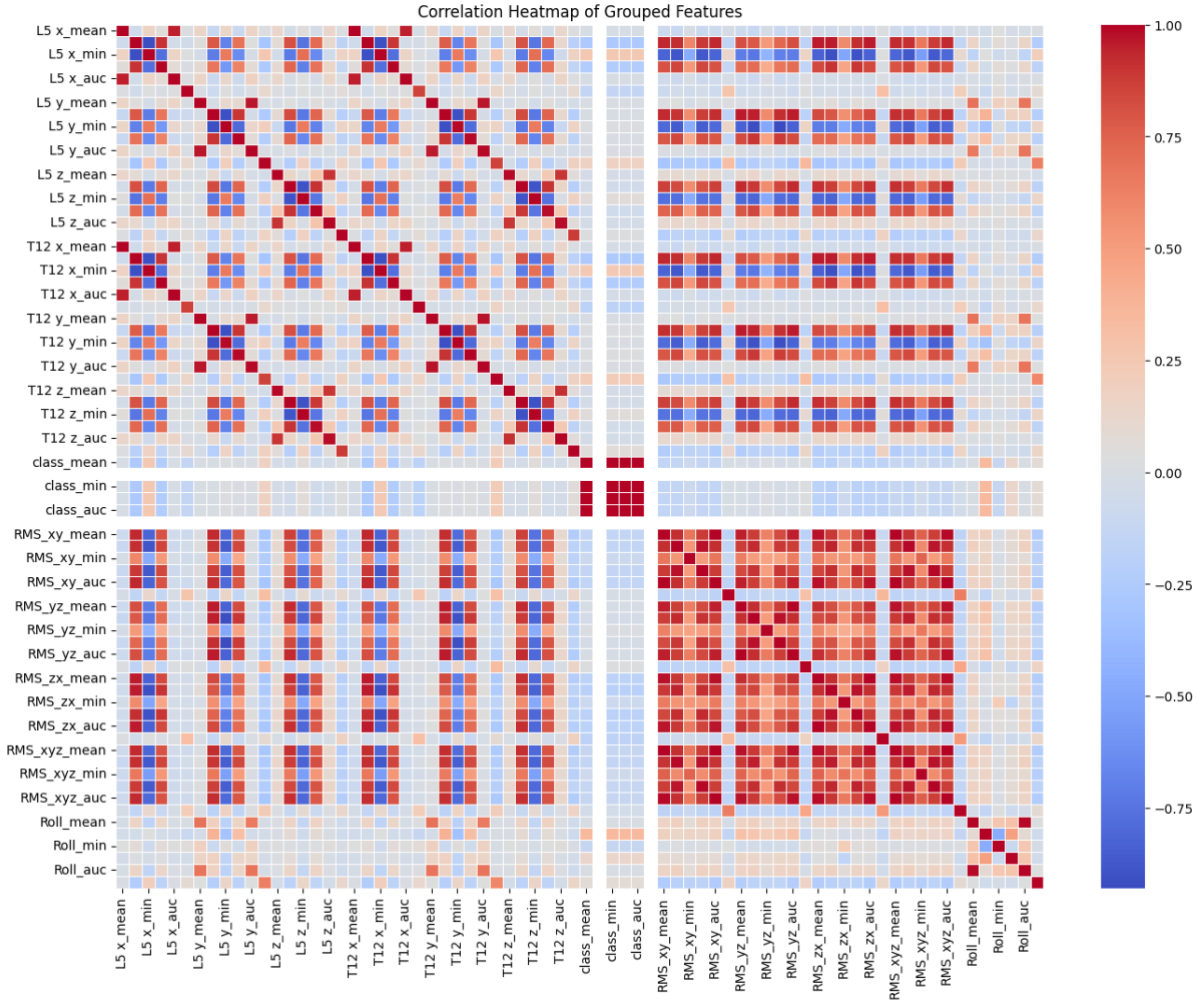
The data inspection confirmed that the preprocessing was successful and that the dataset is ready for further analysis.

Visualizations of Processed Data

Grouped Class Distribution

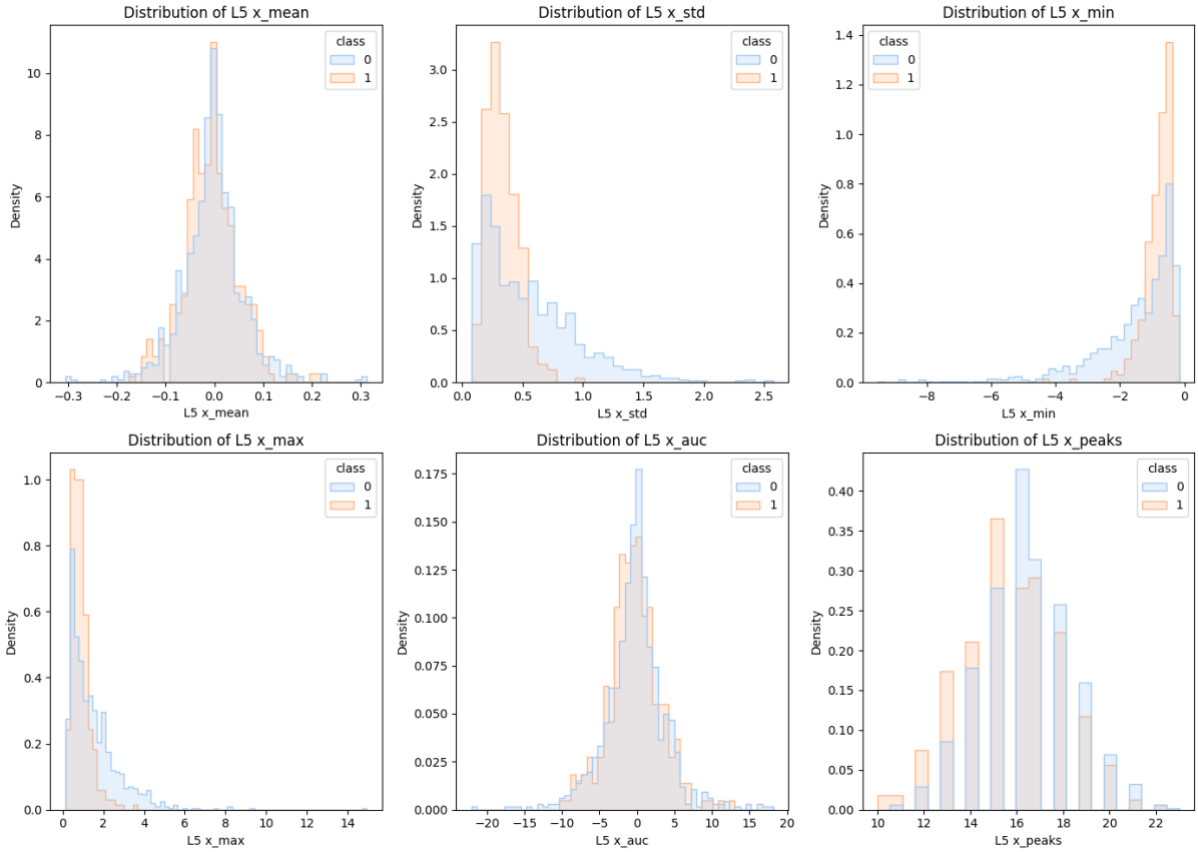


Correlation Heatmap of Grouped Features



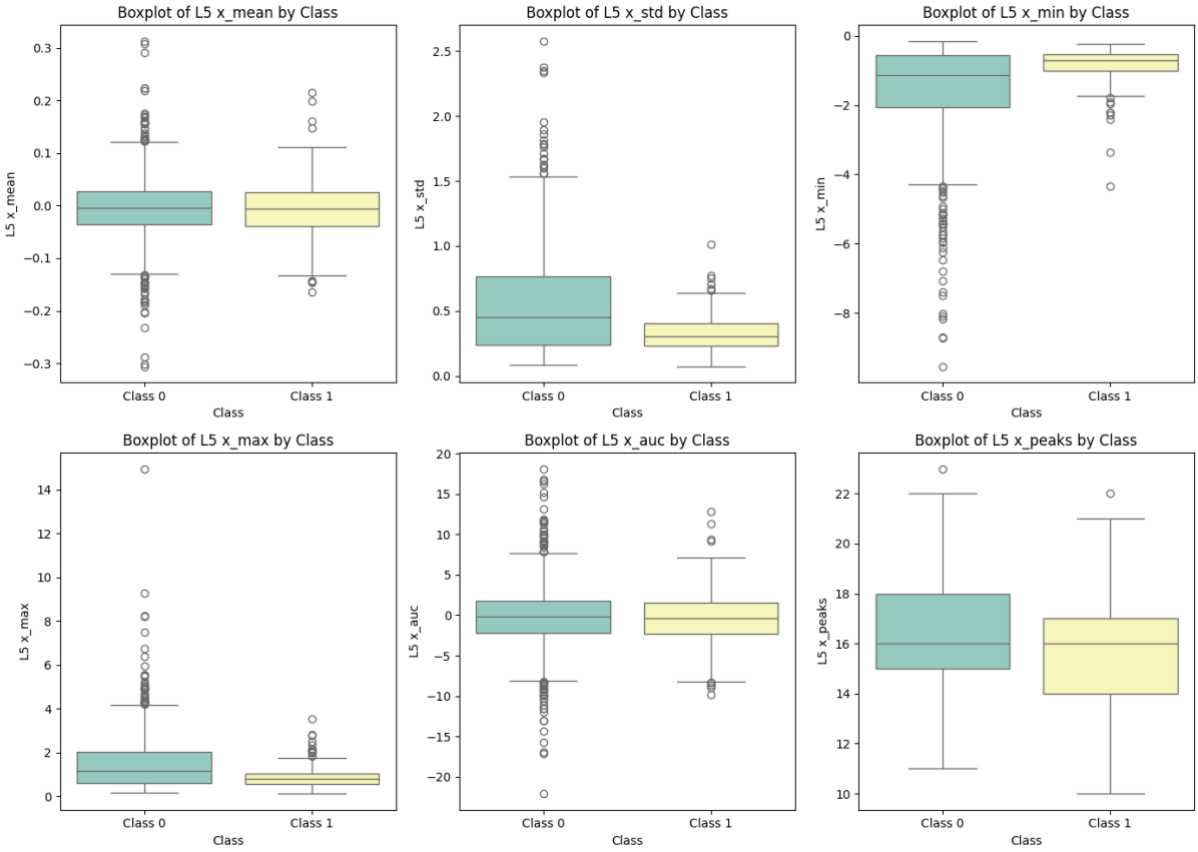
This heatmap helps identify highly correlated features that might be redundant in the model.

Distribution of Selected Grouped Features



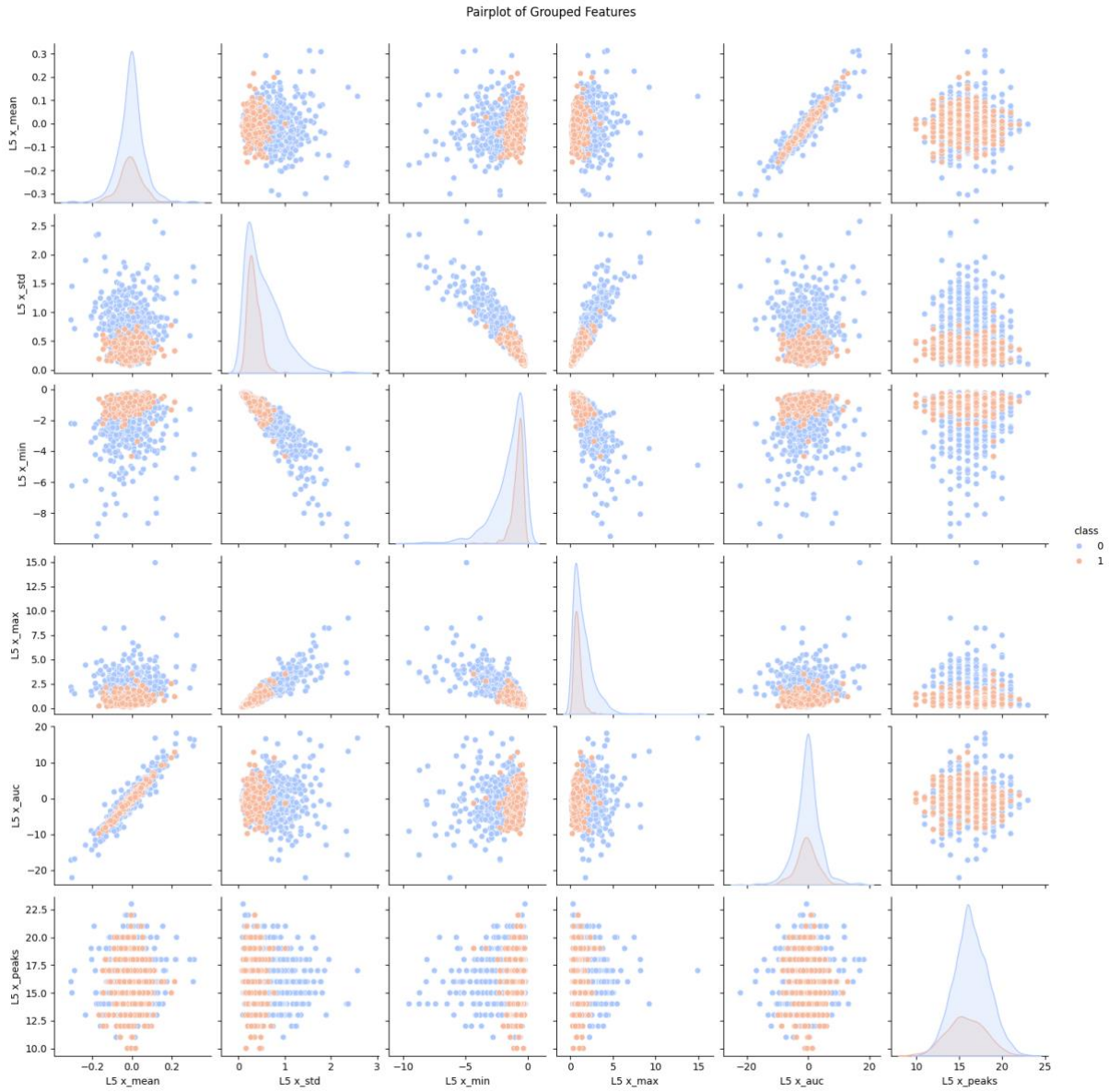
The histograms provide insight into the separability of the classes based on these features.

Boxplots of Selected Grouped Features by Class



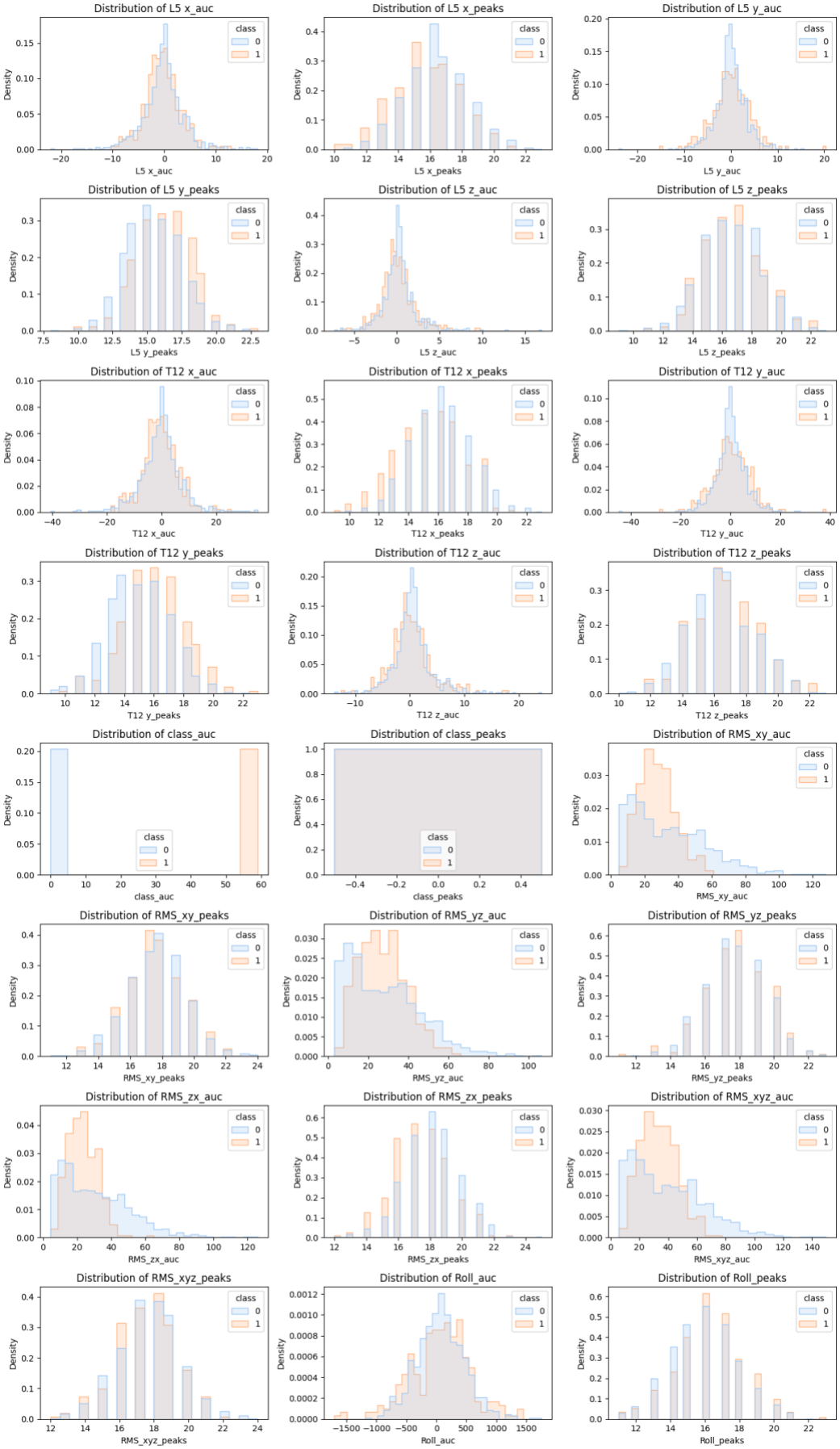
These boxplots help identify features with different distributions between classes, which might be critical for classification.

Pairplot of Grouped Features



The plot reveals interactions between features like L5 x_mean, L5 x_std, and L5 x_min, indicating some correlation and potential importance in distinguishing the classes.

Analysis of Peaks and AUC Distributions



These plots indicate how often significant movements occur and their overall intensity, which can be useful for distinguishing between the two activities.

Comment on Data Pre-processing

The preprocessing step successfully transformed the raw data into a structured format with aggregated features that capture the essential characteristics of the sensor data. These features will be critical in the subsequent modeling phase, where we aim to build a robust classifier to distinguish between boning and slicing activities.

Model Training (2 marks)

Models

- SVM: Trained with a linear kernel and a regularization parameter of C=1.0.
- Random Forest: A collection of 100 decision trees was used for this ensemble learning model.
- SGDClassifier: An implementation of stochastic gradient descent, trained with a maximum of 1000 iterations.
- MLPClassifier: A neural network model with two hidden layers, each containing 50 neurons.

Training Phases

- Initial Training without Hyperparameter Tuning: All models were trained on the original feature set, and their performance was evaluated using 10-fold cross-validation on the test set.
- Hyperparameter Tuning: GridSearchCV was applied to find the optimal parameters for each model. The tuned models were then re-evaluated using cross-validation.
- Training with Feature Selection: The 10 most significant features were selected using SelectKBest, and models were retrained and evaluated.
- Training with Principal Component Analysis (PCA): Dimensionality reduction was performed using PCA to reduce the feature space to 10 components, and models were retrained on this reduced feature set.

Summary Results

SVM Model	Train-test split accuracy	Cross-validation accuracy
Original features	1.0000	1.0000
With hyperparameter tuning	1.0000	1.0000
With K-Best features	1.0000	1.0000
With PCA features	1.0000	1.0000

RandomForest Model	Train-test split accuracy	Cross-validation accuracy
Original features	1.0000	1.0000
With hyperparameter tuning	1.0000	1.0000
With K-Best features	1.0000	1.0000
With PCA features	1.0000	1.0000

SGD Model	Train-test split accuracy	Cross-validation accuracy
Original features	1.0000	1.0000
With hyperparameter tuning	1.0000	1.0000
With K-Best features	1.0000	1.0000
With PCA features	1.0000	1.0000

MLP Model	Train-test split accuracy	Cross-validation accuracy
Original features	0.9944	0.9944
With hyperparameter tuning	0.9944	0.9944
With K-Best features	1.0000	1.0000
With PCA features	0.9972	0.9972

Model Selection (1 mark)

I have chosen SVM model with hyperparameter tuning was selected as the final model due to its simplicity, interpretability, and strong performance base on these criteria:

- **Simplicity**: While all models performed exceptionally well, the SVM model with a linear kernel was chosen due to its simplicity and interpretability. SVMs are less prone to overfitting in high-dimensional spaces, making them a good choice for this task.
- **Computation Efficiency**: SVM also required less computational time compared to Random Forest and MLPClassifier, making it more efficient for large-scale deployment.

Discussion

The results obtained from the model training and evaluation indicate that the dataset is well suited for classification tasks, with most models achieving near-perfect accuracy. However, the consistently high accuracy scores across different models and training phases raise concerns about overfitting.

There is a chance of possible overfitting as the perfect accuracy scores suggest that the models may have memorized the training data, leading to overfitting. This means the models might perform well on the current dataset but may not generalize well to new, unseen data.

There are a few options we that we might do:

1. **Validation on Unseen Data**: It is recommended to test the models on a separate validation dataset to confirm their generalizability.
2. **Regularization**: Applying regularization techniques could help mitigate overfitting.
3. **Exploration of Other Models**: While the selected SVM model performed well, exploring other models such as ensemble methods with regularization could provide additional insights.

Conclusion

In week 3 portfolio, various machine learning models, including SVM, Random Forest, SGD, and MLP, were trained and evaluated on the given dataset. After thorough testing, the SVM model with hyperparameter tuning was chosen as the final model due to its simplicity, strong performance, and computational efficiency. The models demonstrated excellent accuracy, but care must be taken to ensure generalizability due to potential overfitting. Overall, the results highlight the effectiveness of applying machine learning techniques to classify the data with high precision.