Evaluating Logistic Regression for High-Dimensional Human Activity Recognition

Stanislav Gatin
University of Alicante – Polytechnic School
Advanced Learning – Academic Year 2025/2026

Abstract

This work presents a complete analysis of the Machine Learning workflow applied to the *Human Activity Recognition Using Smartphones* dataset. The main objective is to classify six types of physical activities based on inertial sensor data. Challenges such as high dimensionality (561 features) and the simulation of missing values to evaluate imputation strategies were addressed. The methodology included extensive preprocessing: outlier handling using winsorization, statistical (mean/median) and KNN-based imputation, and correlation-based feature selection to reduce multicollinearity. Sixteen experimental configurations were evaluated using Logistic Regression within a cross-validated pipeline. Results showed that the model trained on the full dataset, imputed using the median (Method 2) and with outlier treatment, achieved the best performance (F1-Score: 0.986). Additionally, it was demonstrated that dimensionality can be reduced to 25 features while maintaining competitive accuracy above 94%, validating the efficiency of the proposed feature selection approach.

Keywords: Machine Learning, HAR, Classification, Imputation, Feature Selection, Logistic Regression.

1 Introduction

1.1 Selected Dataset

For this practice, the **Human Activity Recognition Using Smartphones** dataset was selected, available in the UCI Machine Learning Repository. This dataset contains accelerometer and gyroscope data collected from smartphones while users performed different physical activities.

This dataset was chosen because it is a quantitative, structured, and relevant source that allows the development of models capable of automatically classifying human activities. Its utility extends significantly into **Human-Centered Computing**, particularly in the following areas:

- **Healthcare and Elderly Care:** The dataset design explicitly accounts for users with slower movements. The signal processing assumed a minimum speed equal to 50% of the average human cadence to ensure that elderly and disabled individuals could benefit from the monitoring system.
- **Unobtrusive Monitoring:** Unlike dedicated body-worn sensors (attached to the chest, wrist, or thighs) which can be uncomfortable and require precise repositioning after dressing, this dataset validates the use of mass-market smartphones as a "flexible, affordable and self-contained solution" for long-term monitoring of Activities of Daily Living (ADL).
- **Fall Detection and Context Awareness:** By analyzing triaxial linear acceleration and angular velocity, the system can gather context information about people's actions, which is foundational for fall detection systems in independent living scenarios.

1.1.1 Technical Specifications and Interesting Characteristics

The dataset possesses several unique characteristics derived from its collection protocol involving 30 volunteers (ages 19-48) wearing a waist-mounted Samsung Galaxy S II.

1. **Windowing Strategy:** To capture coherent motion patterns, the time signals were sampled in fixed-width sliding windows of 2.56 seconds with a 50% overlap. This specific duration was chosen to guarantee that at least one full walking cycle (two steps) is captured per window, assuming a cadence of 90-130 steps/min.
2. **Gravity Separation:** A low-pass Butterworth filter with a 0.3 Hz corner frequency was used to separate the gravitational force component from body motion, assuming gravity has only low-frequency components.
3. **Classification Challenges:** Preliminary benchmarks in the source study highlighted that while dynamic activities are generally distinguishable, static postures present a challenge. Specifically, the *Sitting* activity had the lowest recall (88%) due to a "noticeable misclassification overlap" with the *Standing* class, attributed to the similar device orientation in both postures.

1.2 Objective

The primary goal of this study is to solve a supervised multiclass classification problem within the domain of Human Activity Recognition (HAR). The objective is to construct a model capable of accurately mapping a feature vector, derived from inertial sensor data, to one of six specific Activities of Daily Living (ADL).

Formally, given a dataset collected from a waist-mounted smartphone (Samsung Galaxy S II) containing 561 extracted features per time window, the model aims to predict the target variable $y \in \{\text{WALKING, UPSTAIRS, DOWNSTAIRS, SITTING, STANDING, LAYING}\}$.

The specific sub-objectives of this classification task include:

- **Distinguishing Activity Types:** The model must differentiate between *dynamic activities* (walking variants) and *static postures* (sitting, standing, laying).
- **Resolving Class Overlaps:** A key challenge is to correctly classify non-dynamic activities, specifically distinguishing *SITTING* from *STANDING*, which have historically shown misclassification overlaps due to the similar orientation of the device in both postures.
- **Unobtrusive Monitoring:** The ultimate aim is to validate the feasibility of using a single, mass-market device for continuous monitoring, eliminating the need for complex, multi-sensor body networks.

2 Experimental Setup

2.1 Initial Exploration

The original dataset contains 10,299 samples and 561 continuous numerical variables. There are no categorical variables requiring encoding, but there is a high correlation among variables derived from the same sensors.

Table 1: Dataset characteristics

Characteristic	**Value**
Rows (Total)	10,299
Columns	561
Numerical variables	561
Categorical variables	0
Target variable	Activity (6 classes)
Problem type	Multiclass Classification

2.2 Exploratory Data Analysis (EDA)

2.2.1 Class Distribution Analysis

The dataset comprises six distinct activity classes. Figure 1 illustrates the frequency distribution across these categories. While the dataset is not perfectly uniform, it exhibits a reasonable degree of balance.

The *LAYING* activity constitutes the majority class with 1,944 samples (18.9%), followed closely by *STANDING* (1,906 samples, 18.5%) and *SITTING* (1,777 samples, 17.3%). The dynamic activities show slightly lower frequencies: *WALKING* accounts for 1,722 samples (16.7%), while *WALKING_UPSTAIRS* and *WALKING_DOWNSTAIRS* represent 15.0% (1,544 samples) and 13.7% (1,406 samples) of the data, respectively.

The ratio between the majority class (*LAYING*) and the minority class (*WALKING_DOWNSTAIRS*) is approximately 1.38, indicating only a mild imbalance. Furthermore, the aggregate split between static activities (Laying, Standing, Sitting) and dynamic ones is roughly 54.7% to 45.3%. Given this distribution, standard evaluation metrics such as accuracy remain valid, and aggressive resampling techniques (e.g., SMOTE or random undersampling) were not considered necessary for the initial modeling phase.

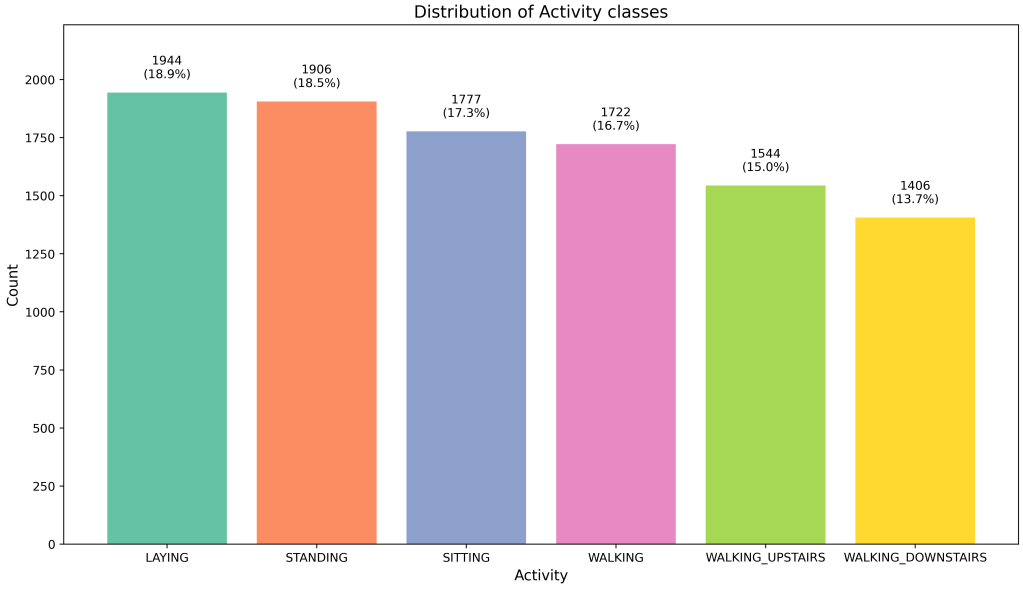


Figure 1: Distribution of samples per activity class. The percentages indicate the relative frequency of each label within the total dataset.

2.2.2 Feature Space Visualization and Separability Analysis

To assess the feasibility of the classification task, we visualized the dataset in a 3-dimensional feature space. This visualization provides insight into the spatial distribution of classes and highlights potential challenges in establishing decision boundaries.

2.2.3 Feature Space Visualization and Separability Analysis

Figure 2 displays the dynamic activities (*WALKING*, *WALKING_UPSTAIRS*, *WALKING_DOWNSTAIRS*) plotted against frequency-domain body acceleration features. While the plot generally reveals a high degree of inter-class overlap, forming a contiguous cloud, a distinct dispersion pattern can be observed for *WALKING_DOWNSTAIRS* (green points).

This separation is likely attributable to the specific biomechanics of descent. Unlike walking on a flat surface or ascending (which rely heavily on concentric muscle action), walking downstairs involves a "controlled fall" of the center of mass. This requires the body to operate primarily in an **eccentric mode** specifically, the quadriceps must actively brake knee flexion, while the gluteal and calf muscles

control the lowering of the pelvis. Furthermore, due to the shorter support phase and higher risk of instability, there is an increased recruitment of stabilizer muscles (core, gluteus medius) and heightened proprioceptive control to maintain balance. These unique physical demands braking forces and rapid stabilization corrections, result in a distinct acceleration signature compared to the more rhythmic patterns of level walking or ascending.

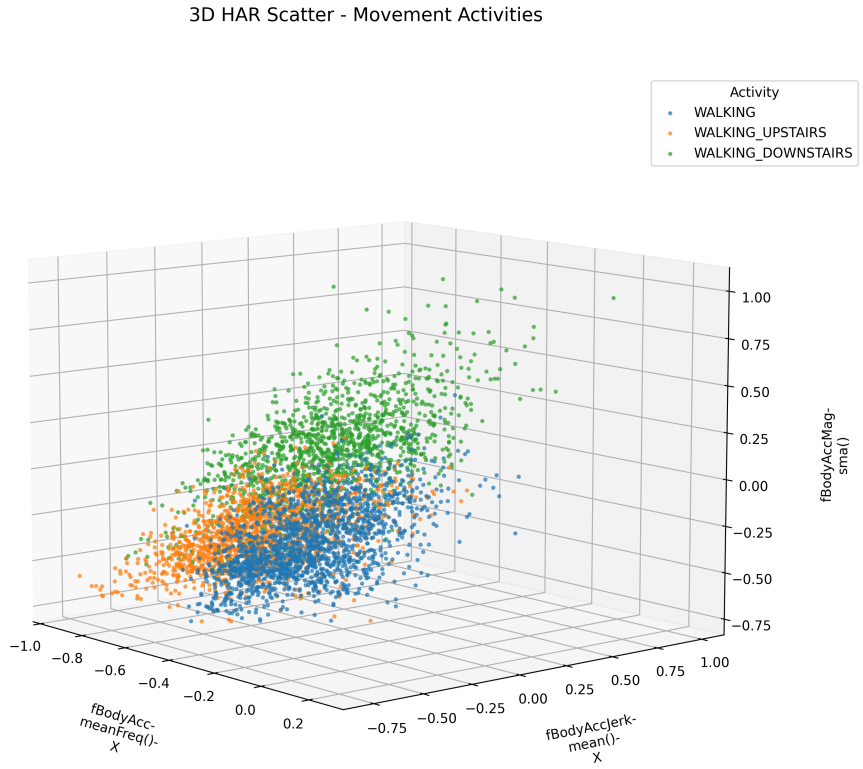


Figure 2: 3D Scatter Plot – Movement Activities. Note that *WALKING_DOWNSTAIRS* shows a divergent spread due to the biomechanical necessity of eccentric braking and stabilization.

In contrast, Figure 3 illustrates the static activities (*SITTING*, *STANDING*, *LAYING*) using time-domain gravity acceleration features. Here, the separability is distinct but non-uniform. The *LAYING* class forms a completely isolated cluster, significantly separated from the others. This separation is attributed to the distinct orientation of the accelerometer axes relative to gravity when the subject is supine (gravity acts primarily on a different axis compared to upright postures).

However, the *SITTING* and *STANDING* classes exhibit a much closer proximity, with noticeable overlapping regions in the lower manifold of the plot. This suggests that without specific features (possibly from a gyroscope to detect hip orientation), distinguishing these two upright postures based solely on acceleration remains a challenge.

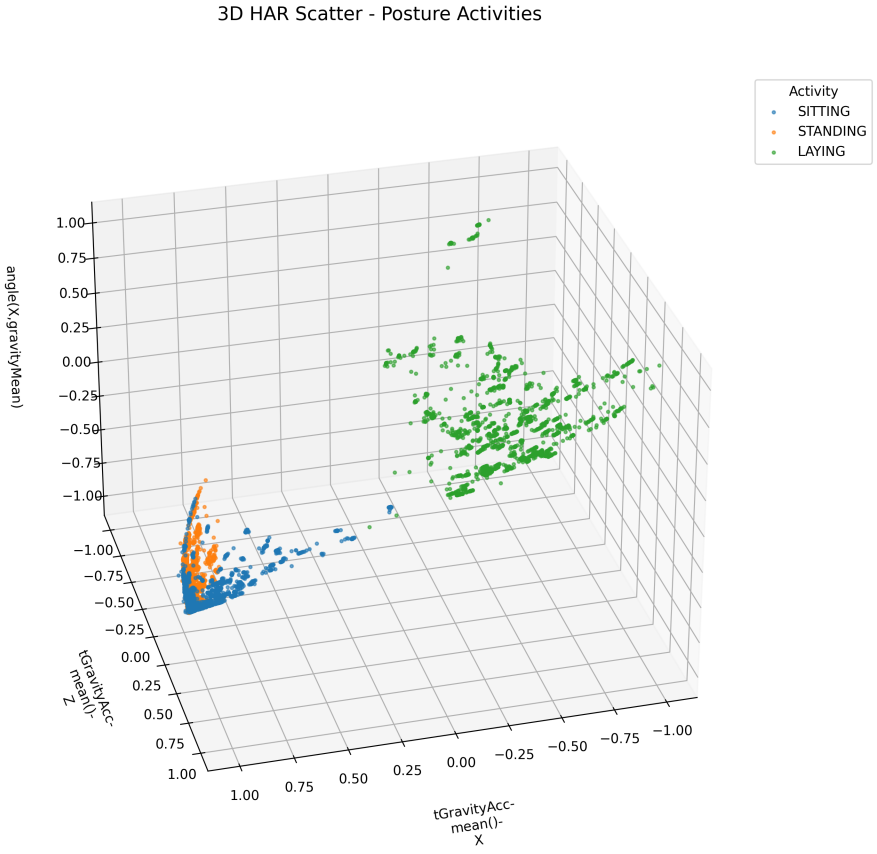


Figure 3: 3D Scatter Plot – Posture Activities. *LAYING* is clearly separable, whereas *SITTING* and *STANDING* show proximity.

Implications for Model Training: The visual evidence suggests two primary behaviors for the learning algorithm:

1. **High Precision for LAYING:** Due to the clear spatial gap seen in Figure 3, the model is expected to classify *LAYING* with near-perfect accuracy and minimal confusion.

2. **Potential Confusion Pairs:** The overlaps observed in both figures indicate critical "confusion zones." Specifically, the model may struggle to distinguish *SITTING* from *STANDING* without sufficiently distinct features. Similarly, the lack of clear separation in the dynamic classes (Figure 2) implies that misclassification among the walking variants is the most likely source of error. If the model fails to learn subtle structural differences within these clusters, it will default to predicting the dominant class in that region, reducing overall sensitivity.

To complement the 3D visualization, Figure 4 presents a pairwise analysis of key features selected from different domains (Angle, Gravity, and Magnitude).

The diagonal density plots reveal a crucial insight: `tBodyAccMag-std()` (bottom-right diagonal) acts as a perfect discriminator between static activities (sharp peak at -1) and dynamic activities (broader distribution). Furthermore, the scatter plots involving `angle(X,gravityMean)` demonstrate how specific angular features help disentangle *LAYING* from the upright postures, although the overlap between *SITTING* and *STANDING* persists even in these specific projections.

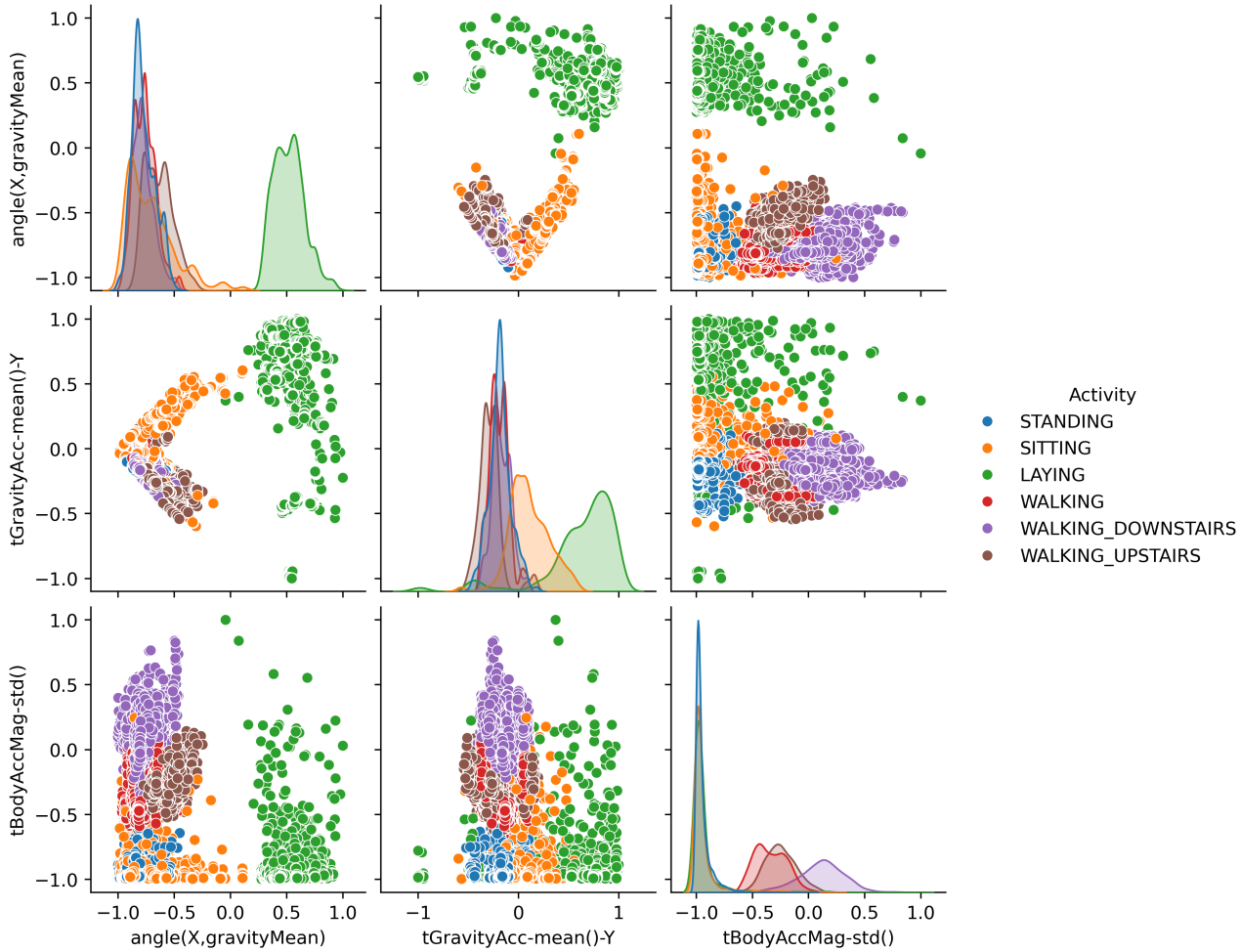


Figure 4: Pairplot analysis of selected features. The diagonal shows the kernel density estimation (KDE) for each activity. Note how `tBodyAccMag-std()` clearly separates static (peaked) from dynamic (spread) activities, while `angle` features assist in posture orientation.

2.2.4 Missing Values Simulation and Handling Strategies

Since the original dataset does not contain missing values, a controlled simulation was performed to test and validate different handling strategies. A small portion of the data was randomly removed to introduce artificial gaps.

Three distinct approaches were evaluated:

1. **Method 1: Complete Case Analysis (Delete rows).** This method involves removing any row containing at least one missing value.

   Limitation: An important limitation of this method was identified during the analysis. Given that the dataset is highly dimensional (561 features per sample), even a low probability of missing data per feature results in a very high probability that a row contains at least one null value. Mathematically, as the number of columns increases, the likelihood of retaining a complete row decreases exponentially.

   In practice, applying this method initially led to the removal of more than 90% of the dataset. Such drastic reduction makes this approach unsuitable for high-dimensional data, as it discards significant amounts of potentially valuable information without determining which features are actually essential for the neural network.

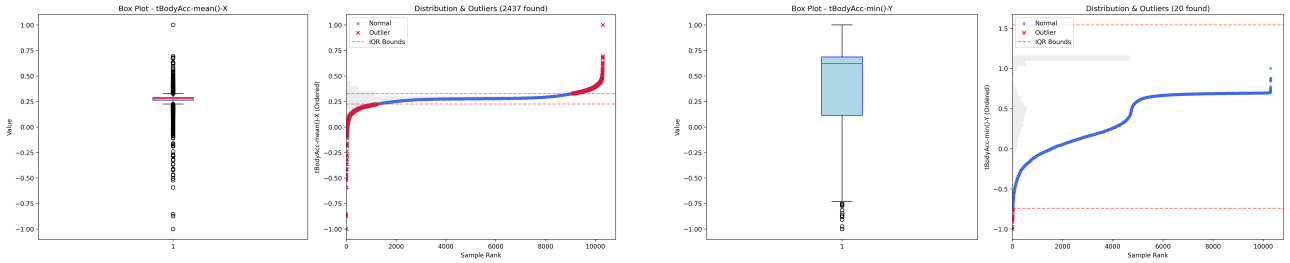
   Note: For the purpose of this practical exercise, the probability of random removal was artificially reduced to preserve a sufficient number of samples for the code to execute, but the method remains conceptually flawed for this specific dataset structure.

2. **Method 2: Statistical Imputation.** Missing values were filled using central tendency measures (mean, median, or mode). This method preserves sample size and is computationally efficient.

3. **Method 3: KNN Imputation.** A K-Nearest Neighbors approach was used to estimate missing values based on the similarity of samples in the feature space. This is generally more accurate than simple statistical imputation as it accounts for local data structure.

2.2.5 Outlier Analysis and Winsorization

The dataset exhibits a significant number of extreme values across various features. As shown in Figure 5, the box plots reveal numerous data points falling outside the interquartile range (represented by the circles beyond the whiskers).



(a) Outliers in tBodyAcc-mean()-X

(b) Outliers in tBodyAcc-min()-Y

Figure 5: Box plots and value distributions showing the presence of extreme values (outliers) in key accelerometer features.

Directly deleting rows containing these outliers would essentially replicate the problem encountered with missing values: it would remove a significant portion of the training data, potentially introducing bias or reducing model generalization.

Selected Strategy: Winsorization

To address this without discarding data, we applied a *Winsorization* technique. This method limits the influence of outliers by capping values at specified percentiles (e.g., 1^{st} and 99^{th} percentiles). Extreme values are replaced by the maximum or minimum thresholds defined by these percentiles. This approach reduces noise and stabilizes the training process while preserving the original dataset size.

2.2.6 Feature Correlation Analysis and Multicollinearity Reduction

To optimize model performance, a rigorous correlation analysis was performed. This step is particularly critical for linear models like **Logistic Regression**, which assumes independence among predictor variables to maintain stability.

In high-dimensional datasets derived from signal processing, *multicollinearity* is a pervasive issue. When two or more features are highly correlated ($r \approx 1.0$), they provide redundant information. For a linear classifier, this creates unstable coefficient estimates (high variance), making it difficult to interpret feature importance and causing the model to overfit to noise rather than learning robust patterns.

In the context of Human Activity Recognition (HAR), this redundancy arises from two primary sources:

1. **Mathematical Derivation:** Features such as mean, max, and energy calculated from the same sliding window often move in unison. For example, if the signal amplitude increases, both the standard deviation and the maximum value will rise simultaneously.

2. **Physical Coupling:** Body movements typically involve coordinated actions across multiple axes (X, Y, Z). A walking motion, for instance, generates simultaneous acceleration patterns in both vertical and forward directions, leading to natural correlations between sensors.

Step 1: Accelerometer Internal Correlation We began by analyzing features derived solely from the accelerometer. As hypothesized, variables generated from the same raw signal exhibit extreme redundancy. Figure 6 displays a distinct "block structure" of correlation.

For instance, the *Signal Magnitude Area (SMA)* and *Energy* features are mathematically dependent on the standard deviation and mean of the axes, creating clusters of near-perfect correlation ($r > 0.95$). Feeding all these variations into the model essentially duplicates the input signal, confusing the solver during gradient descent.

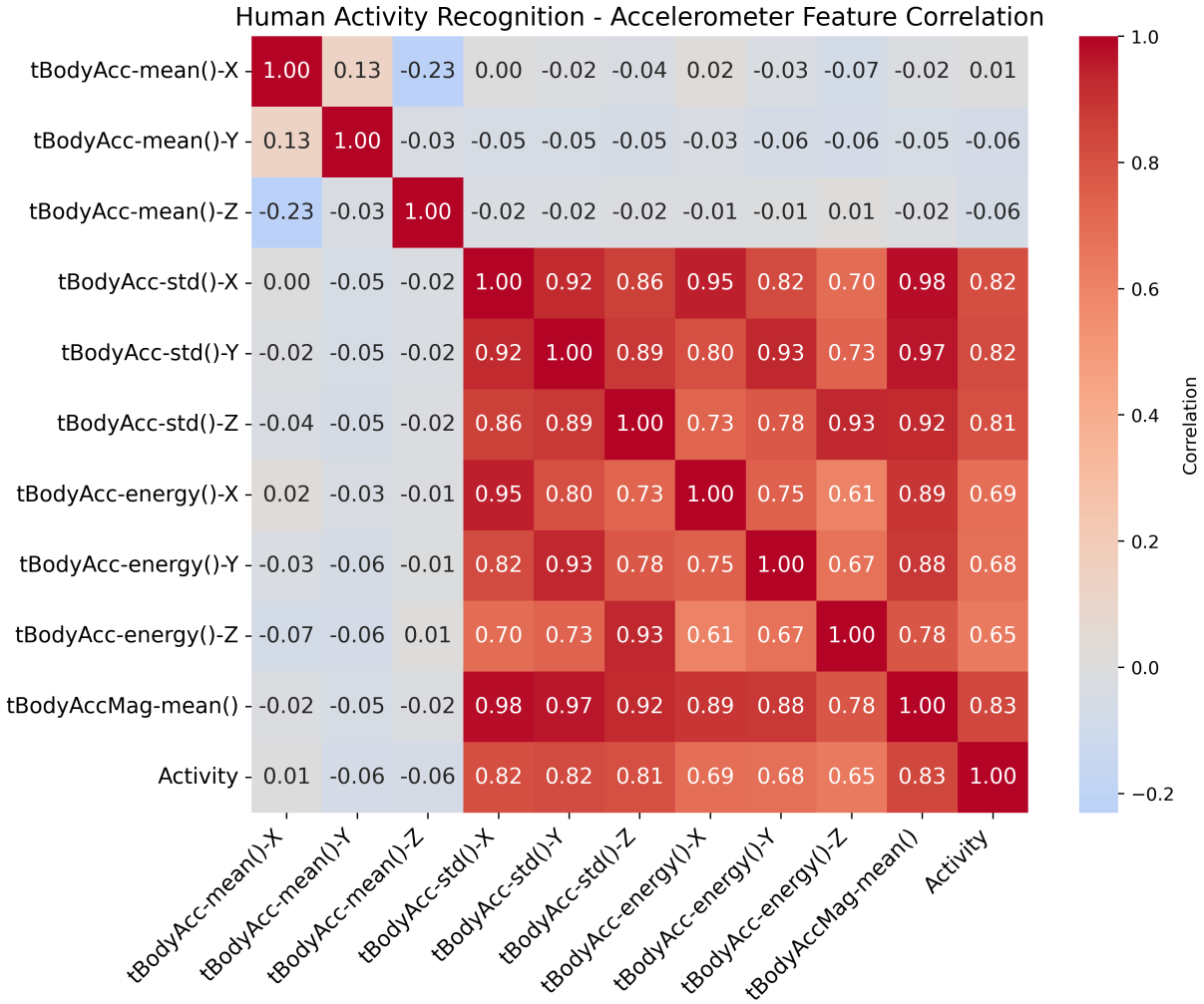


Figure 6: Correlation Matrix (Accelerometer Only). The dense red squares highlight strong internal redundancy caused by mathematical dependencies between derived features.

Step 2: Cross-Sensor Correlation Next, we expanded the analysis to include Gyroscope features (Figure 7). While we anticipated some correlation between sensors (since body movement affects both), the visualization confirms that simply adding more raw variables increases the dimensionality without necessarily adding independent information. The strong collinearity persists, reinforcing the need for dimensionality reduction.

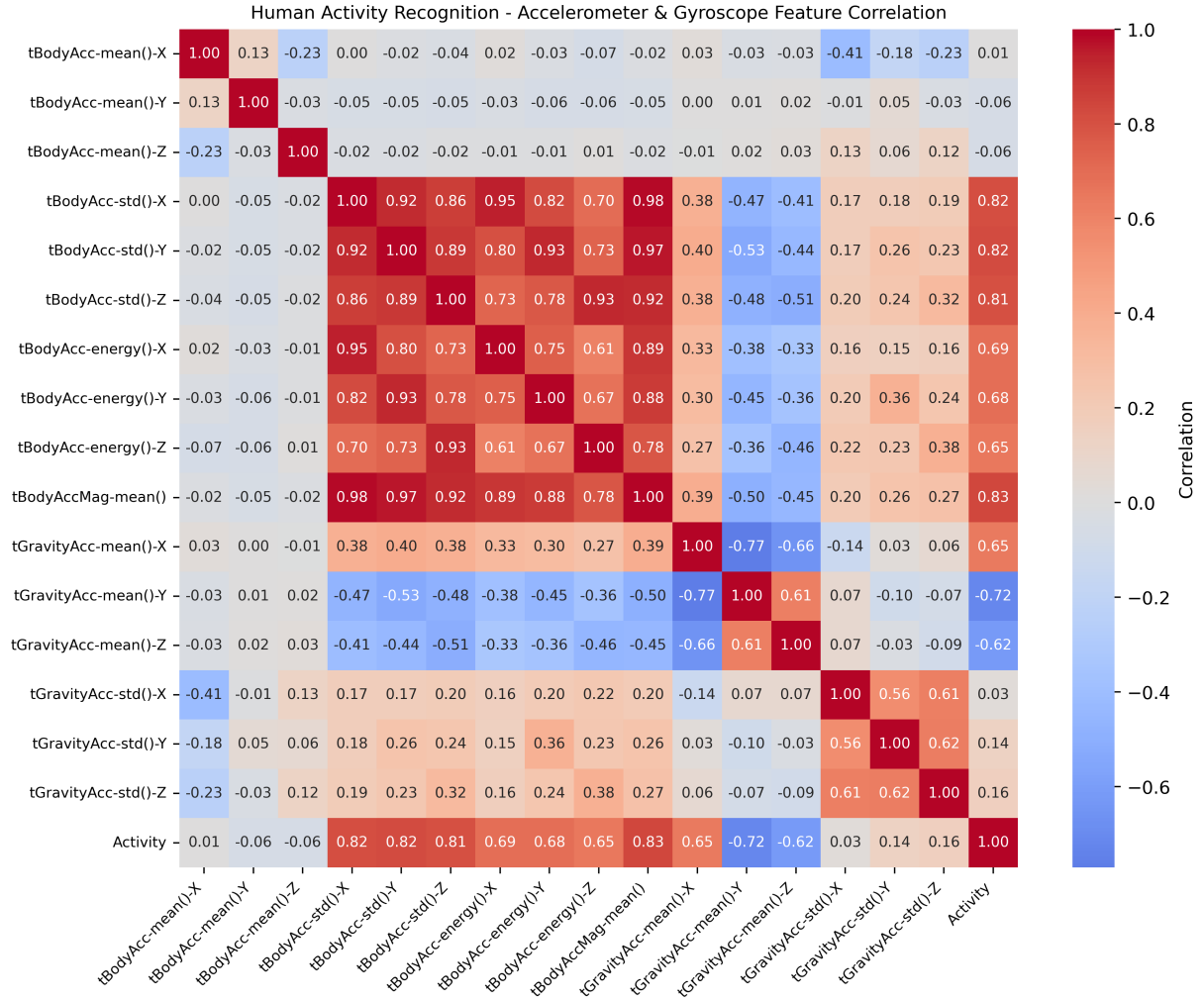


Figure 7: Correlation Matrix (Accelerometer + Gyroscope). Adding more sensors expands the redundancy if feature selection is not applied.

2.2.7 Feature Selection Strategy

Instead of using "black box" dimensionality reduction techniques like PCA, which obscure the physical meaning of the data, a manual selection strategy was employed. We selected **25 key features** to minimize redundancy while maximizing coverage of physical phenomena.

Figure 8 shows the correlation matrix of this final subset.

Analysis of Remaining Correlations: While the massive blocks of redundancy seen in the raw dataset have been eliminated, you will notice a specific cluster of high correlation remaining (variables related to *Gravity* and *Angle*). This retention is intentional and necessary. These features describe the device's orientation relative to the ground. Since the three spatial axes (X, Y, Z) are physically coupled when the device rotates, their correlation is not "noise" but a fundamental characteristic of the posture. Preserving this structure is crucial for the model to accurately distinguish between static activities like *SITTING* (vertical orientation) and *LAYING* (horizontal orientation).

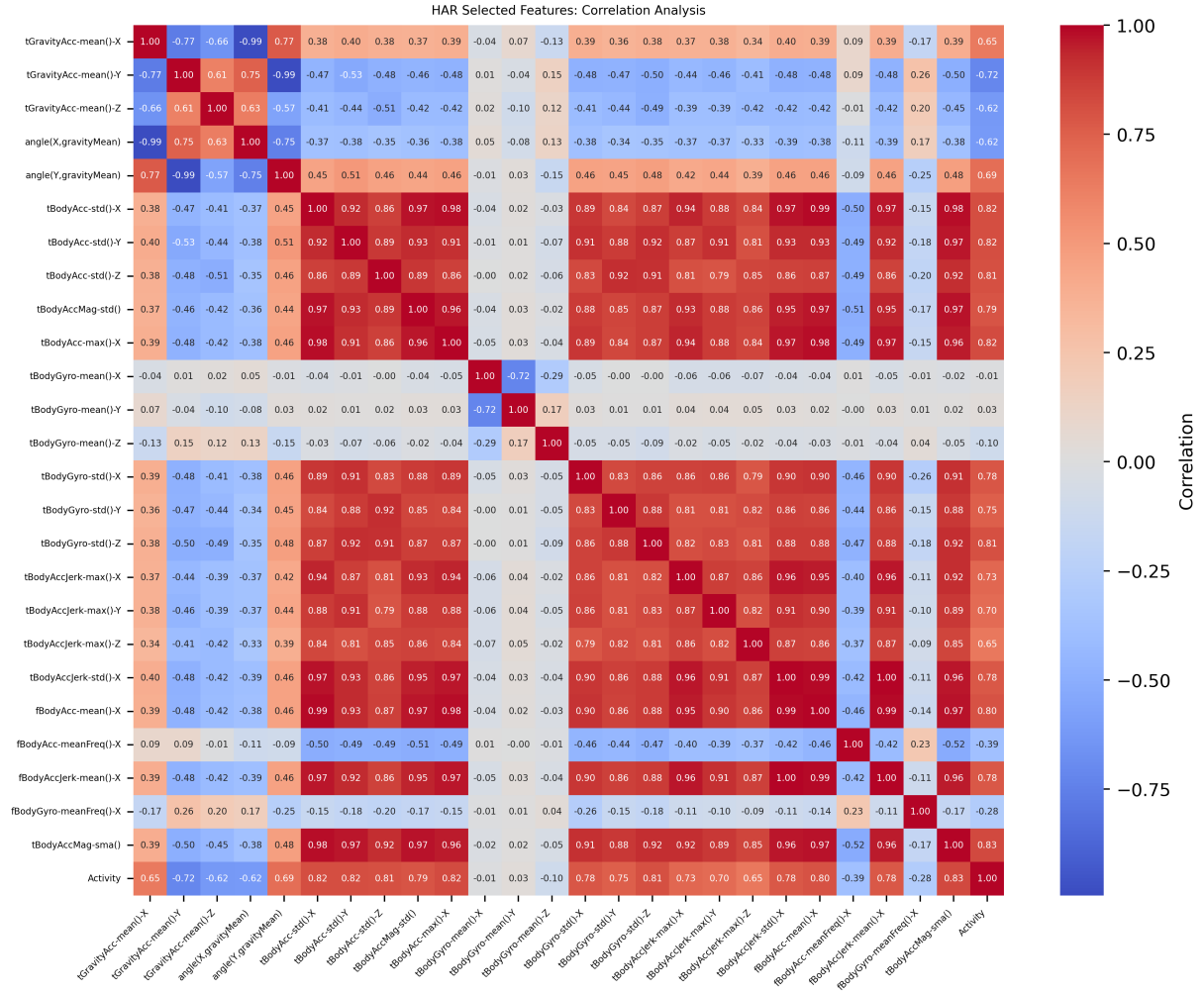


Figure 8: Correlation Matrix of the 25 Selected Features. Note that while overall redundancy is reduced, the Gravity/Angle cluster (top-left) is preserved to maintain posture information.

The final feature set covers the following distinct categories, each serving a specific classification role:

1. **Gravity and Angles (Posture):** *Role: Distinguishing static states.* Features such as `tGravityAcc-mean` and `angle()` provide the direct orientation of the smartphone. Without these correlated features, the model would fail to separate *STANDING* from *LAYING*.
    - `tGravityAcc-mean()-X/Y/Z`
    - `angle(X,gravityMean)`, `angle(Y,gravityMean)`
2. **Body Accelerometer (Intensity):** *Role: Separating static vs. dynamic states.* High variance (`std`) in these features indicates active movement (Walking), while low variance indicates rest.
    - `tBodyAcc-std()-X/Y/Z`, `tBodyAccMag-std()`
3. **Gyroscope (Rotation Patterns):** *Role: Distinguishing dynamic variants.* Essential for activities that have similar acceleration but different rotational signatures, such as the torso rotation difference between *WALKING* and *WALKING_UPSTAIRS*.
    - `tBodyGyro-mean()-X/Y/Z`, `tBodyGyro-std()-X/Y/Z`
4. **Jerk Signals (Sudden Changes):** *Role: Detecting transitions.* Derivatives of acceleration help identify the sharp impacts characteristic of footfalls or sudden stops.

- `tBodyAccJerk-max()-X/Y/Z`, `tBodyAccJerk-std()-X`

5. **Frequency & Global Intensity (Rhythm):** *Role: Capturing periodicity.* Frequency domain features help distinguish rhythmic activities (continuous walking) from sporadic movements.
   - `fBodyAcc-mean()-X`, `fBodyAcc-meanFreq()-X`
   - `tBodyAccMag-sma()`, `tBodyAcc-max()-X`

2.3 Experimental Design and Dataset Variations

To rigorously evaluate the impact of different preprocessing strategies on model performance, a comprehensive comparative analysis was conducted. We generated **24 distinct variations** of the dataset based on three axes of experimentation:

1. **Missing Value Strategy (4 levels):**
   - *Original:* No missing values (Baseline).
   - *Method 1:* Row deletion (High data loss scenario).
   - *Method 2:* Statistical Imputation (Mean/Median).
   - *Method 3:* KNN Imputation (k-Nearest Neighbors).
2. **Feature Space Transformation (3 levels):**
   - *Full Feature Set:* All 561 original variables (High dimensionality).
   - *Selected Features:* The subset of 25 non-redundant variables identified via correlation analysis (Domain-knowledge reduction).
   - *PCA Encoding:* Principal Component Analysis applied to project the high-dimensional data into a lower-dimensional orthogonal space, maximizing variance retention while reducing noise.
3. **Outlier Handling (2 levels):**
   - *Raw:* No outlier treatment applied.
   - *Winsorized:* Extreme values capped at 1^{st} and 99^{th} percentiles.

The combination of these parameters (4 imputation $\times$ 3 feature sets $\times$ 2 outlier strategies) results in 24 unique datasets. Each dataset undergoes the same training and evaluation pipeline to ensure strict comparability.

2.4 Train-Test Split

For each of the 24 dataset variations, a stratified split was performed to partition the data into training and testing sets. Stratification was strictly enforced to preserve the class distribution percentages observed in the Exploratory Data Analysis phase.

Table 2: Data Split Configuration

Set	Approx. Samples	Percentage
Training Set	~8,239	80%
Test Set	~2,060	20%

2.5 Classification Pipeline

The modeling workflow was implemented using a Scikit-learn `Pipeline` to prevent data leakage and ensure reproducibility. The pipeline consists of the following sequential steps:

2.5.1 1. Standardization

Since Logistic Regression is a distance-based algorithm (uses gradient descent), features must be on the same scale. A `StandardScaler` was applied to normalize the features:

$$z = \frac{x - \mu}{\sigma}$$

where μ is the mean and σ is the standard deviation of the training samples.

2.5.2 2. Model: Logistic Regression

Logistic Regression was selected as the baseline model due to its interpretability and efficiency. To optimize performance, a **Grid Search (GridSearchCV)** was implemented with 5-fold Cross-Validation.

The search space included:

- **Regularization Strength (C):** $[0.01, 0.1, 1, 5, 10, 15, 20]$. Smaller values specify stronger regularization to prevent overfitting.
- **Max Iterations:** $[500, 1000]$ to ensure convergence.

The best model for each dataset was selected based on the **Weighted F1-Score**, which accounts for the slight class imbalance.

Listing 1: Pipeline Implementation

```
pipe = Pipeline([
    ('scaler', StandardScaler()),
    ('model', LogisticRegression(random_state=42))
])

# Hyperparameter Grid
param_grid = {
    'model__C': [0.01, 0.1, 1, 5, 10, 15, 20],
    'model__max_iter': [500, 1000]
}
```

3 Results

3.1 Model Performance Comparison

Table 3 summarizes the performance (Weighted F1-Score) on the test set across the 24 experimental configurations.

Table 3: Complete Experimental Results (24 Datasets)

Category	Dataset Name	Samples	Feat.	Test F1
Full Features	full_original	10,299	561	0.9854
	full_method1 (Drop)	7,735	561	0.9838
	full_method2 (Median)	10,299	561	0.9859
	full_method3 (KNN)	10,299	561	0.9859
	outclr_full_original	10,299	561	0.9859
	outclr_full_method1	7,735	561	0.9838
	outclr_full_method2	**10,299**	**561**	**0.9864**
	outclr_full_method3	**10,299**	**561**	**0.9864**

Continued on next page...

Table 3 – continued from previous page

Category	Dataset Name	Samples	Feat.	Test F1
Selected (25)	selected_original	10,299	25	0.9388
	selected_method1	7,735	25	0.9424
	selected_method2	10,299	25	0.9398
	selected_method3	10,299	25	0.9389
	outclr_selected_original	10,299	25	0.9417
	outclr_selected_method1	7,735	25	0.9399
	outclr_selected_method2	**10,299**	**25**	**0.9432**
	outclr_selected_method3	10,299	25	0.9417
PCA (100)	pca_original	10,299	100	0.9656
	pca_method1	7,735	100	0.9696
	pca_method2	10,299	100	0.9655
	pca_method3	10,299	100	0.9646
	pca_outclr_original	10,299	100	0.9660
	pca_outclr_method1	7,735	100	0.9696
	pca_outclr_method2	**10,299**	**100**	**0.9665**
	pca_outclr_method3	10,299	100	0.9656

4 Analysis and Discussion

The comparative analysis reveals a clear performance hierarchy based on feature density:

Full Features > PCA (100) > Selected Features (25)

4.0.1 1. Feature Engineering: Quantity vs. Quality

- **Full Features (Best Performance, F1 ≈ 0.986):** The complete set of 561 variables consistently outperformed all other configurations. This suggests that despite the high multicollinearity, the "tail" of the information (variables often discarded as noise) contains critical signals for distinguishing the most difficult classes, such as *SITTING* vs. *STANDING*.
- **PCA (Middle Ground, F1 ≈ 0.967):** Using the top 100 principal components yielded a result ∼2% lower than the full set. This indicates that while PCA captures the majority of the variance, the remaining 461 components are not purely noise but contain subtle discriminative information. However, PCA significantly outperformed the Manual Selection method (∼0.967 vs ∼0.943), proving that 25 variables are insufficient to capture the full complexity of human motion.
- **Selected Features (Efficiency, F1 ≈ 0.943):** Manual selection resulted in the lowest accuracy but the highest efficiency (95% dimensionality reduction). This trade-off is acceptable for low-power embedded systems where computational resources are scarce.

4.0.2 2. Missing Value Strategies: Imputation vs. Deletion

The experiment highlighted a critical flaw in row deletion strategies for high-dimensional data:

- **Method 1 (Row Deletion):** This strategy caused a massive reduction in sample size ($N = 10,299 \rightarrow 7,735$). While the F1-score on the *remaining* data was high (e.g., 0.9696 in PCA), this is misleading. The model was trained on a biased, smaller subset, making it less robust to real-world variations.

- **Methods 2 & 3 (Imputation):** Both Median (Method 2) and KNN (Method 3) imputation allowed us to retain the full dataset size ($N = 10,299$). *Crucially, Method 2 (Median) performed equally to Method 3 (KNN).* Since Median imputation is computationally instant ($O(N)$) while KNN is expensive ($O(N^2)$), **Median Imputation is the superior choice** for this pipeline.

4.0.3 3. Impact of Outlier Handling

The application of Winsorization (denoted as `outclr`) provided a consistent, albeit small, performance boost across all feature categories:

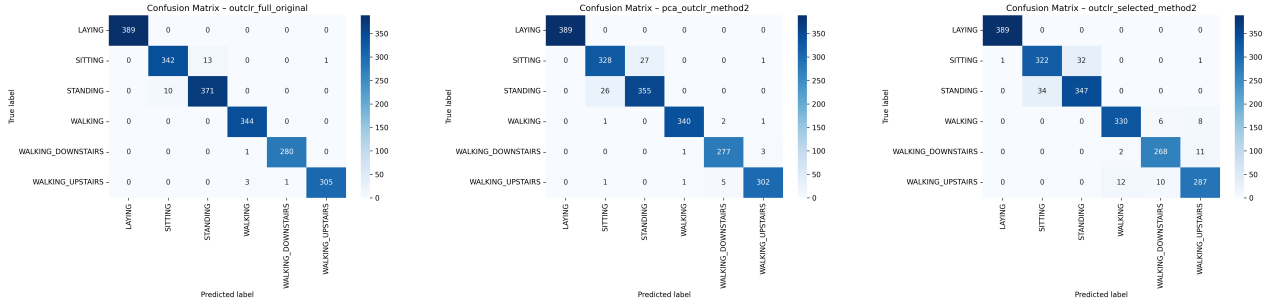
- **Full Features:** 0.9859 → 0.9864
- **Selected Features:** 0.9398 → 0.9432 (Significant improvement)
- **PCA:** 0.9655 → 0.9665

This confirms that extreme values in accelerometer data (likely caused by sensor noise or device impacts) destabilize the Logistic Regression gradients. Capping these values stabilizes the decision boundary without discarding valuable data points.

Final Recommendation: The optimal pipeline for maximum accuracy is **Full Features + Median Imputation + Winsorization**.

4.0.4 Error Analysis via Confusion Matrices

To visually inspect the classification errors and understand *where* the reduced models lose precision, Figure 9 displays the confusion matrices for the best performing model in each category.



(a) Full Features (561) (b) PCA (100) (c) Selected (25)

Figure 9: Comparison of Confusion Matrices. Note the significant increase in misclassification between SITTING and STANDING in the reduced feature set (c).

Visual Interpretation of Errors:

1. **High Accuracy for Dynamics:** All three models distinguish the *LAYING* activity perfectly (100% accuracy), as seen in the first row. Dynamic activities (Walking variants) also show distinct separation across all models, proving that 25 features are sufficient for basic motion detection.
2. **The "Confusion Zone" (Sitting vs. Standing):** The critical difference lies in the center of the matrices (indices 1 and 2):
   - **Full Features (a) & PCA (b):** These models show minimal confusion. For instance, the Full model has very few off-diagonal elements between SITTING and STANDING.
   - **Selected Features (c):** The reduced model struggles significantly here. As seen in Figure 9c, there is a visible cluster of errors: **34 Sitting samples were misclassified as Standing** and **32 Standing samples as Sitting**.

This confirms that while the 25 manually selected features capture general kinematics, they lack the subtle gravity/orientation nuances required to perfectly distinguish between the two upright static postures.

5 Conclusions

The complete Machine Learning workflow for human activity recognition was successfully implemented.

- **Objectives achieved:** The six activities were classified with overall accuracy above 98%.
- **Main finding:** Outlier handling via winsorization combined with data imputation is the most effective strategy for this dataset, outperforming sample removal.
- **Final model:** Logistic Regression ($C = 1$) trained on the full, imputed, and winsorized dataset (F1-Score: 0.9864).
- **Key insights:** Manual feature selection improves interpretability but sacrifices accuracy compared to using all available features with regularization.
- **Future improvements:** Exploring non-linear models or neural networks could help resolve ambiguity between static classes such as SITTING and STANDING.

As future work, non-linear models (such as Random Forest or SVM with RBF kernel) could be explored to improve discrimination between static classes (SITTING vs STANDING), which exhibited the highest error.