THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

COMP4702 Report

Author: Cao Quoc Thang Hoang

Student ID: 47594876

A report submitted for COMP4702: Exploring Machine Learning Approaches for Classifying Table Tennis Swing Patterns at The University of Queensland in Semester 1 - 2025

Contents

List of Figures

List of Tables

Chapter 1

Overview

This report presents a continuation of research on the *TTSWING* dataset [1], investigating the use of supervised machine learning techniques to classify health-related physical factors—specifically **age**, **height**, and **weight**—based on swing motion data from healthy players. The aim is to establish performance benchmarks that may define a baseline for what constitutes a *healthy status* among players with similar swing characteristics.
Rather than detecting specific health anomalies, this study focuses on analysing the capability of various supervised models to learn and distinguish these physiological attributes from sensor-derived features. This analysis is further extended by evaluating and ranking feature importance to understand which motion characteristics contribute most to model performance and predictive accuracy.
The central idea is that age, height, and weight—when accurately inferred from swing dynamics—can serve as key indicators of biomechanical efficiency, and deviations from these expected patterns may inform future anomaly detection systems. For instance:

- A model that can robustly classify a player's age group based on motion signals suggests the presence of age-sensitive biomechanical traits;
- Accurate prediction of height or weight from sensor data could indicate consistent kinematic patterns associated with these physical traits;
- Feature rankings that highlight metrics such as kurtosis, entropy, or RMS as important contributors may offer interpretability and domain insights into motion characteristics.

The primary objective of this report is to develop, implement, and evaluate supervised machine learning models to:

- Classify player age, height, and weight based on accelerometer and gyroscope-derived motion features;
- Compare model performance across three different supervised learning approaches;
- Identify and rank the most influential features that drive classification performance;
- Explore the implications of feature importance for understanding swing efficiency and physiological representation.

Rather than diagnosing medical conditions, the study focuses on whether consistent patterns in swing motion can be used to define health-related baselines across players, enabling future expansion into anomaly or health risk detection. The approach will be guided by exploratory data analysis (EDA) and refined through iterative model evaluation and interpretation.

Chapter 2

Data Cleaning and Exploratory Data Analysis

Before proceeding with this section, please note that the codebase for this project is available in my GitHub repository, which is linked in the appendix. I have intentionally avoided including the code directly in this report in order to maintain a focus on the logical structure rather than the implementation details.

2.1 Remove invalid and Null data

Although there are no null values in the dataset, however, there are some faulty data where values are not all 0 or not correct (e.g., age is ”???” or date is ”0/01/1900” in figure 2.1) which needed to be removed.

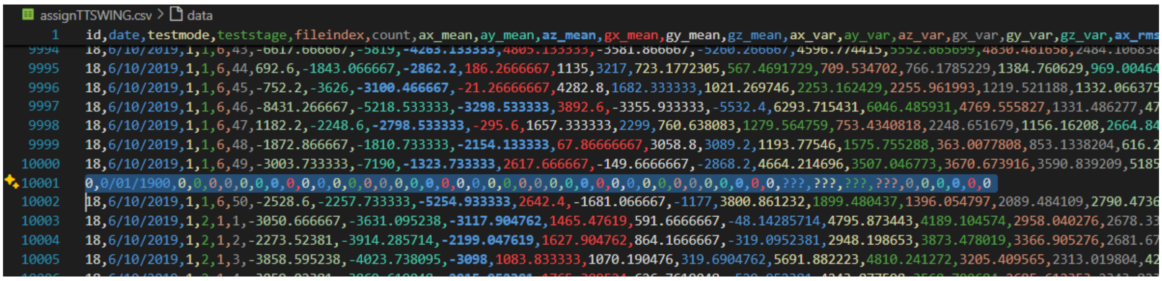

Figure 2.1: Example of faulty data in assignTTSWING dataset

2.2 Handle duplicates

This issue became apparent during the process of visualising the data for each participant in separate subplots, following a filtering procedure based on several criteria: test mode, test stage, file index, and recording date. For instance, after applying the following filters:

- `testmode == 0`
- `teststage == 0`
- `fileindex == 1`

- `date == '2020-02-18 00:00:00'`
- `handedness == 1`
- `holdRacketHanded == 1`

The resulting visualisation, as shown in Figure 2.2 where the vertical axis is the mean values and the horizontal axis is the count from 1 to 50, reveals that while the player with `id = 10` exhibits consistent and well-structured data (50 data points, which is equal to the max count), players with `id = 12` and `id = 80` display multiple accelerometer readings per axis (100 data points - double the count), resulting in overlapping or duplicated lines. This indicates potential data duplication or recording inconsistencies for those participants.

Figure 2.2: Duplicate values problem

In order to avoid losing crucial data but maintain the correctness of each trial pattern, any duplicate data is handled by splitting it and assuming that it is performed by extra participants. For example, the bad data above will be converted to 2.3, where players with id 102, 104 and 702 are newly created. This increases the total number of players from 92 to 108 players. Table 2.1 shows the original ID, new ID, and number of duplicate entries (1 entry is performed by 1 player in 50 counts).

Table 2.1: List of Player Duplicates and Their New IDs

Player ID	New ID	Number of Duplicates
6	98	10
10	102	1
12	104	6
15	107	7
17	109	8
30	122	6
32	124	8
42	134	7
46	138	8
48	140	18
50	142	12
64	156	7

2.3 Balancing data

The plot, 2.4, illustrates the slight imbalance of the dataset in terms of height, weight, and especially different age ranges. Specifically, although there are no rigid thresholds to determine if the dataset is imbalanced, the percentage of the young (labelled as "low") in the dataset is more than 50% of the total number of participants, whereas others are only at 27.6% and 18.6% and potentially lead to the bias towards the majority class, such as the young age range, and less sensitive to middle-age participants. However, if the oversampling technique is used, it may destroy the relationship and distribution between Age and other metadata such as participants' height and weight, which are already in good ratio. Moreover, from [1], the highest accuracy they can achieve for age classification is up to 90% indicates that the matter of imbalance is acceptable and not required to be handled.

2.4 Consideration of Principal Component Analysis (PCA) vs Manual Feature Selection

Within the scope of this dataset, Principal Component Analysis (PCA) proves valuable as a dimensionality reduction technique. It mitigates the complexity inherent in datasets containing more than 40 features and over 97,000 entries by identifying the most significant patterns of variance and transforming the original variables into a smaller, more manageable set of uncorrelated components. This transformation facilitates faster training times and reduces overfitting in certain models. However, the use of PCA introduces several trade-offs, including reduced interpretability of the principal components, potential information loss, and sensitivity to feature scaling and outliers.

Figure 2.5 presents the explained variance ratio across the first 34 principal components, derived from a comprehensive set of statistical features—namely mean, variance, root mean square (RMS), maximum, minimum, Fast Fourier Transform (FFT), power spectral density (PSDX), kurtosis, skewness, and entropy—computed from both accelerometer and gyroscope data. The figure illustrates that there are at least 21 rows to preserve all possible data (¿99%) and if we only take 5 components, we are going to lose approximately 4% of the total information.

However, the focus of this report is the explainability of the model after training (which features play an important role in the classification) and analysing the predicted output, so manual feature selection

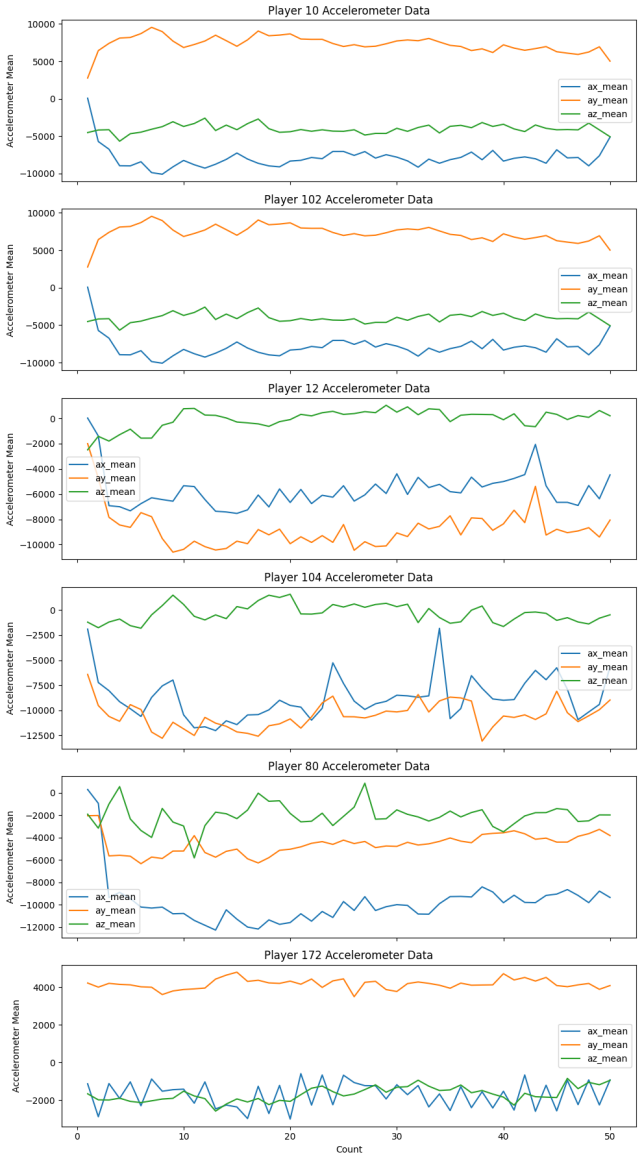

Figure 2.3: Data after handle Duplicate values problem

will be used to reduce the number of dimensions instead of Principal Component Analysis (PCA).

The number of selected components is determined based on the cumulative variance explained, ensuring that at least 95% of the original data variability is preserved. In this particular case, the threshold is met with $n_{\text{components}} = 4$. It is important to note that multiple PCA analyses are conducted depending on whether specific black-box features (denoted as `newv#`) are included or excluded from the feature set.

2.5 Correlation Visualizations

In exploring the relationship between motion features and demographic classifications, 3D scatter plots were used to visualise the separation of height classes (low, medium, and high) using the mean values of both accelerometer and gyroscope signals. As illustrated in Figures 2.6 and 2.7, the data does not exhibit distinct or well-separated clusters that correspond to height classes. Similar patterns of weak

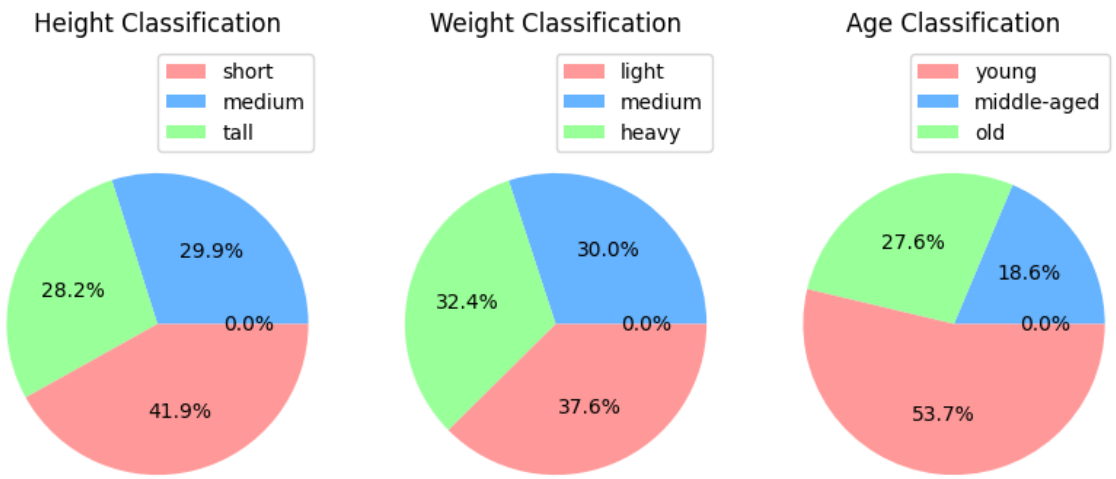

Figure 2.4: Distributions of Height, Weight and Age

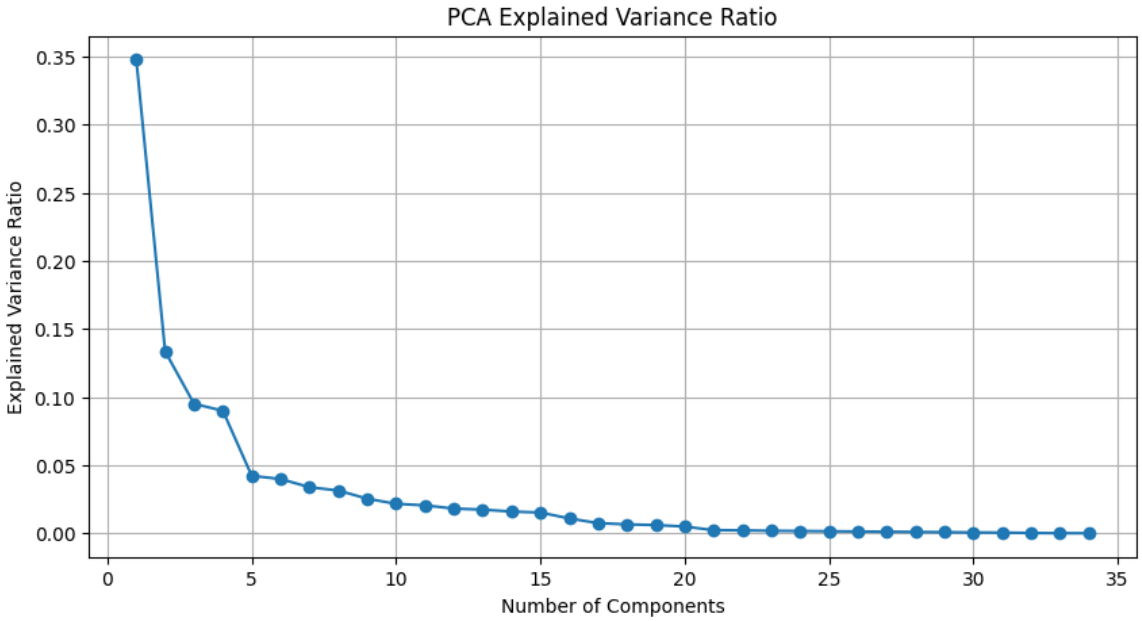

Figure 2.5: PCA Explained Variance Ratio of 34 components

separation are observed for other demographic variables, such as weight and age, when considering both their mean and variance.

These observations suggest that simple unsupervised methods—such as K-Nearest Neighbors (K-NN), which assume spherical or convex class boundaries—are unlikely to yield satisfactory performance. Instead, more sophisticated models that can capture complex, non-linear, and non-spherical distributions are required for effective unsupervised classification.

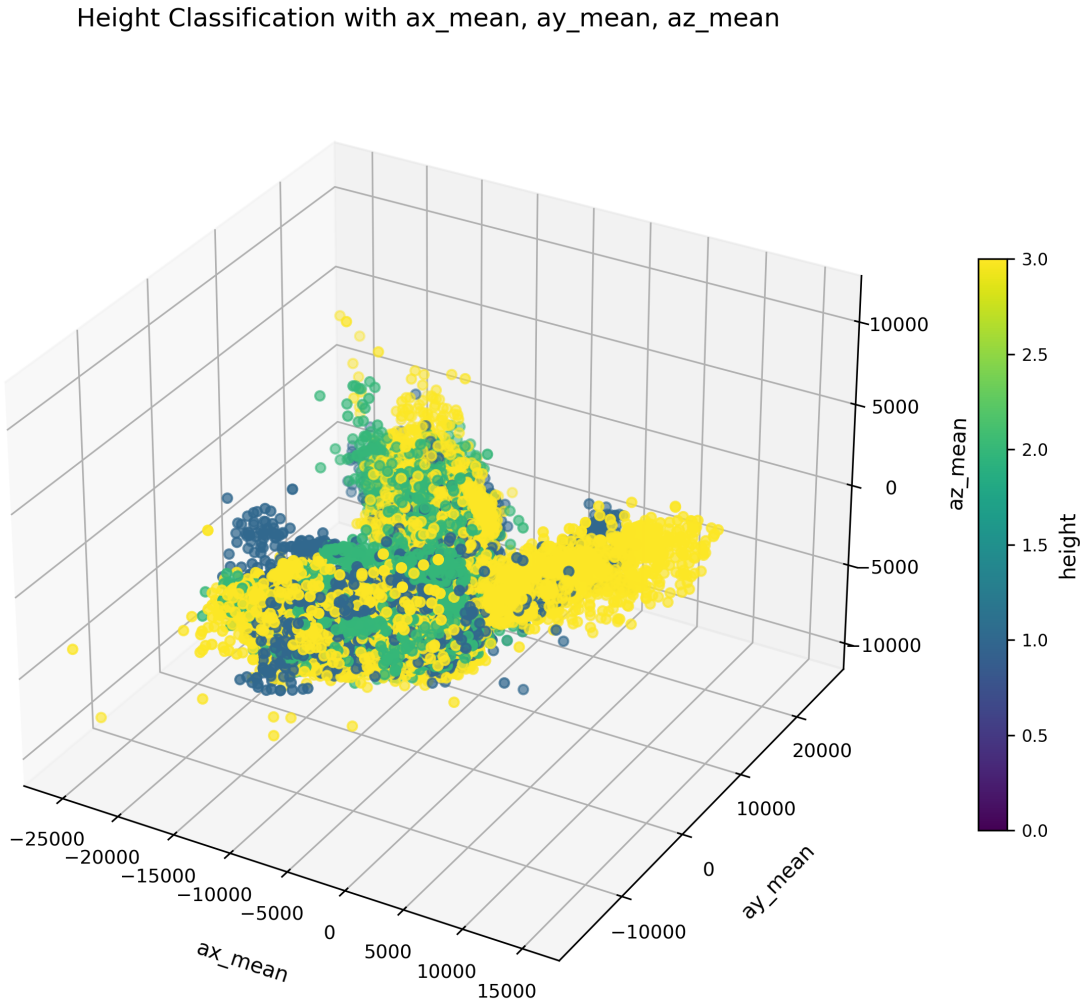

Figure 2.6: Height Classification with ax_mean, ay_mean, az_mean

2.6 Model Selection and Interpretability

As mentioned in the previous section, the most important criterion of the supervised learning approach, interpretability, is a key consideration. Explainable models not only provide insights into the most influential features affecting prediction accuracy but also support model validation and trust in real-world deployment. For this reason, interpretable algorithms such as K-NN, Decision Trees and Random Forests were initially prioritised. These models offer transparency in terms of feature importance, decision boundaries, and the reasoning behind each prediction—factors that are especially important in understanding how sensor-based motion data correlates with physical attributes such as height, weight, and age.

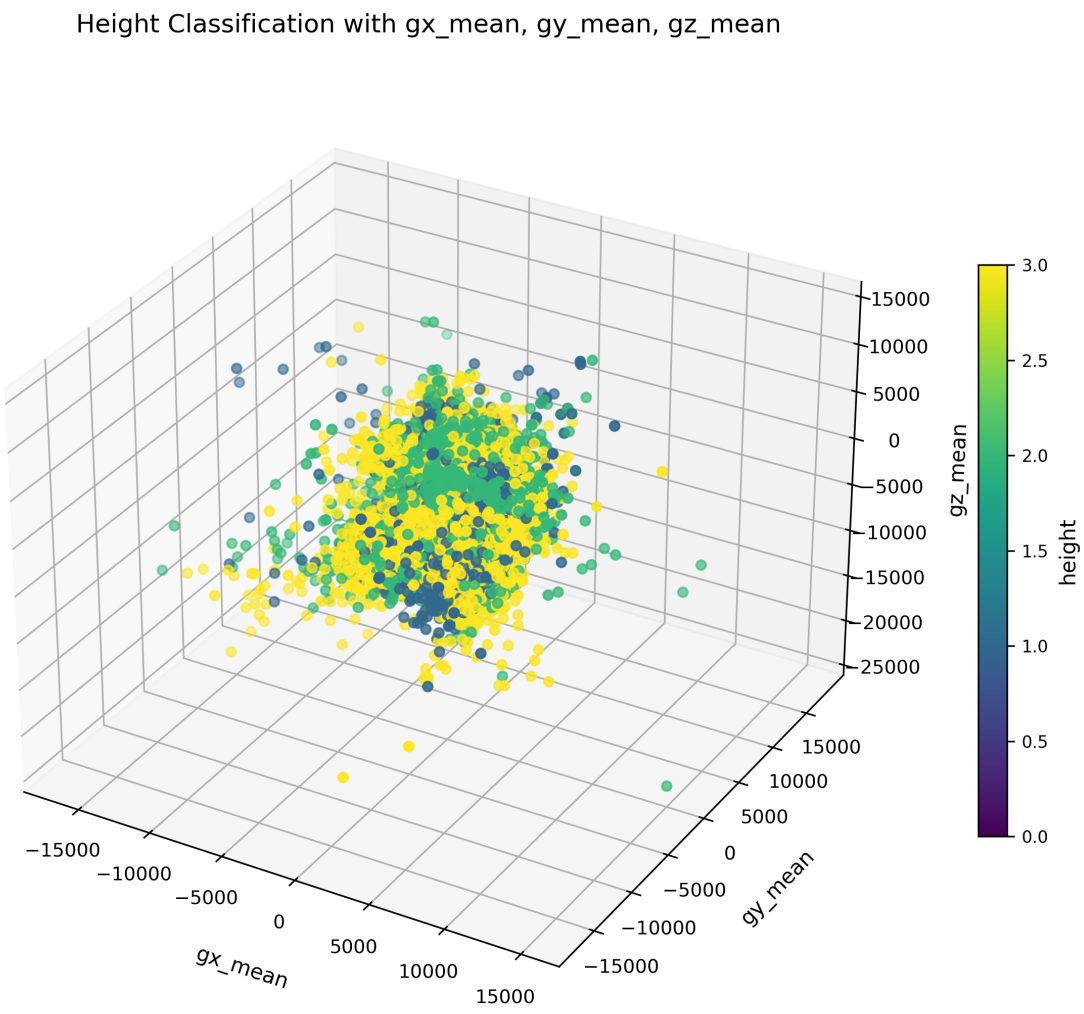

Figure 2.7: Height Classification with gx_mean, gy_mean, gz_mean

Chapter 3

Supervised Classification Algorithms

This chapter explores the application of supervised learning algorithms—specifically K-Nearest Neighbors (K-NN), Decision Trees, Random Forests—for classifying swing motion data in relation to player health status. These algorithms are selected for their demonstrated effectiveness in identifying patterns that may reflect deviations from expected physical performance benchmarks. The chapter details the training procedures, performance evaluation metrics, and typical outcomes observed when these models are applied to motion sensor datasets in health-oriented contexts.

3.1 K-Nearest Neighbors

K-Nearest Neighbors (K-NN) serves as an intuitive starting point for classification tasks due to its simplicity and transparency. The algorithm operates on the fundamental principle that similar instances tend to belong to the same class, making it inherently interpretable. For swing motion analysis, K-NN provides a straightforward approach to identifying abnormal patterns by comparing new observations with known examples in the feature space.

3.1.1 Training and Experimental Setup

This section outlines the experimental procedures conducted to evaluate and optimise the performance of the K-Nearest Neighbours (K-NN) model, with justifications for the chosen parameters, preprocessing steps, and evaluation strategies. The implementation followed the pseudocode in Method 2.1 from [2].

1. Data Preprocessing

To ensure that all features contribute equally to distance calculations, all input columns were standardised using the `StandardScaler` from `scikit-learn` [3]. Standardisation transforms the features to have zero mean and unit variance, which is crucial for distance-based models like K-NN that are sensitive to the scale of input data.

2. Train-Test Split

The dataset was split into training and testing subsets using a stratified 70:30 ratio. Stratification preserves the class distribution across both subsets, ensuring that the model is trained and evaluated on

representative samples. This split ratio balances the need for a sufficiently large training set with an adequately sized testing set to evaluate generalisation.

3. Hyperparameter Selection

The optimal number of neighbours k and the most appropriate distance metric were determined using grid search combined with 5-fold cross-validation:

- **Number of Neighbours (k):** Various values of k were tested to identify the one that yields the highest classification accuracy. A small k can lead to noisy predictions, whereas a large k may smooth over local structure. Cross-validation was used to select a value that balances bias and variance.

- **Distance Metrics:** Step out of simple Euclidean Distance in [2], multiple similarity metrics were evaluated:

  - **Euclidean Distance** (Minkowski with $p = 2$) [4] — the standard metric in K-NN.

$$D_{\text{Euclidean}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

  - **Manhattan Distance** (Minkowski with $p = 1$) — useful for high-dimensional data.

$$D_{\text{Manhattan}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

  - **Cosine Distance** — beneficial when magnitude differences between features are less meaningful.

$$D_{\text{Cosine}}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$$

  - **Haversine Distance** — appropriate for geospatial data.

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)$$
$$c = 2\arctan 2\left(\sqrt{a}\,, \sqrt{1-a}\right)$$
$$D_{\text{Haversine}} = R \cdot c$$

    where:

    * ϕ_1, ϕ_2 are the latitudes of the two points in radians,
    * $\Delta\phi = \phi_2 - \phi_1$,
    * $\Delta\lambda = \lambda_2 - \lambda_1$ is the difference in longitudes in radians,
    * r is the Earth's radius (mean radius = 6,371 km).

  The metric that produced the highest cross-validation accuracy was selected for final evaluation. Please note that all metrics above are specified from [5].

4. Cross-Validation Strategy

To assess the generalisability of the model, 5-fold cross-validation was employed. This technique involves partitioning the training data into five equal-sized folds, using four for training and one for validation in each iteration. The use of five folds offers a good trade-off between bias and variance in model performance estimation and avoids the computational overhead of more extensive splitting strategies.

6. Summary of Experiments

The following experiments were conducted:

- **Full-batch classification:** Used to identify the best combination of k and distance metric that yields the highest accuracy on the test set.
- **5-fold Cross-Validation:** Performed to evaluate the model's robustness and generalisation to unseen data.

3.1.2 Results and Evaluations

Full-Batch Classification Results

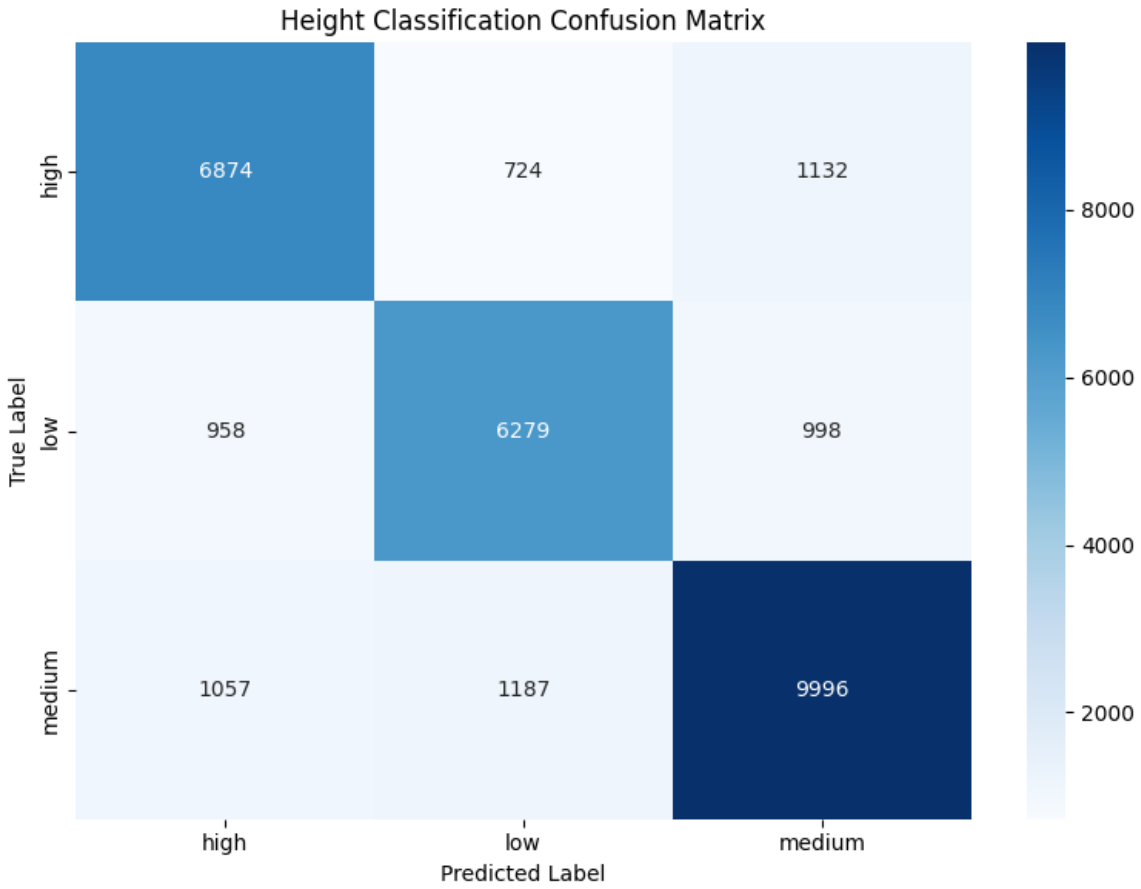

Figure 3.1: KNN Height Classification Confusion Matrix

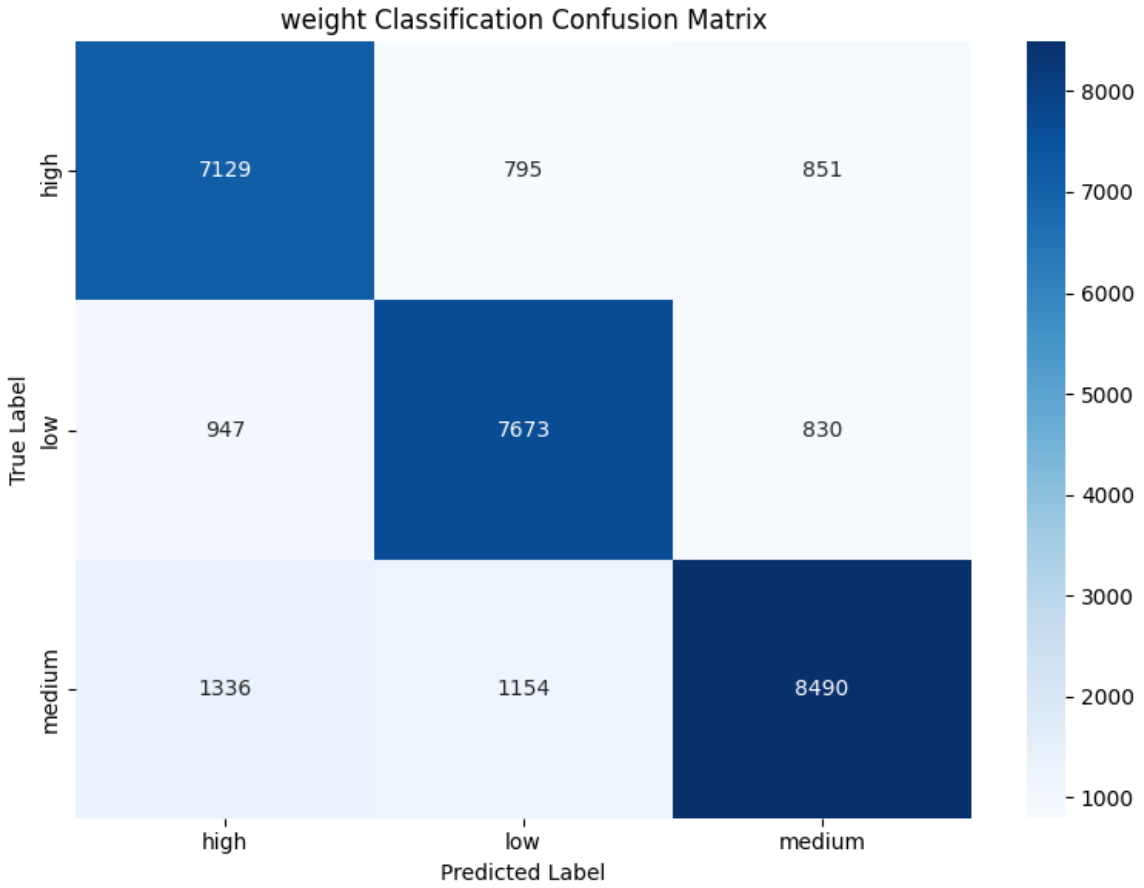

Figure 3.2: KNN Weight Classification Confusion Matrix

From the confusion matrices 3.1, 3.2 and 3.3, the K-NN classifier achieved test accuracies of **79.26%** (height), **79.75%** (weight), and **82.05%** (age). Age classification also yielded the best balanced precision (0.81), recall (0.79), and F1-score (0.80). Misclassification analysis for age revealed **5,243** incorrect predictions, with the most frequent confusions occurring between high-medium (1,422), low-medium (1,398), and medium-low (1,120) classes.

These moderately high results suggest the presence of data-related issues such as noise, class imbalance, overfitting/underfitting, and potentially irrelevant features or suboptimal model choices. Further diagnostic analysis is required to pinpoint the root causes.

Feature Importance and Correlation Analysis

Feature importance analysis reveals that for height prediction, features like `a_entropy`, `g_entropy`, and `az_mean` showed the highest individual correlations (Figure 3.4). These metrics capture motion variability and vertical acceleration, potentially relevant for estimating height.

However, their correlation coefficients were low ($\tilde{0}.1\%$), and using all features yielded significantly higher performance (88.05% accuracy), compared to using only derived sensor features (60.58%) or FFT features (40.44%). This suggests that weak correlations may still contribute meaningfully when combined in high-dimensional feature space.

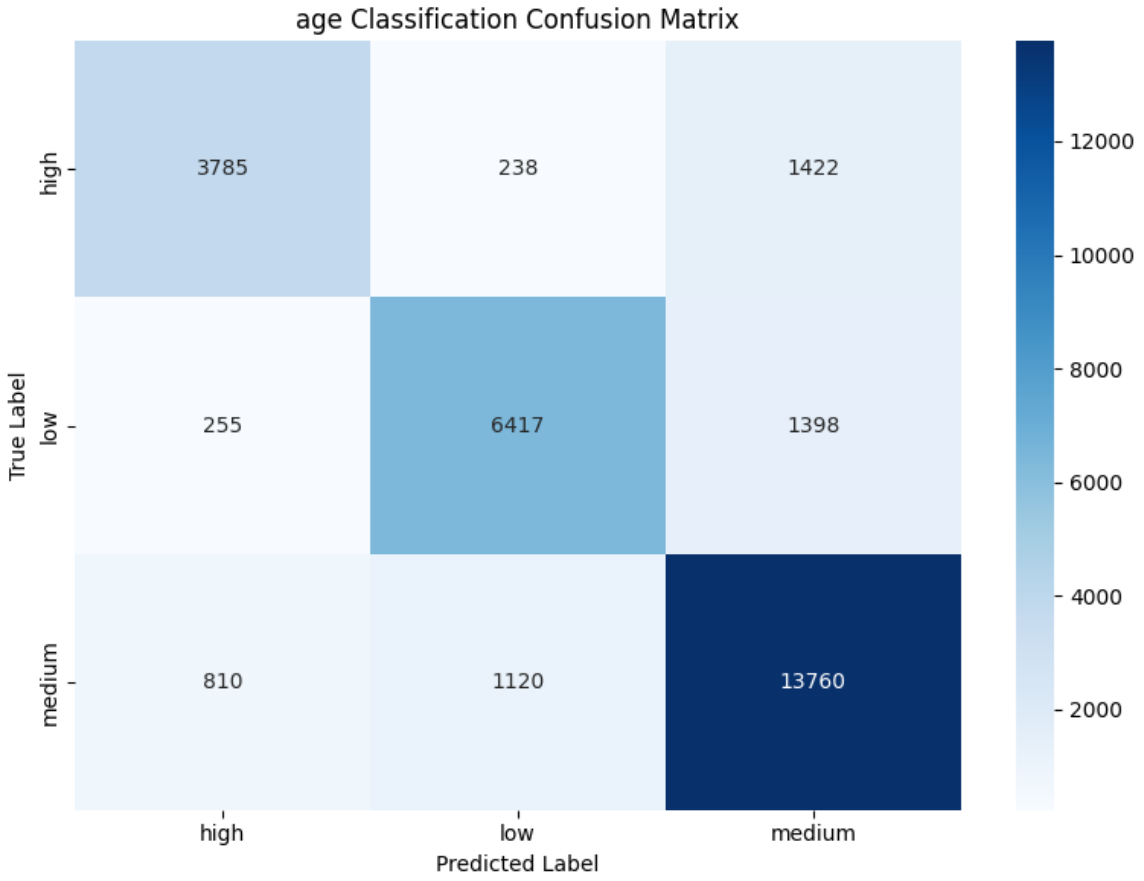

Figure 3.3: KNN Age Classification Confusion Matrix

Hyperparameter Tuning (Optimal k)

Although the dataset includes three classes per target, $k = 3$ does not yield peak performance (3.7, 3.8 and 3.9). Instead, accuracy improves from $k = 3$ to $k = 7$–8 before declining, especially for age prediction. This suggests overlapping feature distributions and hidden subgroups, with lower k values being too sensitive to noise.

5-Fold Cross-Validation and Generalisation Performance

The learning curves in Figure 3.10 show a **60% performance gap** between training and validation scores, indicating overfitting. Despite optimal hyperparameter tuning, the model struggles to generalise, likely due to high sensitivity to noise and limited representation in each fold. This suggests potential improvement through noise-resilient models like Decision Trees.

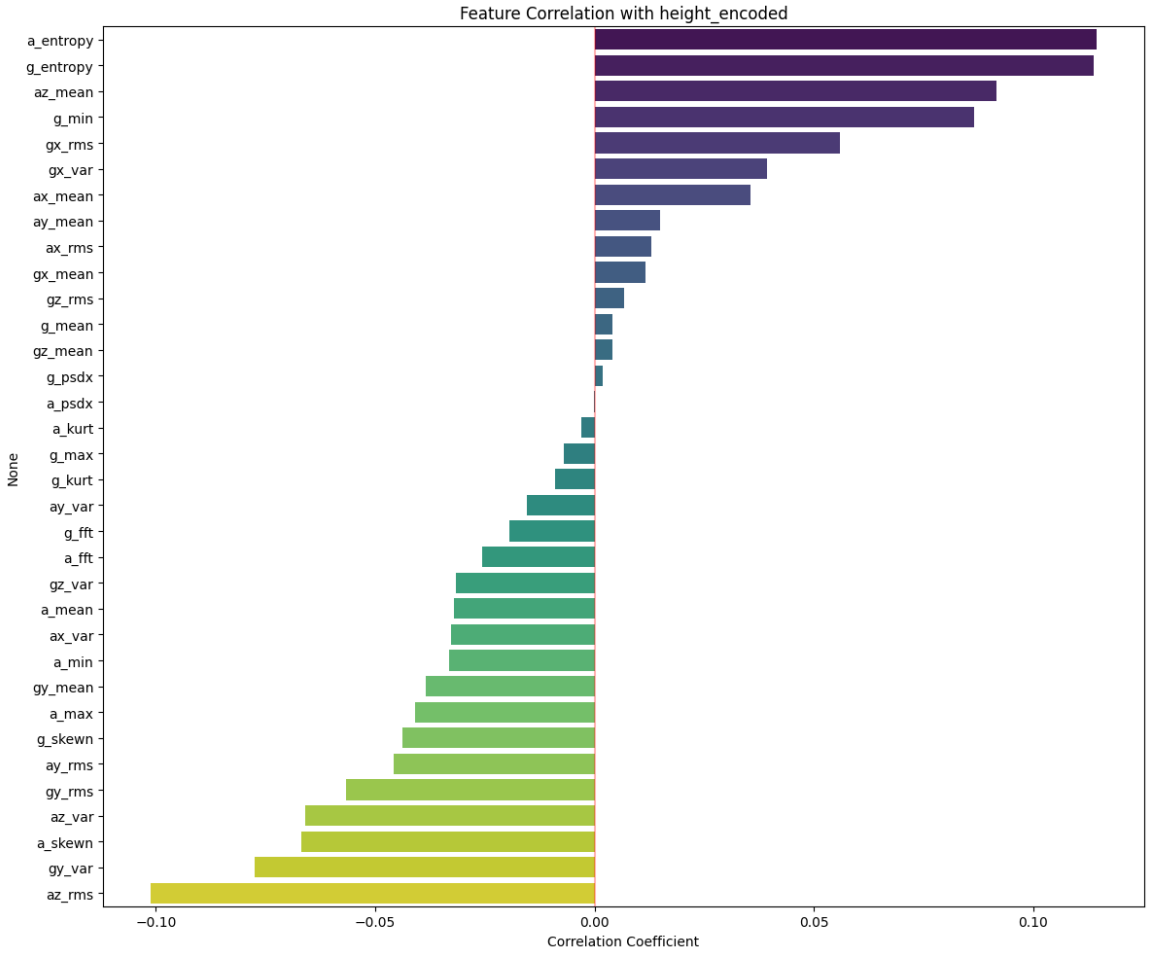

Figure 3.4: KNN Height Classification Feature Correlation Analysis

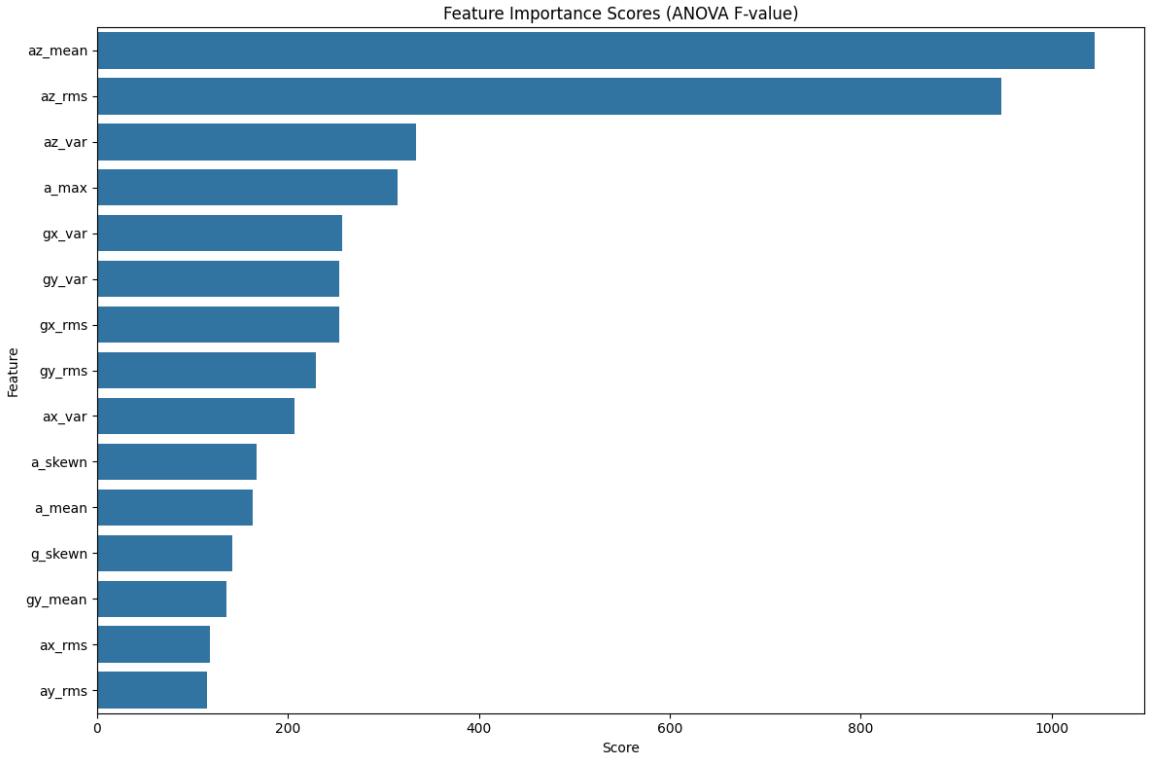

Figure 3.5: KNN Weight Classification Feature Importance Analysis

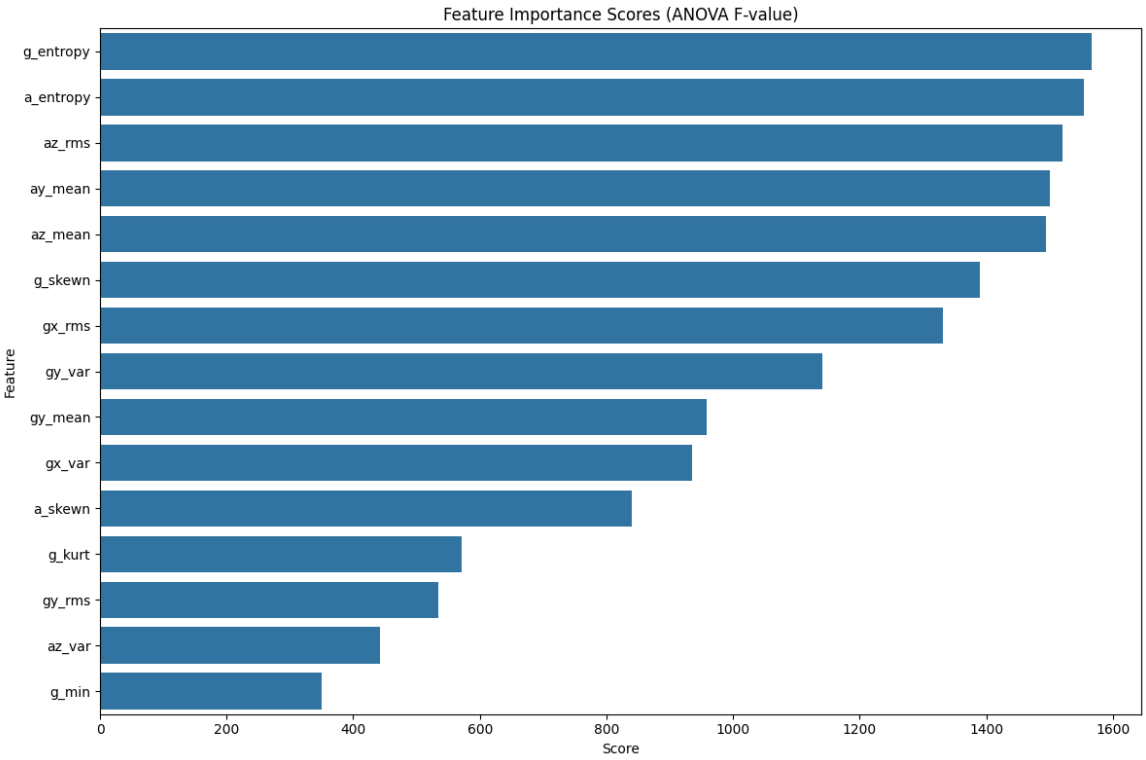

Figure 3.6: KNN Age Classification Feature Importance Analysis

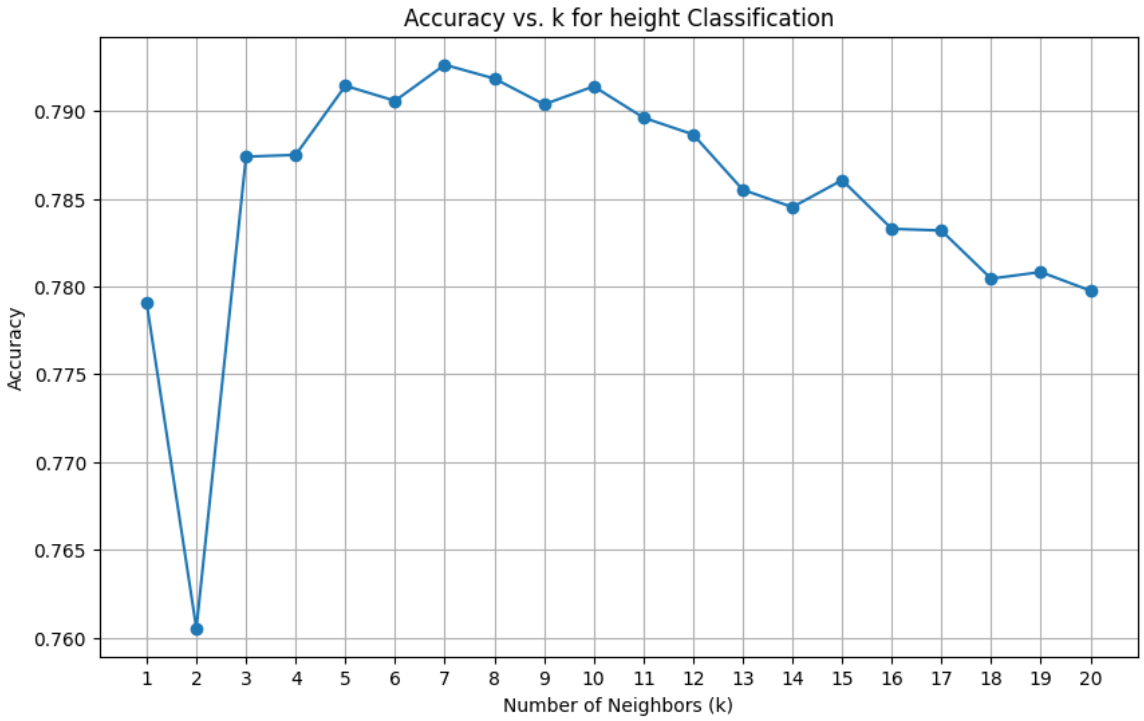

Figure 3.7: Accuracy of K-NN Model for Height Classification Across Different k Values

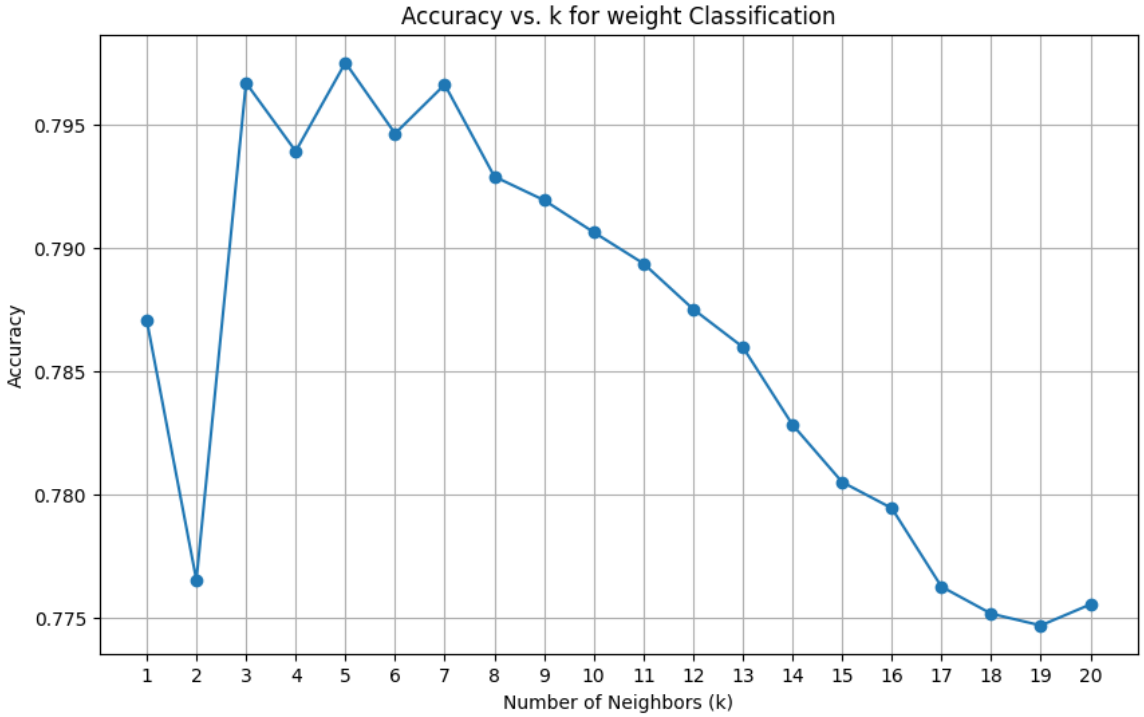

Figure 3.8: Accuracy of K-NN Model for Weight Classification Across Different k Values

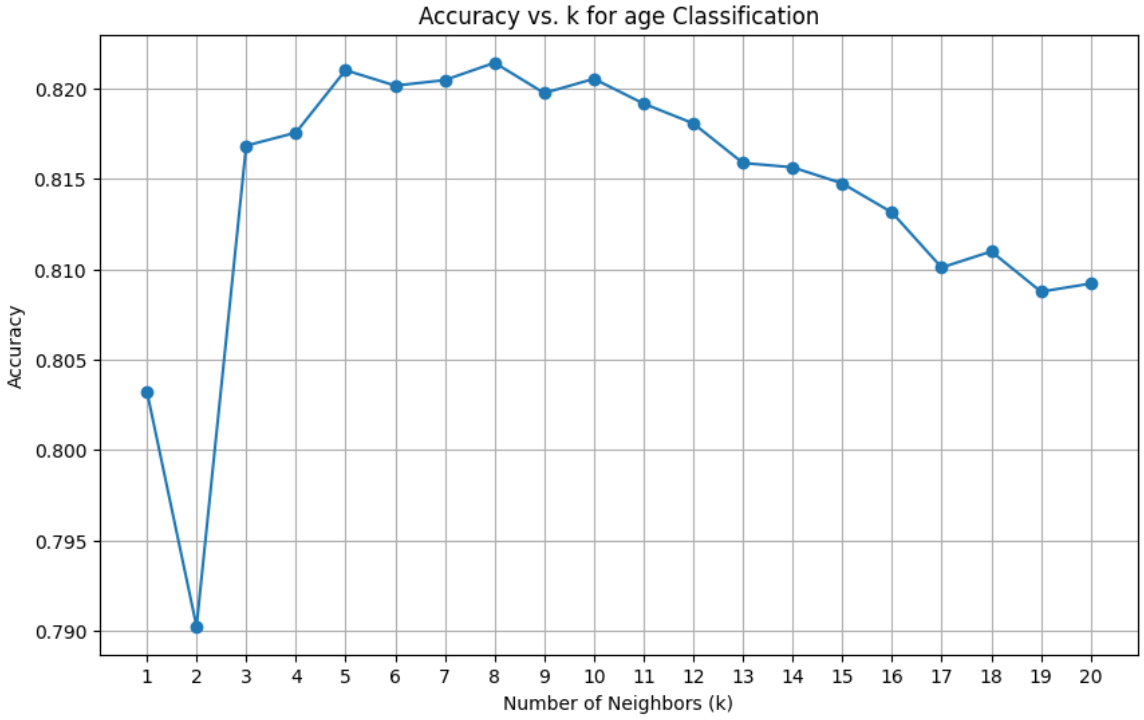

Figure 3.9: Accuracy of K-NN Model for Age Classification Across Different k Values

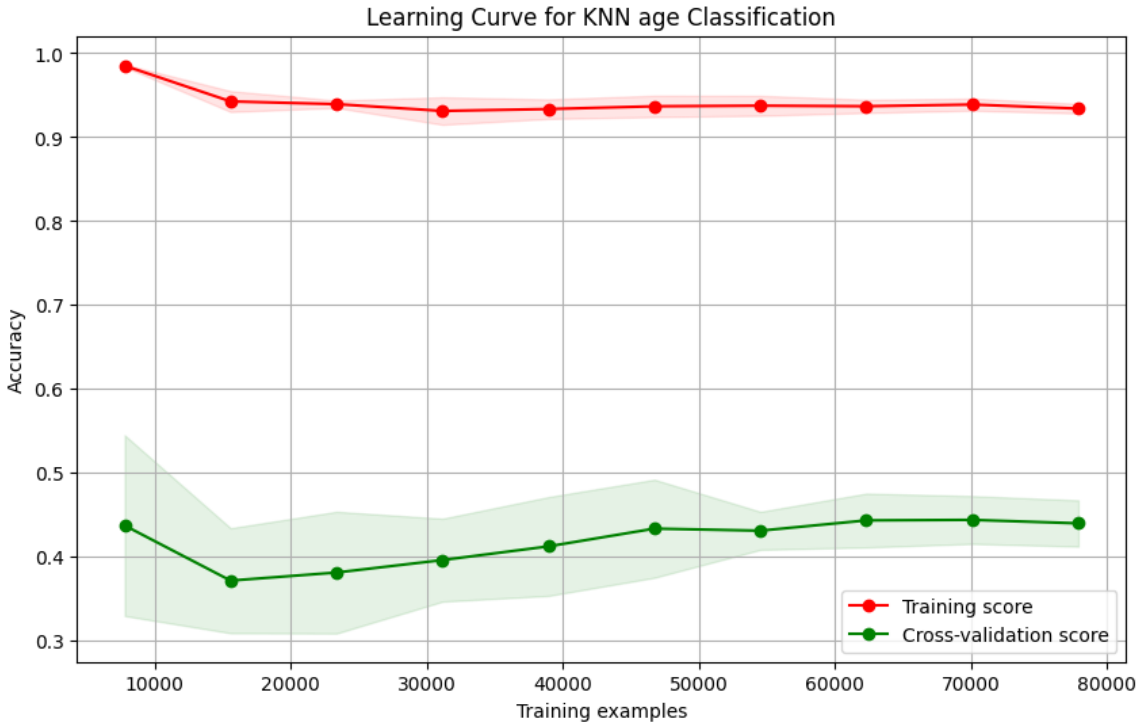

Figure 3.10: Performance Analysis Using Learning Curves (Full-Batch vs. 5-Fold)

3.2 Decision Tree

Decision Trees represent an advancement in both model complexity and explanatory power compared to K-NN. These hierarchical models partition the feature space through a series of interpretable if-then rules, making them particularly valuable in healthcare applications where transparency in decision-making is crucial. Also, Decision Trees capture non-linear relationships in the data while maintaining a clear visualisation of the classification process.

3.2.1 Training

The Decision Tree classifier [6] was trained using the CART (Classification and Regression Trees) algorithm, a popular method known for its interpretability and efficiency in handling both numerical and categorical features. The initial configuration employed *Gini impurity* as the splitting criterion.
To overcome the limitations observed in the K-NN model—specifically its sensitivity to noise, poor generalisation on unseen data, and dependence on local structure—a more robust training approach was adopted for Decision Tree classification. Unlike K-NN, which was initially trained on the full batch and later validated, the Decision Tree classifier was trained using **5-fold cross-validation** across all three classification targets: height, weight, and age. This strategy allows the model to be evaluated on multiple subsets of the dataset, reducing variance in performance estimates and helping to mitigate overfitting.
A comprehensive hyperparameter optimisation was conducted via grid search, targeting the following key parameters:

- **Splitting Criterion:** Three different criteria were evaluated:
  - *Gini impurity*, the default measure in CART, evaluates the probability of misclassification and is computationally efficient, formulated in Equation 2.7b [2].
  - *Entropy*, also known as *information gain*, considers the amount of information (in bits) required to classify the samples and is formulated in Equation 2.7c in [2].
  - *Log-loss* (logistic loss), inspired by probabilistic models, is introduced in [7] and better aligns with probabilistic decision boundaries for multi-class classification.
- **Maximum Tree Depth:** A range of values was explored: $\{3, 5, 7, 10, \text{None}\}$. Similar to the k parameter in K-NN, the tree depth governs the model complexity. Shallow trees may underfit, while overly deep trees risk overfitting. Including the `None` option (i.e., unlimited depth) allows exploration of the trade-off between learning capacity and generalisation.
- **Minimum Samples for Splitting and Leaf Nodes:** To further control model complexity and avoid overly specific partitions, I tuned:
  - `min_samples_split`: the minimum number of samples required to split an internal node,
  - `min_samples_leaf`: the minimum number of samples required to be at a leaf node.

  Values of $\{2, 5, 10\}$ were tested for both parameters. These thresholds affect how easily the tree grows; higher values enforce stricter branching conditions and reduce the risk of capturing noise.

This hyperparameter search was repeated independently for each of the three target labels. The goal was to identify configurations that not only yield the best cross-validated accuracy but also demonstrate

resilience to noise and irrelevant features—an area where K-NN struggled. By focusing solely on cross-validation (rather than full-batch fitting), the model selection process prioritised generalisation and robustness, better aligning with the overall objective of improving performance on noisy and real-world motion data.

3.2.2 Results and Evaluation

Classification Performance Overview

Table 3.1: Classification Performance for **AGE** (Best Accuracy)

Class	**Precision**	**Recall**	**F1-score**	**# instances**
High	0.67	0.67	0.67	18,150
Low	0.75	0.75	0.75	26,900
Medium	**0.81**	**0.81**	**0.81**	52,300
Accuracy		**0.7663**		
Macro Avg	0.74	0.74	0.74	97,350
Weighted Avg	0.77	0.77	0.77	97,350

Table 3.2: Classification Performance for **HEIGHT**

Class	**Precision**	**Recall**	**F1-score**	**# instances**
High	0.73	0.73	0.73	29,100
Low	0.72	0.72	0.72	27,450
Medium	**0.78**	**0.78**	**0.78**	40,800
Accuracy		**0.7464**		
Macro Avg	0.74	0.74	0.74	97,350
Weighted Avg	0.75	0.75	0.75	97,350

Table 3.3: Classification Performance for **WEIGHT**

Class	**Precision**	**Recall**	**F1-score**	**# instances**
High	0.69	0.71	0.70	29,250
Low	0.72	0.72	0.72	31,500
Medium	**0.73**	0.72	**0.73**	36,600
Accuracy		**0.7150**		
Macro Avg	0.71	0.71	0.71	97,350
Weighted Avg	0.72	0.71	0.72	97,350

Among the three target variables 3.1,3.2 and 3.3, the **AGE** classification task achieved the highest mean accuracy of **0.7663**, outperforming HEIGHT (0.7464) and WEIGHT (0.7150). The result suggests that age is most predictable from the features provided. Also, this is similar to the result achived from K-NN model training. Importantly, it is very surprised that the average for all metrices (precision, recall, and F1-score for all 3 targets don't have any significant difference and only range from 0.7 to 0.74, this point out that decision tree itself have a relatively good and consistent performance on TTSWING dataset.

Although there are slight difference, these results may imply that age-related patterns are more separable by the decision tree, likely due to clearer splits or correlations in the input features, while weight classification may require more complex or nonlinear models to improve performance. Moreover, this indicate that previous assumption is correct (there are high amount of noise within the data that make the cross validation results for K-NN greatly reduced).

Best Splitting Criterion

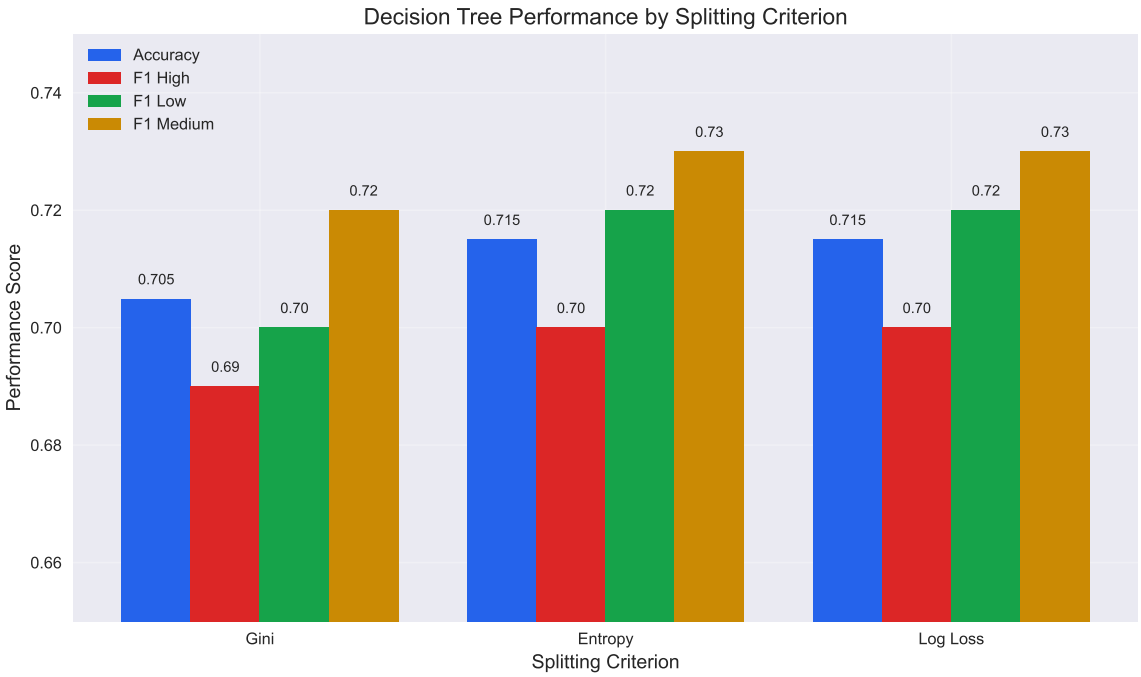

Figure 3.11: Accuracy and F1-score comparison for WEIGHT classification across different criteria

Among all evaluated splitting criteria shown in 3.11, both **'entropy'** and **'log_loss'** achieved the highest overall classification accuracy of **71.5%** and average F1-score of **0.72**, outperforming `gini` which achieved a slightly lower accuracy of **70.5%** and F1-score of **0.70**. More importantly, 'entropy' consistently outperformed 'gini' across all individual class F1-scores, particularly achieving **F1=0.73** for the medium class, which is the most populated and central to balanced classification. This suggests that maximising information gain via 'entropy' leads to better feature splits and overall generalisation for this dataset.

Best Maximum Tree Depth

Figure 3.12: Accuracy and F1-score comparison for WEIGHT classification with varying max depth using entropy criterion

The best models were obtained with `max_depth=None`, achieving both the highest average F1-score (**0.72**) and accuracy (**71.5%**). As shown in Figure 3.12, increasing the tree depth gradually improved performance, with diminishing returns after depth 15. This indicates that deeper trees were beneficial for capturing complex patterns in the motion data. However, the slight plateau in performance suggests caution against uncontrolled depth expansion, as it may risk overfitting, especially with unbalanced classes or noisy features.

Best Minimum Samples for Splitting and Leaf Nodes

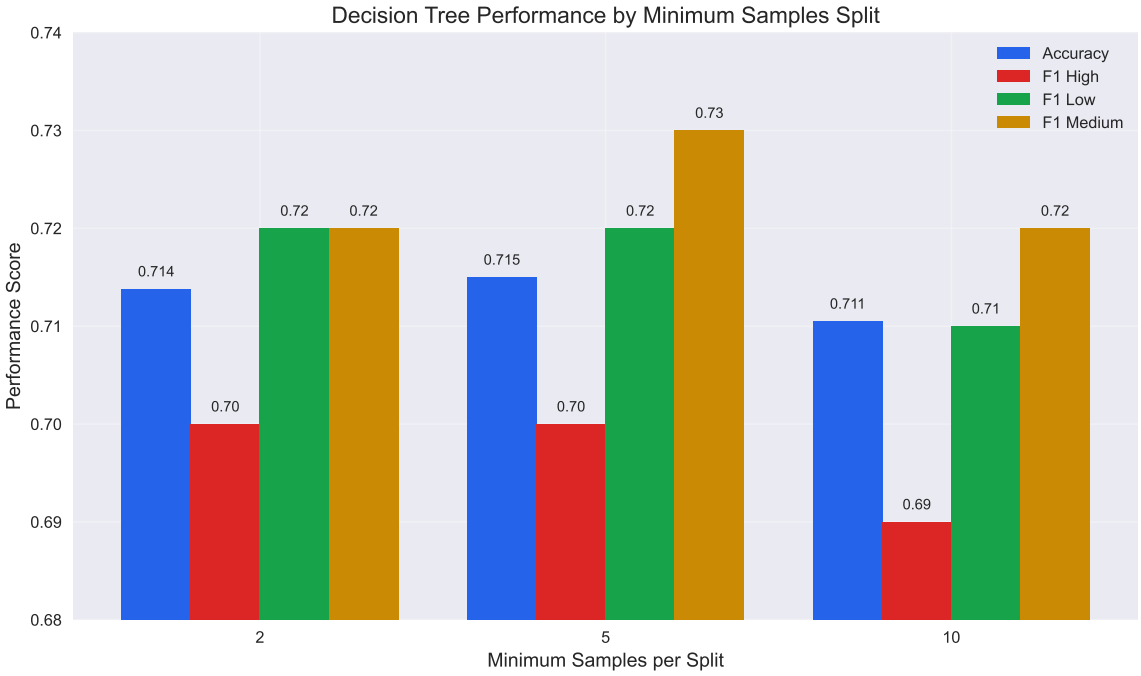

Figure 3.13: Accuracy and F1-score comparison for WEIGHT classification across different min_samples_split values using entropy criterion

The optimal minimum sample split size differed across target labels. For WEIGHT classification, `min_samples_split=5` yielded the best performance, with the highest accuracy (**71.5%**) and average F1-score (**0.72**) (Figure 3.13). Compared to smaller splits (e.g., 2), this setting slightly improved generalisation by reducing overfitting on small sample partitions. It also produced the best F1-score for the medium class (**0.73**), indicating better handling of the dominant class. For other labels such as AGE and HEIGHT, `min_samples_split=2` remained preferable, reflecting label-specific model behaviour. Overall, a moderate value like 5 offered the best trade-off between model complexity and robustness for the WEIGHT prediction task.

Feature ranking

Beyond standard performance metrics, the Decision Tree model was evaluated for its interpretability through domain expert review. Specifically, the hierarchical decision structure provided clear insights into feature importance, with early branching nodes highlighting the most discriminative swing characteristics. This interpretability is illustrated in Figures 3.14, 3.15, and 3.16, which visualise the pruned decision trees trained on Fold 1 of cross-validation. It is important to note that these trees are depth-limited for clarity and serve purely as representative visualisations; they do not capture the full generalisation behaviour of the model across all folds.

Across all three classification tasks (age, weight, and height), several features consistently appear as important, notably **ay_mean**, **az_mean**, and **ax_mean**. These features, representing mean acceleration values along different axes, likely capture essential motion or posture patterns relevant to physical attributes. Among them, **ay_mean** is the most dominant, ranking highest in all three tasks, especially for height classification (importance = 0.0910), indicating a strong correlation between lateral movement and physical characteristics.

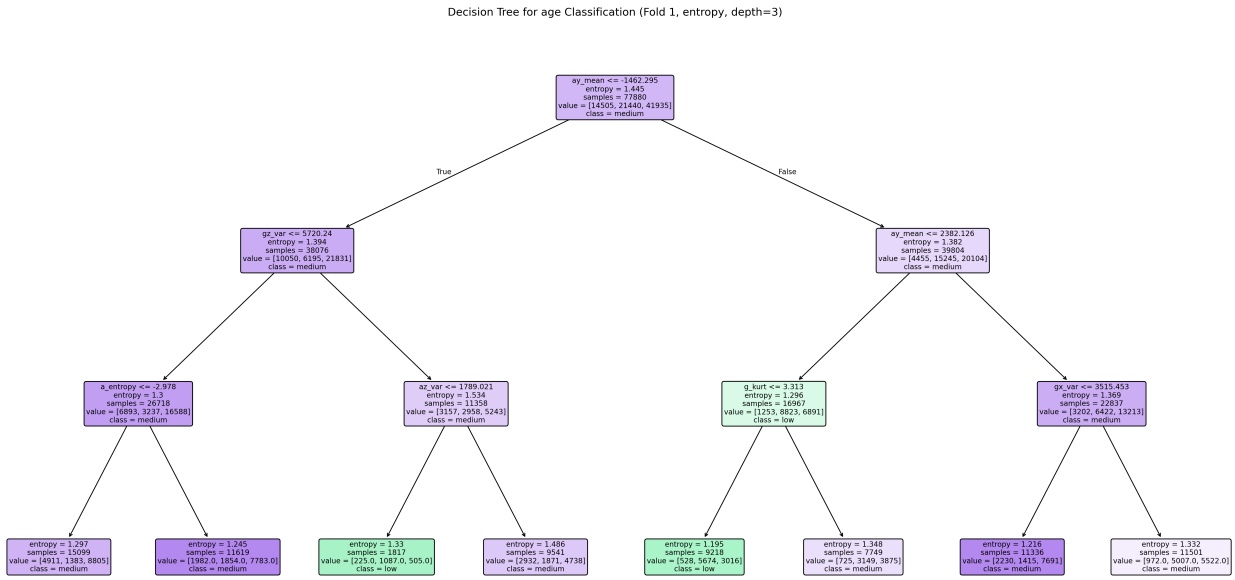

Figure 3.14: Pruned Decision Tree Visualisations for Age Classification Across 5 Cross-Validation Folds (fold 1)

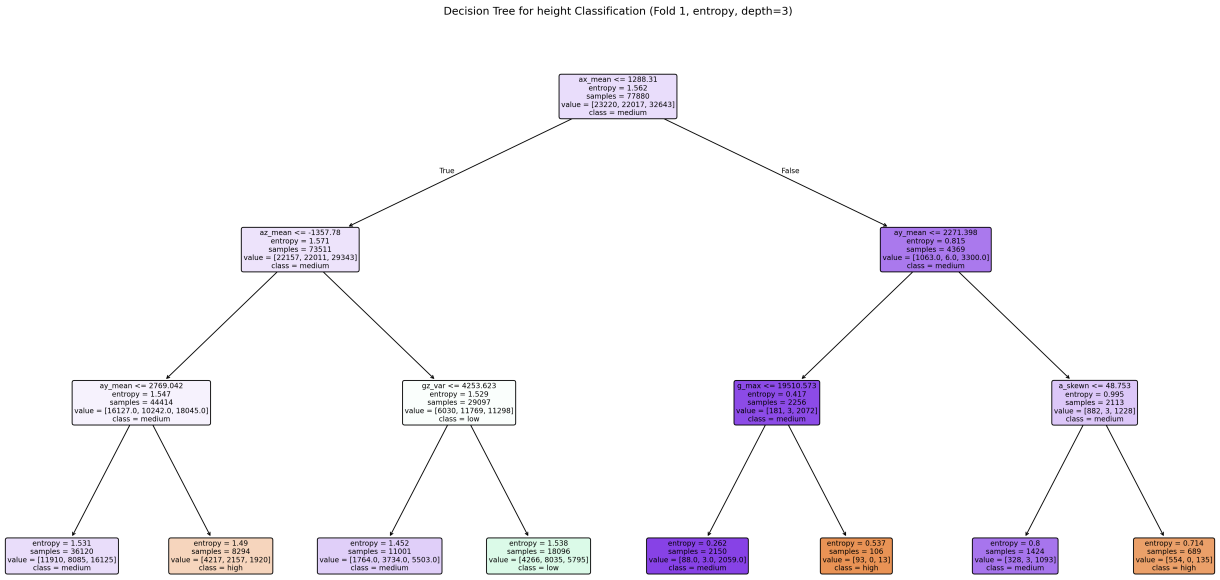

Figure 3.15: Pruned Decision Tree Visualisations for Height Classification Across 5 Cross-Validation Folds (fold 1)

In addition to these shared features, I observe variations in other key features across the tasks. For instance, **az_rms** and **gx_mean** are more prominent for age and weight classification, suggesting that vertical acceleration variability and rotation around the x-axis may provide age- or weight-specific motion cues. Conversely, **gy_var** and **ay_var** emerge as more influential in height prediction, possibly reflecting height-related differences in dynamic stability or gait rhythm.

These differences may be attributed to how each physical attribute influences or constrains movement. For example, height may affect stride length and posture, while weight may be more related to force and stability, thus emphasising different sensor signals. The unique patterns in the gyroscope and

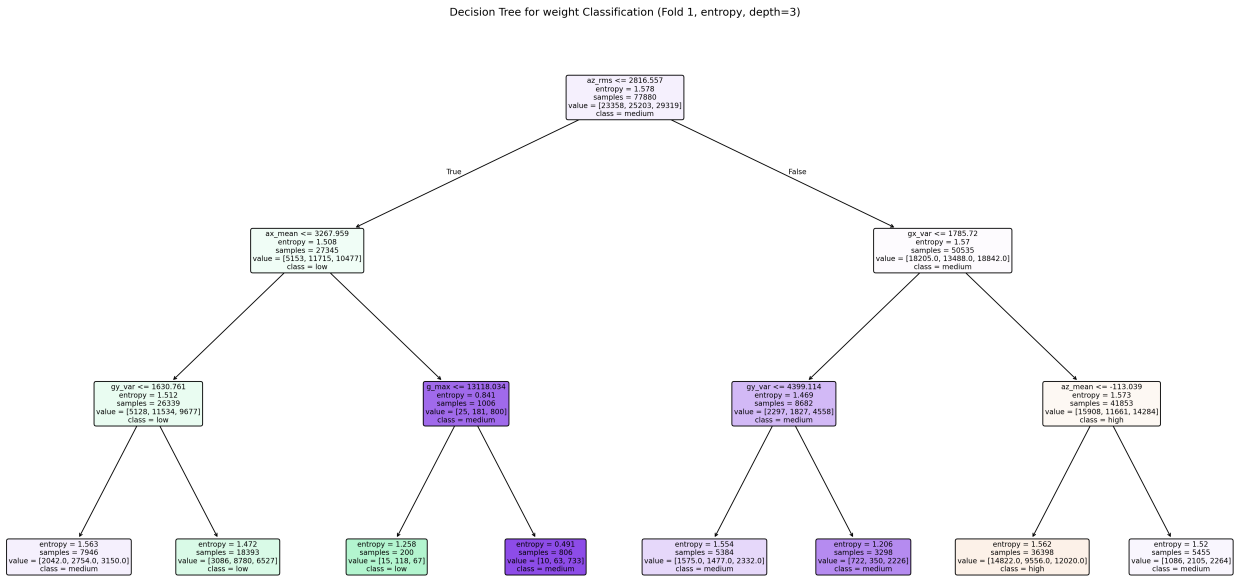

Figure 3.16: Pruned Decision Tree Visualisations for Weight Classification Across 5 Cross-Validation Folds (fold 1)

acceleration features across classification tasks highlight the complexity and multidimensionality of human movement data.

Despite these interpretability benefits, individual Decision Trees exhibited high variance, where small changes in the training data often led to substantially different tree structures. To address this instability and improve generalisation, an ensemble approach—Random Forest—was employed. By aggregating predictions from multiple diverse trees trained on different data subsets, Random Forest (my next model) effectively reduces variance while retaining the strengths of decision tree-based modelling.

3.3 Random Forest

Random Forests are widely recognised as effective baseline models due to their interpretability and capacity to model non-linear feature interactions. While individual Decision Trees offer simplicity in both implementation and explanation, they are prone to overfitting when no Max-depth is specificed. Random Forests, on the other hand, aggregate the predictions of multiple decision trees trained on bootstrapped subsets of the data, yielding improved generalisation performance, especially in high-dimensional feature spaces. Their robustness to noise and ability to capture complex relationships make them suitable for initial experimentation with swing data classification.

3.3.1 Training

The Random Forest classifier [8] was trained as an ensemble of 100 decision trees, leveraging the collective wisdom of multiple weak learners to overcome the limitations observed in both K-NN and single Decision Tree models. While K-NN suffered from sensitivity to noise and poor generalisation, and Decision Trees were prone to overfitting despite hyperparameter tuning, Random Forest addresses these challenges by reducing variance through (1) bootstrap aggregating (bagging) and (2) randomly selecting features from feature subgroups at each split (section 7.2 from [2]). Specifically, the aggregation in (1) is implemented by applying a majority vote to predict the outcome, as mentioned in section 1.11.2 in [9].

Building upon the insights from the study Decision Tree section, where *entropy* was identified as the optimal splitting criterion through comprehensive grid search evaluation, the Random Forest model adopted **entropy** (information gain) as its splitting criterion. This choice was motivated by entropy's superior performance across all three classification targets (height, weight, and age) in the previous Decision Tree experiments, where it consistently outperformed both Gini impurity and log-loss criteria (except log-loss when weight was the target).

Following the robust evaluation methodology established for Decision Tree classification, the Random Forest model was trained using **5-fold cross-validation** across all three classification targets. This approach ensures reliable performance estimates while reducing variance compared to single train-test splits, addressing the generalisation issues encountered with the initial K-NN implementation. The model configuration employed the following key parameters:

- **Number of Estimators:** 100 trees were used in the ensemble, providing a balance between computational efficiency and model stability. Unlike single Decision Trees that can overfit to training data, the ensemble approach reduces variance through (1) and (2).
- **Splitting Criterion:** *Entropy* was selected based on the superior performance demonstrated in the Decision Tree experiments, where it consistently achieved the highest cross-validation accuracy across all target variables.
- **Tree Depth and Splitting Parameters:** The default scikit-learn parameters were initially employed (`max_depth=None`, `min_samples_split=2`) , with the option for hyperparameter optimisation via grid search when the `optimize_hyperparams` flag is enabled. This approach allows for flexible model complexity while maintaining the robust cross-validation framework.

- **Random Feature Selection:** Athougth I don't specify any maximum depth, at each split, a random subset of features is considered (typically $\sqrt{n_{features}}$ for classification (from 1.11.2.3. Parameters [9])), introducing additional randomness that helps prevent overfitting and improves generalisation—directly addressing the noise sensitivity issues observed in K-NN.

The training process of the Random Forest model is integrated with model evaluation and interpretability analysis. Notably, no hyperparameter tuning is conducted for the Random Forest. This decision is based on two key reasons: 1) I adopt the best-performing hyperparameter values identified during the previous Decision Tree experiments, and 2) I aim to maintain a generalised model to avoid overfitting and ensure broader applicability.

To evaluate model performance and interpretability, I employ the following analyses:

(1) **Confusion Matrix Analysis**: This provides a detailed breakdown of classification results, helping to identify which classes are frequently misclassified. It offers a straightforward way to assess the accuracy and precision of the model on a per-class basis.

(2) **Feature Importance Extraction**: The Random Forest ensemble enables the assessment of the relative contribution of each sensor-derived feature to the model's predictions. This analysis provides insights into which motion characteristics are most influential for each classification target.

(3) **Individual Tree Visualisation**: Selected decision trees from the ensemble are visualised to gain a clearer understanding of the decision-making process within the Random Forest. This helps interpret how specific features influence individual predictions, providing a more granular view of model behaviour.

3.3.2 Results and Evaluation

Model Performance Analysis

The Random Forest models demonstrated strong and consistent performance across all three classification tasks where the trend observed from K-NN and Decision Tree are remain unchanged (Age remains to be the highest accuracy). Table 3.4 presents a comparison of the classification performance for age, height, and weight prediction tasks.

Table 3.4: Random Forest Performance Metrics for Each Attribute

Attribute	Accuracy	Precision	Recall	F1-score	Weighted F1
Age	**0.8700**	0.87	0.87	0.87	0.8714
Height	0.7900	0.79	0.79	0.79	0.7907
Weight	0.8200	0.82	0.82	0.82	0.8209

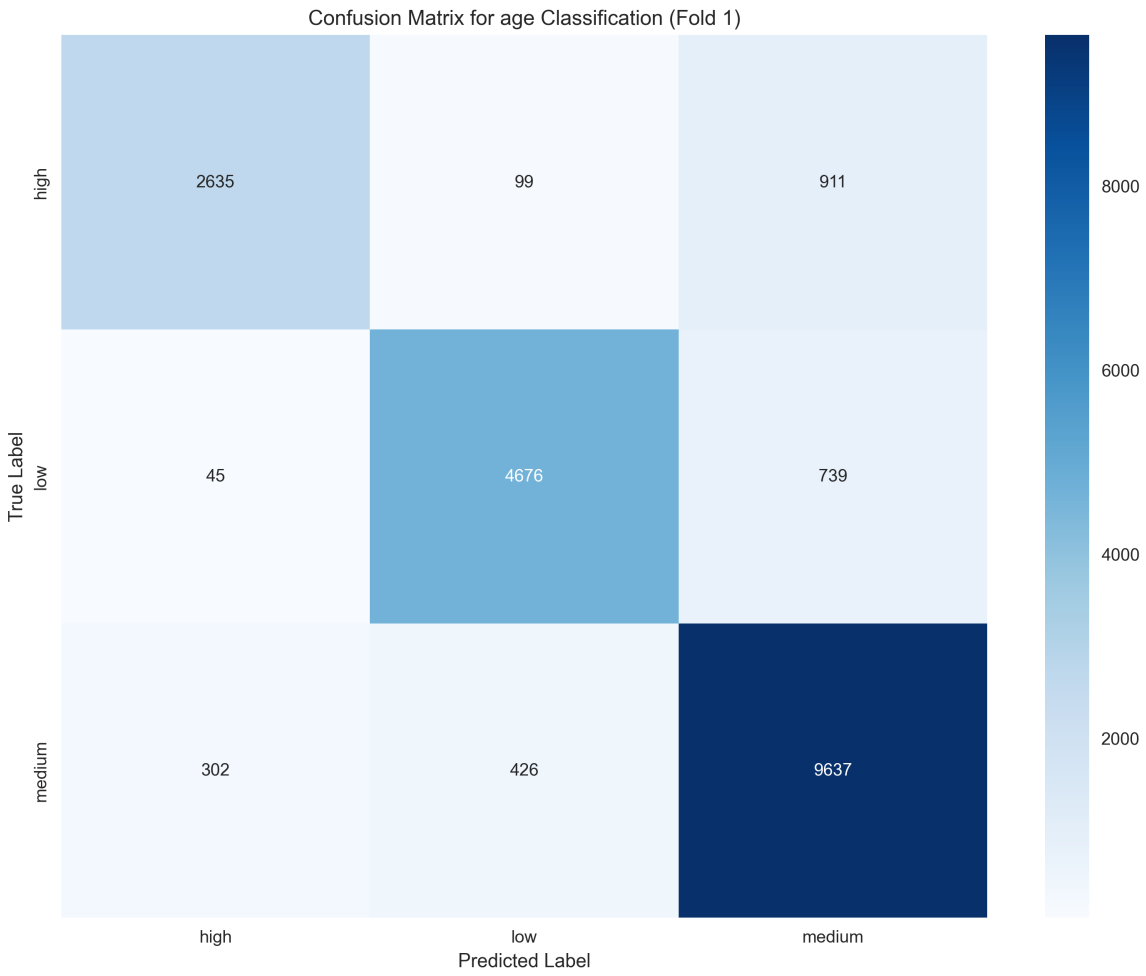

Figure 3.17: Fold 1 Confusion matrix for Age

There are 3 key findings that can be seen from the table 3.4. Firstly, compared to the single Decision Tree results, Random Forest achieved substantial improvements of approximately 10-15 percentage points across all tasks, demonstrating the effectiveness of ensemble methods in reducing variance and improving generalisation, which also confirms that the analyses in the previous section are correct.

Secondly, age classification consistently outperformed height and weight prediction, suggesting that age-related movement patterns are more distinguishable in the sensor data. This aligns with physiological expectations, as age-related changes in motor control and physical capabilities create more pronounced differences in movement signatures.

And thirdly, although age received the highest scores, all models showed relatively balanced performance across different classes (this can be shown in the confusion matrix 3.17) likely due to larger sample sizes and more representative feature distributions.

Feature Importance Analysis

The Random Forest ensemble approach provides valuable insights into feature importance through its inherent ability to measure the contribution of each feature across multiple decision trees. The feature importance analysis reveals consistent patterns across all three classification tasks, while also highlighting task-specific sensitivities that reflect the underlying biomechanical relationships between sensor data and physical attributes.

The most striking finding is the universal dominance of acceleration mean features, particularly **ay_mean** (Y-axis acceleration mean), which consistently ranks as the most important feature across age (about 0.0576), height (about 0.0556), and weight (about 0.0489) classification tasks. This pattern is complemented by **az_mean** (Z-axis acceleration mean) as the second most important feature and **ax_mean** (X-axis acceleration mean) as the third. This hierarchical importance suggests that the mean acceleration values capture fundamental motion signatures that are strongly correlated with all three physical attributes (this is justified by picking one of 100 trees in one of the trained Random Forest models 3.18, mean acceleration is frequently placed in the top features regardless of target and depth).

Respectively, the predominance of Y-axis acceleration (ay_mean) can be attributed to its representation of lateral movement patterns, which are likely influenced by gait characteristics, balance, and postural adjustments that vary systematically with age, height, and weight. Moreover, the Z-axis acceleration (az_mean) captures vertical movement components related to step impact and gravitational effects, while X-axis acceleration (ax_mean) represents forward-backward motion dynamics. And together, these three features form a comprehensive representation of the primary motion axes during human movement.

Beyond these universal features, the Random Forest model reveals task-specific feature preferences that provide insights into how different physical attributes manifest in sensor data. Age classification shows particular emphasis on acceleration RMS values (az_rms: 0.0415) and gyroscope means (gy_mean: 0.0440), suggesting that age-related changes in movement are captured through motion variability and rotational patterns. This aligns with biomechanical literature indicating that aging affects movement stability and coordination.

Height classification demonstrates strong reliance on gyroscope variance features (gy_var: 0.0352), which likely reflects how taller individuals exhibit different rotational dynamics during movement due to altered center of mass and limb proportions. The gyroscope variance captures the variability in rotational movements, which may be more pronounced in taller individuals due to increased leverage effects and different stability requirements.

Weight classification highlights acceleration variance features (az_var: 0.0364) and gyroscope variance (gy_var: 0.0350), indicating that weight-related differences are captured through movement variability rather than mean values. This suggests that heavier individuals may exhibit greater variability in both linear and rotational movements, possibly due to increased inertial effects and altered movement efficiency.

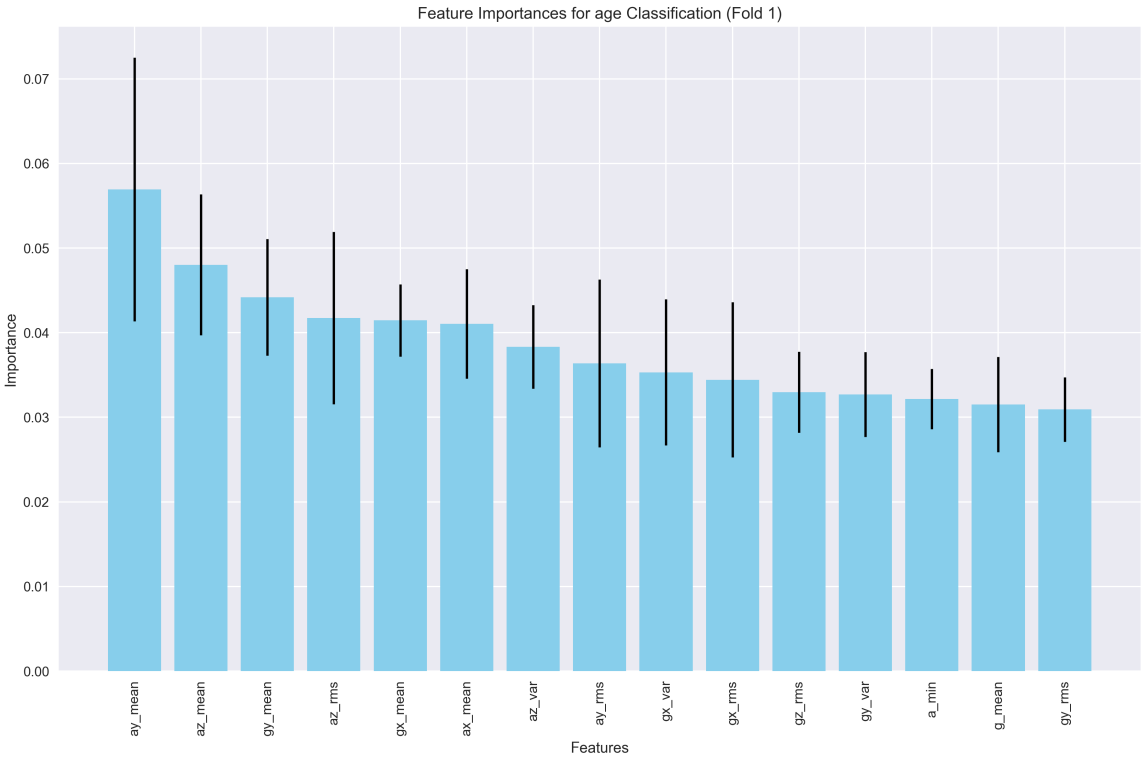

Figure 3.18: Feature importance comparison in Age Random Forest models

The sample decision tree visualisation (Figure 3.19) from the Random Forest ensemble illustrates how these important features are utilised in practice. The tree structure shows the hierarchical decision-making process, with entropy-based splitting criteria that maximise information gain at each node. The root node begins with acceleration entropy (a_entropy), followed by acceleration RMS features (az_rms) and gyroscope variance (gy_var), demonstrating how the ensemble combines multiple discriminative features to achieve robust classification.

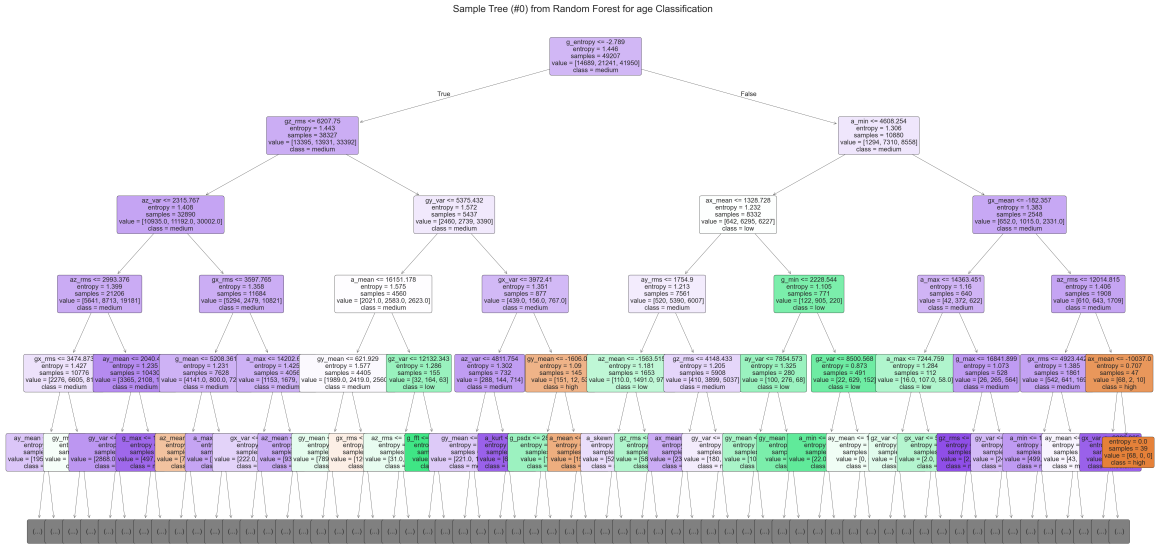

Figure 3.19: Training Progress: Tree 0/100 for Age Prediction (Random Forest)

Chapter 4

Conclusion

This study explored the progression of supervised learning models for predicting physical attributes from sensor data, focusing on K-Nearest Neighbors (KNN), Decision Trees (DT), and Random Forests (RF). Each model was selected to balance trade-offs between simplicity, interpretability, and robustness (or so-called ”generalisation” as mentioned in the previous chapter).

4.1 K-Nearest Neighbors: Foundation and Limitations

The KNN algorithm served as the foundational approach, demonstrating the viability of distance-based classification for sensor data analysis by making no assumptions about the underlying data distribution.

The KNN analysis revealed that individual feature correlations were remarkably weak, yet the algorithm achieved strong ensemble performance (88.05% accuracy for full batch height classification). This counterintuitive finding highlighted a crucial insight: in high-dimensional feature spaces, the collective contribution of weakly correlated features can yield substantial predictive power. Features such as a_entropy, g_entropy, and az_mean emerged as relatively important for height prediction, suggesting that motion variability and vertical acceleration patterns contain meaningful information about physical attributes.

However, KNN’s distance-based approach presented significant interpretability challenges. While the algorithm could identify which combinations of features led to successful classifications, it provided limited insight into why specific features were important or how they related to the underlying biomechanical processes. This black-box nature hindered understanding of the fundamental relationships between sensor measurements and physical attributes, creating a gap between algorithmic performance and domain knowledge integration.

Additionally, KNN’s computational complexity and sensitivity to the curse of dimensionality became apparent when working with high-dimensional sensor data. The algorithm’s reliance on distance calculations across all features meant that irrelevant or noisy dimensions could significantly impact performance, necessitating careful feature selection and preprocessing.

4.2 Decision Trees: Interpretability and Instability

The transition to Decision Trees represented a strategic shift toward interpretability, addressing KNN's primary limitation while maintaining strong predictive performance. The tree visualisations revealed transparent, hierarchical decision-making processes based on sensor features, offering direct insights into strongly discriminated characteristics between different physical attribute classes where domain experts could readily interpret and validate.

Specifically, the Decision Tree analysis consistently identified acceleration mean features (ay_mean, az_mean, ax_mean) as primary discriminators across all three classification tasks. This finding validated the biomechanical intuition that mean acceleration values capture fundamental movement signatures related to gait patterns, postural stability, and locomotor dynamics that vary systematically with age, height, and weight. The hierarchical structure of the trees demonstrated how these features work together to create decision boundaries, with early branching nodes highlighting the most discriminative characteristics.

Beyond the universal importance of acceleration means, Decision Trees revealed task-specific patterns that provided valuable biomechanical insights. For age classification, the trees emphasized features related to movement variability and coordination, reflecting age-related changes in motor control and stability. Height classification showed preference for features capturing stride characteristics and postural adjustments, while weight classification focused on force-related measurements and movement efficiency indicators.

However, Decision Trees suffered from a critical limitation: high variance and instability. Small changes in training data frequently led to substantially different tree structures, making the models unreliable for consistent feature importance assessment. This instability stemmed from the greedy, top-down splitting approach that could amplify minor data variations into major structural differences. The resulting trees, while individually interpretable, lacked the robustness needed for reliable deployment in real-world applications.

4.3 Random Forest: Stability and Comprehensive Understanding

The evolution to Random Forest represented the synthesis of the previous approaches' strengths while addressing their fundamental limitations. By combining multiple Decision Trees trained on different data subsets with randomised feature selection, Random Forest achieved the stability that individual trees lacked while preserving the interpretability advantages over KNN.

The Random Forest feature importance analysis validated the consistency of acceleration mean features (ay_mean, az_mean, ax_mean) as top predictors across all three algorithms, confirming their fundamental importance in characterising physical attributes from sensor data. However, Random Forest provided more nuanced and reliable importance rankings due to its ensemble averaging approach, which reduced the impact of individual tree variance and data-specific anomalies.

The hierarchical importance pattern—with ay_mean (Y-axis acceleration) consistently ranking highest, followed by az_mean (Z-axis) and ax_mean (X-axis)—revealed biomechanical insights that were obscured in the previous approaches. The Y-axis dominance suggests that lateral movement

patterns, influenced by gait characteristics, balance adjustments, and postural control, carry the most discriminative information about physical attributes. This finding aligns with biomechanical literature emphasising the importance of mediolateral stability in human locomotion.

Random Forest also revealed task-specific feature preferences with greater reliability than individual Decision Trees. Age classification's emphasis on acceleration RMS values and gyroscope means reflected age-related changes in movement variability and rotational dynamics. Height classification's reliance on gyroscope variance features captured the altered rotational patterns associated with different limb proportions and center of mass positions. Weight classification's focus on acceleration and gyroscope variance features indicated how mass differences manifest through movement variability and efficiency patterns.

The ensemble nature of Random Forest effectively addressed the overfitting tendencies observed in individual Decision Trees while maintaining intuitive feature importance interpretation that was lacking in KNN. This combination of stability, interpretability, and performance made Random Forest particularly valuable for understanding the complex biomechanical relationships underlying sensor-based physical attribute classification.

Finally, this report is not exhaustive. Other types of models, such as boosting ensembles (e.g., AdaBoost), Gaussian Processes, Convolutional Neural Networks (CNN), or simple feedforward networks could also be applied and may potentially outperform the three models evaluated in this study.

Bibliography

[1] C.-Y. Chou, Z.-H. Chen, Y.-H. Sheu, H.-H. Chen, and S. K. Wu, "Ttswing: a dataset for table tennis swing analysis," *arXiv preprint arXiv:2306.17550*, 2023.

[2] A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön, *Machine Learning: A First Course for Engineers and Scientists*. Cambridge University Press, 2022.

[3] D. Cournapeau, "Standardscaler." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[4] ——, "Kneighborsclassifier." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[5] ——, "Distancemetric," accessed: 2025-05-24. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.DistanceMetric.html

[6] ——, "Decisiontreeclassifier." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[7] ——, "log_loss," accessed: 2025-05-24. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html

[8] ——, "Randomforestclassifier." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[9] ——, "Ensembles," accessed: 2025-05-25. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.htm

[10] OpenAI, *GPT-4o Model Documentation*, OpenAI, 2025, cited for information on GPT-4o's capabilities in brainstorming and grammar correction. [Online]. Available: https://platform.openai.com/docs/models/gpt-4o

[11] ——. (2025) Gpt-4o mini model documentation. OpenAI. Cited for information on GPT-4o Mini's capabilities in brainstorming and grammar correction. [Online]. Available: https://platform.openai.com/docs/models/gpt-4o-mini

[12] Anthropic, *Claude 3.7 Sonnet Model Documentation*, Anthropic, 2025, cited for information on Claude 3.7 Sonnet's capabilities in code generation. [Online]. Available: https://www.anthropic.com/claude/sonnet

Appendix

GitHub Repository

The complete source code, data processing scripts, and experimental configurations used in this study are publicly available in the GitHub repository below:

`https://github.com/thanghoang7020202/TTMotionHealthAnalytics`

Lecturers and tutors from COMP4702 as well as other researchers and practitioners are encouraged to explore the repository for replication, further analysis, or extension of this work.