# CSE343 Machine Learning, Monsoon 2024

## Machine Learning for Strength Exercise Classification and Repetition Counting

Grishma Bellani (2022189) Riya Gupta (2022410) Shreyansh Srivastav (2022485) Shrutya Chawla (2022487) Vimansh Mahajan (2022572)

## Abstract

*Strength training, in addition to aerobic exercises, is an important component of a balanced exercise program. However, mechanisms for automatically tracking free weight exercises have not yet been fully studied. The aim of this project is to further explore the possibilities of context-aware applications within the strength training domain. It analyzes wristband accelerometer and gyroscope data obtained during barbell exercises. The goal is to explore, build, and evaluate models that can, just like human personal trainers, track exercises and count repetitions. The methods evaluated in this paper use a supervised learning approach for classification. LINK*

## 1. Introduction

Wearable sensor technologies, including accelerometers and gyroscopes, have significantly advanced the capabilities of activity monitoring, particularly for aerobic exercises like running and cycling. However, the application of these technologies to the automated tracking of strength training exercises remains limited. Strength training, which is vital for muscle development and overall fitness, has not received the same level of support from wearable devices, leading to a gap in accurate tracking and real-time feedback for exercises involving free weights.

To address this, the current project utilizes MetaMotion sensors to collect motion data during barbell exercises. By analyzing the accelerometer and gyroscopic data from these sensors, the study seeks to develop machine learning models that can classify different exercise movements and count repetitions accurately. This approach aims to enhance the role of wearable devices in strength training by providing comprehensive exercise monitoring, thus bridging the gap between current fitness technology and the needs of strength training enthusiasts.

## 2. Literature Survey

- https://pdf.sciencedirectassets.com/278653 This paper develops models to evaluate exercise quality based on force, displacement, velocity, and repetition time, enabling automatic feedback for athletes. Data was collected from 15 athletes using a leg press machine with a load cell and rotary encoder to capture force and displacement. The study employed Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), achieving 85% classification accuracy on the validation set after 81 epochs
- https://ieeexplore.ieee.org/document/8526202 The system utilizes a wrist-worn MEMS accelerometer (MPU6050) to track movement by measuring acceleration along three axes (X, Y, Z), enabling step counting and calorie expenditure calculations over 30 days. Users manually input their daily calorie intake, which, along with the calories burned, helps assess their health status (fit/unfit). Logistic Regression is the primary machine learning model used to predict fitness status, achieving an accuracy of 85.42%.

## 3. Dataset

### 3.1. Dataset Description and Extraction

The dataset utilized in this study was collected using the MetaMotion sensor watch, capturing both accelerometer and gyroscope data over a 20-day period with five participants. This dataset encompasses a variety of strength exercises and corresponding rest periods, designed to replicate realistic strength training scenarios based on the Starting Strength program. Data collection included two repetition schemes (5 and 10 repetitions) and resting data to analyze state transitions between rest and activity.

Participants (N=5)

| Participant | Gender | Age | Weight (Kg) | Height (cm) | Experience (years) |
|---|---|---|---|---|---|
| A | Male | 23 | 95 | 194 | 5+ |
| B | Male | 24 | 76 | 183 | 5+ |
| C | Male | 16 | 65 | 181 | <1 |
| D | Male | 21 | 85 | 197 | 3 |
| E | Female | 20 | 58 | 165 | 1 |

Figure 1. Participants

| Feature | Description |
|---|---|
| **epoch** | Timestamp representing the moment the data was recorded. |
| **acc_x, acc_y, acc_z** | Acceleration values along the x, y, and z axes, respectively. |
| **gyr_x, gyr_y, gyr_z** | Gyroscope data along the x, y, and z axes, capturing rotational movements. |
| **participant** | Identifier for the individual performing the exercises. |
| **category** | Indicates the weight category, either heavy or medium. |
| **label** | The specific type of exercise performed (e.g., squat, bench press, deadlift). |
| **set** | Identifies the set number within the exercise routine. |

Table 1. Dataset Features

Data extraction involved continuous logging of sensor readings during both exercise and rest phases, with each log timestamped using the epoch column for chronological alignment. The raw data was then segmented by exercise type.

### 3.2. Dataset Preprocessing

The raw dataset contained a total of 69,677 entries, each consisting of an epoch timestamp and corresponding x, y, and z-values for both the accelerometer and gyroscope. The following preprocessing steps were performed:

- **Data Aggregation**: Since the sensor data was logged at high frequency, it was aggregated into 0.2-second intervals to minimize information loss. Numerical values (e.g., acceleration and gyroscope data) were aggregated using the mean, while categorical values (e.g., exercise labels) were aggregated using the mode.

- **Outlier Detection**: Several outlier detection techniques were evaluated, including Local Outlier Factor (LOF), Interquartile Range (IQR) Method, and Chauvenet's Criterion. Chauvenet's Criterion was selected as it effectively detected a manageable number of outliers without excessively removing significant data points.

- **Handling Missing Values**: Missing values were addressed using data interpolation.

- **Low-pass Filtering**: To reduce noise in the sensor data, a Butterworth low-pass filter was applied to the accelerometer and gyroscope features. This filtering removed high-frequency noise, focusing on the more prominent movements related to exercise repetitions. A cutoff frequency of 1.3 Hz was selected after analyzing the data, which corresponds to the average frequency of the exercise movements.

- **Principal Component Analysis (PCA)**: After evaluating the explained variance, it was found that the first 3 components captured the majority of variance in the data, and these components were included in the final dataset for model training.

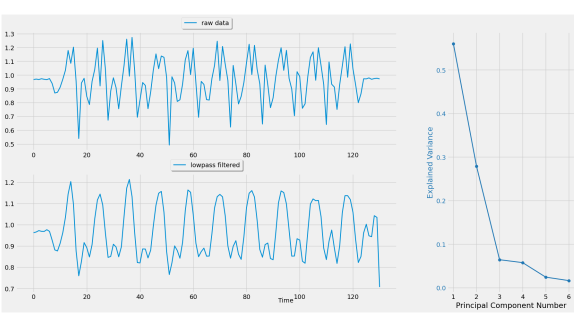

Figure 2. Low-pass filter (left) and principal components (right).

### 3.3. Exploratory Data Analysis (EDA)

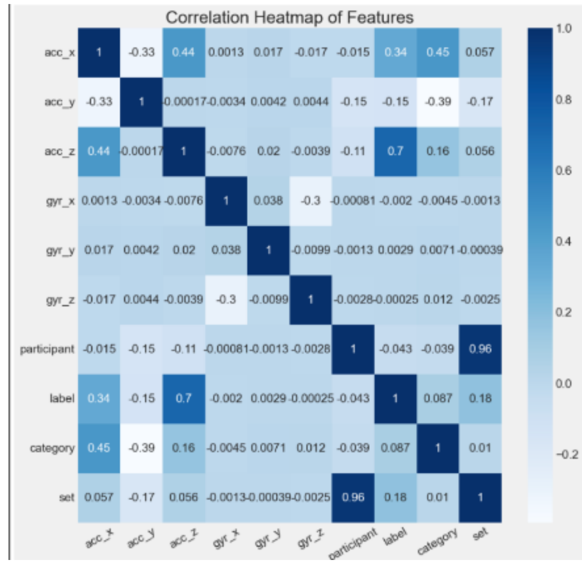

Figure 3. Correlation Matrix.

From the Heatmap `acc_x` and `acc_z`, are crucial predictors for activity classification, with `acc_z` showing a strong correlation of 0.70 with the label. Meanwhile, gyroscope data displays weak correlations, indicating a limited influence on classification outcomes. A significant portion of the data comes from exercises like bench press, overhead press, and squats, enhancing the model's ability to predict these movements. Most exercises are performed with medium weights, offering consistent data for reliable model training.

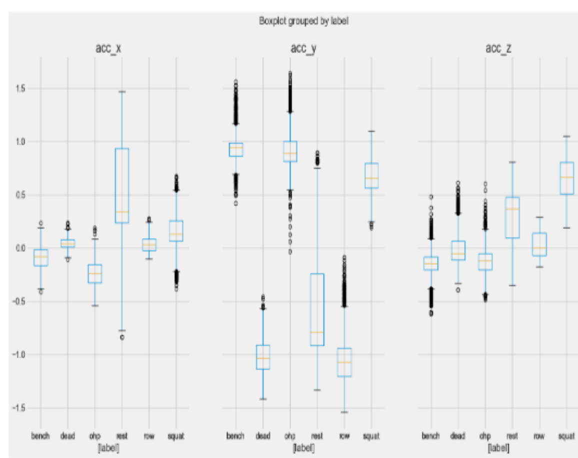

Figure 4. Box Plot.

`acc_x`: Bench shows a tight distribution near 0, while rest has a broader spread, indicating variability during rest.

`acc_y`: Rest clusters around negative values (minimal movement), while bench, dead, and row have similar medians but wider variability. OHP shows more positive values.

`acc_z`: Bench has a narrow range with outliers, rest clusters near 0 (minimal movement), and squat shows a slightly wider spread t

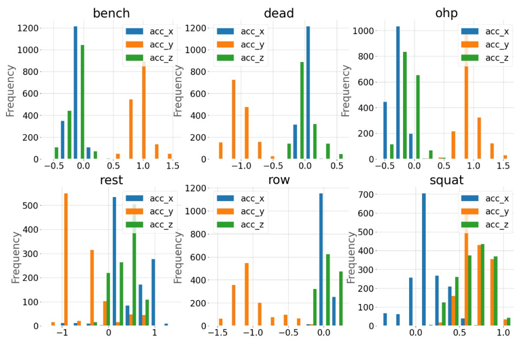

Figure 5. Normal Distribution for Accelerometer Data

Since the data is mostly normally distributed so we can use Chauvenet's Criterion for Outlier Detections.

## 4. Methodology

- **Data Collection**: The primary dataset was obtained from MetaMotion sensor watches, containing acceleration (`acc_x`, `acc_y`, `acc_z`) and gyroscope (`gyr_x`, `gyr_y`, `gyr_z`) readings.
- **Data Aggregation**: The raw sensor data was aggregated into 0.2-second intervals to facilitate analysis.
- **Exploratory Data Analysis (EDA)**: Visualization techniques were applied to the aggregated data, revealing distinct patterns.
- **Outlier Detection and Noise reduction**: Outlier detection was performed using Chauvenet's Criterion while missing values were managed through interpolation. The duration of exercise sets was calculated based on sampling frequency. A Butterworth low-pass filter was applied to reduce noise.
- **Principal Component Analysis (PCA)**: PCA was employed to reduce dimensionality, retaining the 3 principal components that explained the majority of variance in the dataset.
- **Feature Engineering**: Aggregated features, Fourier transformation, clustering, and temporal aggregation were done to find potential features which discover deeper patterns in the data.
- **Overlapping Windows:** To mitigate overfitting and eliminate overlapping windows, we selected alternate rows from the dataset, reducing its size by half.
- **Modeling**: Neural Network, Random Forest, KNN, Decision Tree, and Naive Bayes were trained and their performance was noted, followed by hyperparameter tuning to find the best accuracy possible.
- **Repetition Counting**: A custom algorithm was developed to count exercise repetitions using accelerometer and gyroscope data. The signal was preprocessed with a Butterworth low-pass filter, and repetitions were identified by detecting peaks in the filtered data. This approach was validated by comparing predicted repetitions against labeled ground truth, with performance measured using mean absolute error.

## 5. Feature Engineering

Here we discuss how additional features were derived from the data.

- **Aggregated Features**: The scalar magnitudes r of the accelerometer and gyroscope were computed in order to further utilize the data. The benefit of employing r as opposed to any specific data direction is that it can manage dynamic reorientations and is indifferent to device orientation [29]. The calculation of r is:

$$r_{\text{magnitude}} = \sqrt{x^2 + y^2 + z^2}$$

- **Temporal Aggregation**: Temporal aggregation involves summarizing data over a fixed period to capture patterns and reduce noise. In this case, features like mean and standard deviation were computed over 4-second windows.
- **Frequency Domain**: The analysis of frequency data using Fourier transformations helps identify patterns in movement. The data is decomposed into frequency components, including **maximum frequency, signal-weighted average, and spectral entropy.** These features are useful for detecting repetitive movements in strength training and distinguishing between smooth and erratic actions.
- **Clustering:** Similar data points were grouped together with k = 5, K-mean clustering were applied and it was found that cluster 1 captured the bench press and overhead press data, Cluster 2 captured the deadlift and Cluster 3 captured the row. Cluster 5 captured squats. Cluster 4 contains rest data but fails to capture this accurately.

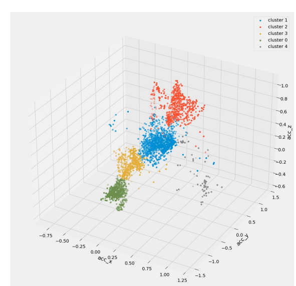
Figure 6. Clustering

## 6. Modeling

- **Dataset splitting**: The dataset was split into training and testing data with a 75:25 ratio.
- **Feature selection**: Five sets of feature lists were created to test the performance of models on varied datasets.

  Note: Selected Features have been identified through forward feature selection using a simple decision tree. Results showed that after 10 features, performance plateaued. The 5 features with the most predictive power are listed in the table.

| Feature Set | List of Features |
|---|---|
| Feature Set 1 | Basic Features |
| Feature Set 2 | Basic Features + Aggregated Features + PCA Features |
| Feature Set 3 | Feature Set 2 + Temporal aggregation Features |
| Feature Set 4 | Feature Set 3 + Frequency Features + Cluster Features |
| Selected Features | pca 1, acc y, pca 3, gyr x temp std ws 4, acc r pse |

Table 2. Feature Sets and Selected Features

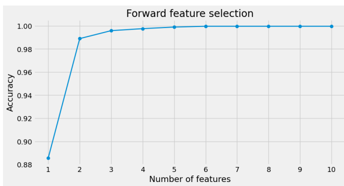


Figure 7. Forward Feature Selection

- **Initial model evaluation**: The LazyPredict library was employed to benchmark various models on all datasets quickly. LazyPredict provides an automated evaluation of multiple models with minimal setup. It was observed that Random Forest consistently outperformed other models during this initial phase.

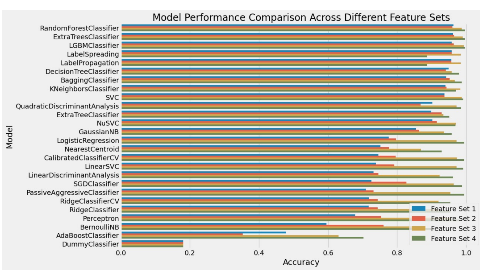


Figure 8. Model Performance Comparison

- The various models tested included Neural Network, Random Forest, KNN, Decision Tree, and Naive Bayes. Grid search, along with regularization, was performed on all models to optimize performance across different feature sets.

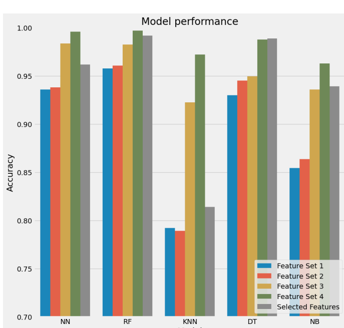


Figure 9. Model Performance

- **Best Model**: Random Forest, followed closely by Neural Networks, demonstrated the highest performance across different feature sets. Grid search with 5-fold cross-validation was conducted to find the optimal hyperparameters, resulting in the following configuration:
  - **Minimum samples per leaf**: 2
  - **Number of estimators**: 100
  - **Criterion**: Gini

As a result, an overall accuracy of 98.51% was observed.

| Models | Accuracy |
|---|---|
| Decision Tree | 96.87% |
| KNN | 85.79% |
| Naive Bayes | 91.11% |
| Neural Networks | 96.29% |
| Random Forests | 97.77% |

Table 3. Average Accuracy Across Different Feature Sets

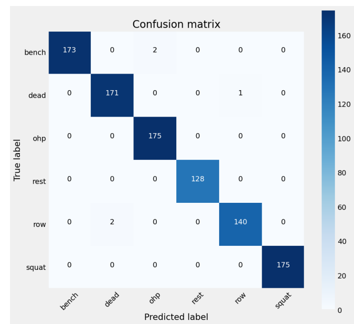


Figure 10. Confusion Matrix

## 7. Results and Analysis

- **Data Size Reduction**: The dataset was successfully reduced from 69,677 entries to 9,009 through aggregation, effectively retaining essential movement patterns while managing large data volumes.

- **Sensor Data Comparison**: Analysis indicated that accelerometer data exhibited fewer outliers compared to gyroscopic data.

- **PCA for Dimensionality Reduction**: Using three principal components from PCA captured the majority of variance, simplifying the analysis while preserving critical information for modeling.

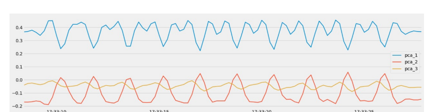

Figure 11. pca components

- **Outlier Detection** Detected ad removed over 185 outliers using Chauvenets criterion.

- **Feature Engineering**: Found **10 features** that had the most predictive power using **Forward Feature Selection**.

- **Repetition Counting**: A repetition counting algorithm successfully identified exercise repetitions using accelerometer and gyroscope data. The method achieved a mean absolute error of 0.88 repetitions per set when compared with the labeled ground truth, demonstrating its reliability and accuracy.

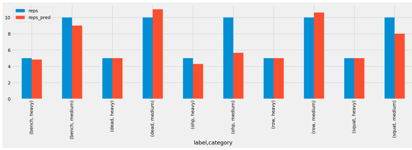

Figure 12. Counting Repetition

- **Models** Found Random Forest to be the highest performing model with an accuracy of **97.77%** on the test set and after hyperparameter tuning increased it to **98.51%.**
- **Forward Feature Selection** identified 10 key features that were sufficient to train the model effectively. The accuracy achieved using these selected features was comparable to the accuracy obtained with the full feature set.

## 8. Conclusion

This study successfully demonstrates the potential of wearable sensor technology combined with machine learning to improve the monitoring of strength training exercises. By leveraging MetaMotion sensors, extensive preprocessing, and feature engineering, we developed a Random Forest model achieving 98.51% accuracy in classifying exercises and counting repetitions. These findings highlight the reliability of accelerometer data over gyroscope data for such tasks, and the importance of dimensionality reduction and feature selection in optimizing model performance. Our work bridges the gap between fitness technology and strength training, providing a foundation for more advanced, context-aware fitness applications in the future.

## 9. Contributions

- **Grishma Bellani**: Data Visualization, Feature Engineering, Outlier detection, Low pass filter & PCA, neural network.
- **Riya Gupta**: Data Collection, Random forest, Feature Engineering , Repetiton Counting Algo, Result and Analysis.
- **Shreyansh Srivastav**:Data Pre-Processing, EDA, Outlier detection, Predicitve Modeling, Low-pass Filter & PCA.
- **Shrutya Chawla**:Predicitve Modeling, Result and Analysis, Fourier Transformation  Clustering,Data Pre-Processing.
- **Vimansh Mahajan**: Data Collection, Fourier Transformation  Clustering, Repetiton Counting Algo.

## References

[1] `https://link.springer.com/book/10.1007/978-3-319-66308-1`.

[2] `https://mbientlab.com/metamotions/`.

[3] `https://github.com/mhoogen/ML4QS/`.

[4] `https://ieeexplore.ieee.org/document/8526202`.

[5] `https://www.sciencedirect.com/science/article/pii/S1877705812017092?fr=RR-2&ref=pdf_download&rr=8d6aeb0a482fa833`.