# Physical Activity Prediction Report

**Group 17** - **Team Members** : Krishen Jagani, Waasi A Jagirdar, Jason Misquitta, Gautam Vanishree Raghu, Yile Xu

## Abstract:

Wearable devices like smartwatches and fitness trackers generate vast amounts of physiological and motion-related data through sensors such as heart rate monitors and gyroscopes. The ability to gather useful high-level information like the physical activity the user is currently performing (e.g., walking, running, jogging, exercising), will open up to various real-world applications. This data can be used by healthcare professionals to monitor the patient's physical activities, providing insights to daily lifestyle activities (recommending standing breaks, encouraging exercising, etc) or enabling app developers to create innovative applications and systems to understand and cater to individual user needs based on their current physical activities.

This project aims to develop and evaluate machine learning models to classify activities. Various models including Logistic Regression, Decision Trees Random Forest, Support Vector Machines and Neural Networks were trained to classify activities. Moreover, hyperparameter tuning is performed to obtain the optimal model for each of the techniques mentioned.

## Dataset and Preprocessing:

The PAMAP2 Physical Activity Monitoring dataset (https://archive.ics.uci.edu/dataset/231/pamap2+physical+activity+monitoring) consists of over 2 million rows of observations collected from 9 different subjects. There are 32 physical activity monitoring metrics as features and 12 different physical activities (walking, cycling, playing soccer, etc.) as the target.

The preprocessing techniques involved removal of missing values, dropping the column consisting the subject IDs and excluding all the rows that were classified as transient activities. These rows were removed as it was noise data and was not considered for further analysis (as suggested by the data source). Further, feature scaling was performed on all the columns to bring the data values on the same scale.

## Models and Methodology:

### 1. Logistic Regression:

Logistic regression is a linear classification algorithm that models the probability of a target class using a logistic (sigmoid) function. While it is traditionally designed for binary classification, it can be optimized and applied to this multi-class classification problem with 12 classes. After training and testing logistic regression with default hyperparameters, grid search with 5-fold cross-validation was employed to find the best set of hyperparameters for the model. The grid search explored the following hyperparameters including regularization strength C, type of penalty, and class_weight.

**2. Decision Trees:**

Decision trees are efficient in dealing with non linear decisions and are highly interpretable because of the greedy approach that they use for creating decision boundaries. The training of Decision Trees was tuned using GridSearchCV to optimize the model using the parameter tree depth. The search space that was explored to get the best depth was [1,10]. Moreover, Gini Index is used as the measure of impurity to create the splits at the internal nodes.

**3. Random Forests and XGBoost:**

Random forest is an ensemble method (bagging) that combines predictions from multiple decision trees to improve classification accuracy and reduce the variance in the model. This leads to an enhanced and robust performance of the overall model. The Random Forest model utilizes 100 trees to classify the physical activities. Further, features like heart rate, IMU_chest_temperature, IMU_hand_temperature are the most important features to distinguish between physical activities.

Secondly, we also performed XGBoost on the dataset that applies gradient boosting with 100 estimators to classify the physical activities.

**4. Support Vector Machines:**

Through the SVM , we aim to find the optimal hyperplane that maximizes the margin between different classes in the feature space. Our dataset involves sensor readings like heart rate, gyroscope, and accelerometer values, which are likely to have nonlinear relationships. RBF(Radial Basis Function) can effectively capture such complexities. Training an SVM involves solving a quadratic optimization problem which is of the time complexity O (n squared) so for the initial demo in the Train_funcs file , we have trained the SVM with 20% of the dataset subsampled which gives an accuracy of 98.8%. In the final combined codebase , the entire dataset has been used for training leading to an accuracy of 99.53% This shows how effective and powerful an SVM can be.

**5. Neural Network:**

The neural network consists of three hidden layers. The first and the second layer consists of 32 neurons each and the third layer consists of 64 neurons. The output layer consists of 12 neurons corresponding to the 12 classes we have. ReLu activation is applied to all the hidden layers and softmax activation function is applied to the output layer. A Batch Normalization is applied after every hidden layer.

Hyperparameter tuning is performed to obtain the optimal model, focussing on the batch size and the optimizer. Batches of sizes 32 and 64 were evaluated and both Adam and SGD optimizers were tested. The models were trained for 10 epochs using categorical cross entropy loss function.

## Results:

| Model | Testing Accuracy with Best Hyperparameters |
|---|---|
| Logistic Regression | 82.14 % |
| Decision Trees | 86.35 % |
| Random Forest | 99.986 % |

| | |
|---|---|
| XGBoost | 99.984 % |
| Support Vector Machines | 99.53% |
| Neural Network | 99.31 % |

## Result Discussion:

- **Logistic Regression** results in the lowest testing accuracy of 82.14% among all models, because it is inherently a linear model, making it unable to effectively capture complex and non-linear relationships in the data regardless of hyperparameter tuning. With 12 classes and 33 features, this dataset requires more complicated models to achieve significant performance gains.
- **Decision Trees** achieved a testing accuracy of 86.35%, outperforming logistic regression due to their ability to model non-linear decision boundaries. However, since a single tree is prone to overfitting, it was outperformed by ensemble models.
- **Random Forest** and **XGBoost** achieved near-perfect testing accuracies of 99.986% and 99.984%, respectively. The ensemble approach of Random Forest (bagging) and XGBoost (boosting) increases robustness and reduces variance. Their success can also be attributed to their feature importance rankings, where heart rate, IMU_chest_temperature, and IMU_hand_temperature played crucial roles.
- **Support Vector Machines (SVMs)** achieved an impressive testing accuracy of 99.53%, effectively capturing non-linear relationships through the use of an RBF kernel. However, the model's training time was considerable due to the computational complexity of the quadratic optimization process.
- **Neural Networks** achieved a high testing accuracy of 99.31%, leveraging their ability to model non-linear relationships through three hidden layers with batch normalization, ReLU activations, and softmax outputs. Hyperparameter tuning of batch size and optimizer further improved performance.

## Conclusion

To conclude, we successfully classified the physical activities from the PAMAP2 dataset, which contains 12 activity classes and 33 sensor-derived features. Given the non-linear nature of the data, we found that selecting an appropriate model was critical for achieving high classification accuracy.

Ensemble methods like Random Forest and XGBoost achieved very high accuracies of 99.986% and 99.984%, demonstrating their ability to reduce variance and improve robustness. Neural Networks (99.31%) and SVMs (99.53%) also performed well, leveraging non-linear decision boundaries. Decision Trees (86.35%) outperformed logistic regression (82.14%) but were surpassed by their ensemble counterparts. The limited performance of logistic regression highlights its inability to model non-linear relationships, reinforcing the need for more expressive models.

Future applications in physical activity recognition should prioritize ensemble methods or deep learning models to better capture complex, multi-class interactions and achieve superior classification performance.