Report 2: Activities detection

Federico Villata, s247586,
ICT for Health attended in A.Y. 2022/23

January 17th 2023

1 Introduction

Human activity recognition is the problem of classifying sequences of sensor data recorded by specialized harnesses or smart phones into known well-defined movements. People interested in fitness would like automatic detection of their current activity to more accurately assess the calories burnt.

Clustering algorithm K-Means was used on the public dataset "Daily and Sports Activities Data Set"[1] to recognize activity.

2 Data analysis

The 19 activities available in the "Daily and Sports Activities Data Set" in their abbreviated form are listed in table 1. Data used in the study was collected from 8 subjects, utilizing samples recorded at a rate of 25 Hz from sensors including 3-axis accelerometers, gyroscopes, and magnetometers located on the torso, right arm, left arm, right leg, and left leg. This resulted in a total of 5 sensor positions, with 3 sensors per position, each measuring 3 values on the x-y-z axes, yielding a total of 45 values for each individual sample.

Each subject practiced the activity for 5 minutes and slices of 5 seconds were created, thus totaling 60 slices. The first 30 slices are used to train the model and the remaining 30 are used to test the model's performance.

In the following considerations, only data related to patient number 3 are considered.

1	sitting	2	standing	3	lying.ba	4	lying.ri	5	asc.sta
6	desc.sta	7	stand.elev	8	mov.elev	9	walk.park	10	walk.4.fl
11	walk.4.15	12	run.8	13	exer.step	14	exer.train	15	cycl.hor
16	cycl.ver	17	rowing	18	jumping	19	play.bb		

Table 1: List of activities

3 Feature engineering

The simple statistical features listed in the table 2 were constructed from the raw data to exploit these additional attributes to improve the quality of the results of machine learning processes. The following features were calculated for each signal on each slice: mean, standard deviation, average absolute deviation, minimum value, maximum value, the difference between the maximum and minimum values, median, median absolute deviation, interquartile range, negative values count, positive values count, number of values above mean, number of peaks, skewness, kurtosis, and energy. In particular, the energy of the signal in every axis is computed by taking the mean of the sum of squares of the values in a slice in that particular axis.

1	mean	2	standard deviation	3	average absolute deviation
4	minimum value	5	maximum value	6	difference of max and min values
7	median	8	median absolute deviation	9	interquartile range
10	negative values count	11	positive values count	12	number of values above mean
13	number of peaks	14	skewness	15	kurtosis
16	energy				

Table 2: List of features

4 Feature selection

The feature selection process was carried out in order to minimize the number of sensors used while still achieving good performance. Feature selection is performed before deploying the model on the device which will record the signals so it does not add a computational load on it. Also using fewer features decreases the energy needed to perform the computation.

In order to select only the most relevant features, an ExtraTrees classifier was used, since tree algorithms are one of the most common techniques to perform this task thanks to the fact that they can assign an importance score to each feature.

Similar to random forests, ExtraTrees is an ensemble machine learning approach that trains numerous decision trees and aggregates the results to output a prediction. ExtraTrees uses the entire training dataset to train decision trees.

20 features were used since it was a good balance between the number of sensors used to record the raw signals and classification performance. The chosen features according to their importance score are shown in Figure 1. The importance score is a value between 0 and 1, with higher values indicating that the feature is more important in determining the output class. The scores are computed by averaging the decrease in Gini impurity over all trees in the ensemble. Gini impurity is a measurement used to build decision trees to determine how the features of a dataset should split nodes to form the tree.

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2 \tag{1}$$

where D is the dataset containing the samples from k classes, while p_i is the probability of samples belonging to class i at a given node.

After the feature selection process, it was decided to use 6 sensors, 3 accelerometers on the right leg, left leg, and torso, and 3 magnetometers on the left leg, right leg, and right arm.

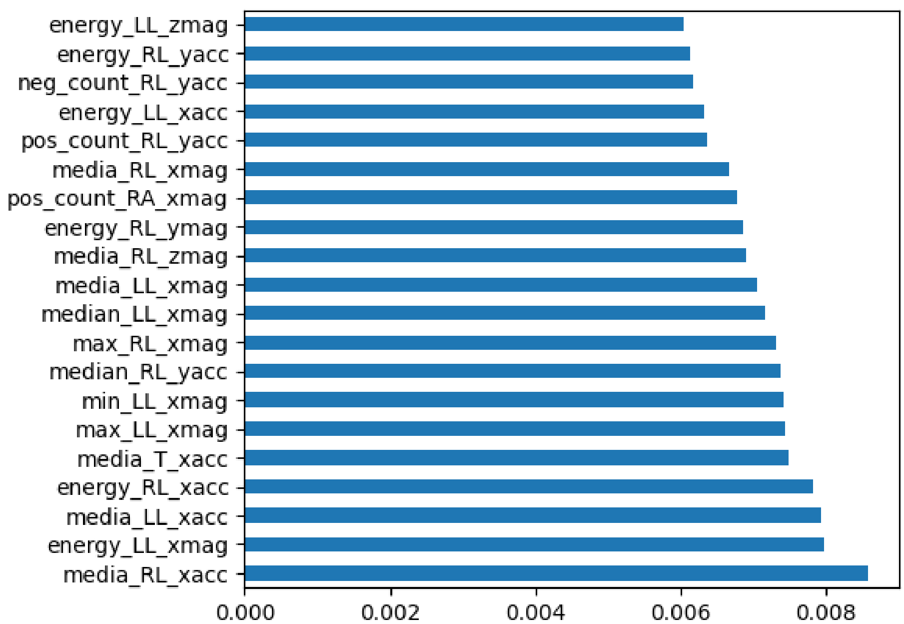
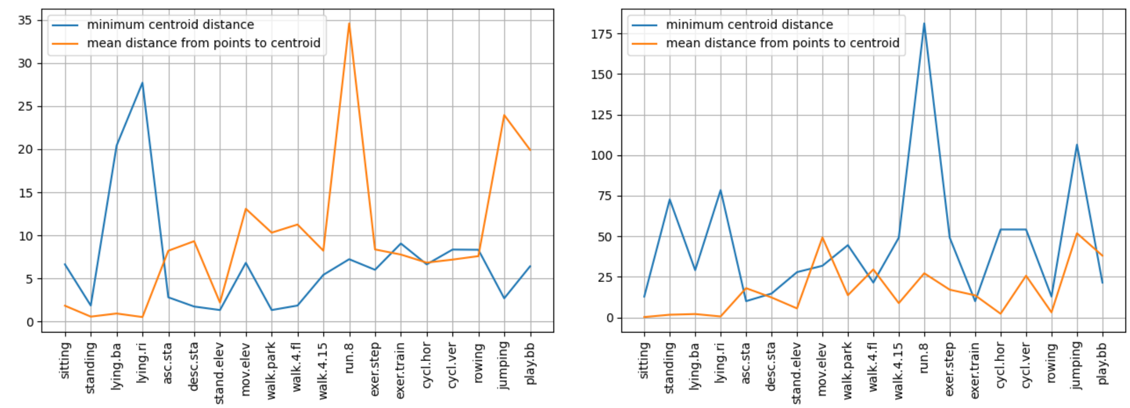


Figure 1: Selected features related to their importance, the name of the features are composed as $\{engineered\ feature\}\_\{used\ sensor\}$

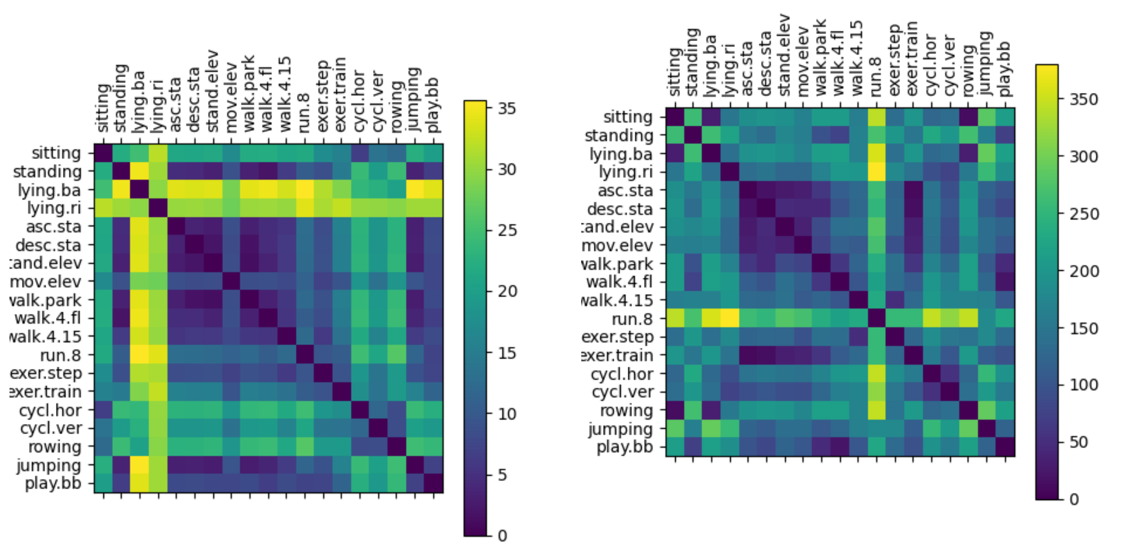
Figure 2a shows the minimum distance of centroids and the mean distance from points to centroids calculated on the raw signal. The mean distance of points from the centroid is higher than the minimum distance between the centroid and the other centroids, so the corresponding activity cannot be detected. Only the first 4 activities can be easily detected. Figure 2b shows that after feature engineering and feature selection, for most of the activities the minimum distance of the centroids is greater than the mean distance of the points from the centroids. This is not true for the following: "ascending stairs", "moving around in an elevator", "walking on a treadmill with a speed of 4 km/h in flat", "exercise on a cross trainer", "playing basketball", which will be difficult to detect.



(a) Centroids distances calculated on the raw signal

(b) Centroids distances after feature engineering and feature selection

Figure 2: Centroids distances before and after feature engineering and feature selection



(a) Distance between centroids of the activities calculated on raw signal

(b) Distance between centroids of the activities after feature engineering and feature selection

Figure 3: Distance between centroids of the activities before and after feature engineering and feature selection

Figure 3a shows the covariance matrix calculated on the raw signal, it can be seen that generally, the activities are more difficult to recognize than in figure 3b, which shows the covariance matrix of the activities after feature engineering and feature selection. Exceptions are "lying on back" and "lying on right side".

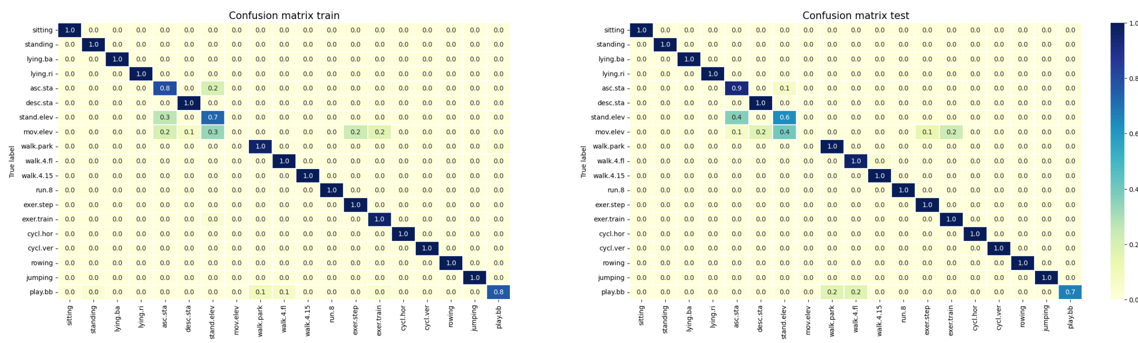
5 K-Means

K-Means is a clustering algorithm used in unsupervised machine learning. It is used to group similar data points together into clusters, based on their feature values. The algorithm works by initializing k centroids (one for each cluster) and then iteratively reassigning each data point to the cluster with the nearest centroid, while also updating the centroid positions to the mean of the points in the cluster. The process continues until the assignment of points to clusters no longer changes. The number of clusters (19) must be specified in advance.

The index of clusters at the output of K-Means is random. Not necessarily activity 1 corresponds to cluster 1. For the mapping, the centroid of the cluster was compared with the centroids of the activities computed on the training set, the cluster is then regarded as the class whose centroid is closest to the cluster centroid.

6 Evaluation metrics

6.1 Confusion matrix



(a) Normalized confusion matrix on training dataset

(b) Normalized confusion matrix on test dataset

Figure 4: Normalized confusion matrices

Figures 4a and 4b show the normalized confusion matrix for training and test datasets. A confusion Matrix for multiclass classification is used to know the performance of a Machine learning classification. It gives a comparison between actual and predicted values. The confusion matrix is a N x N matrix, where N is the number of activities.

6.2 Accuracy

Tables 3a and 3b respectively show the accuracy of the training and the test datasets.

The accuracy of a class can be calculated by the confusion matrix or by the formula 2

Activity	Accuracy
sitting	100%
standing	100%
lying.ba	100%
lying.ri	100%
asc.sta	80%
desc.sta	100%
stand.elev	73.33%
mov.elev	0%
walk.park	96.67%
walk.4.fl	100%
walk.4.15	100%
run.8	100%
exer.step	100%
exer.train	96.67%
cycl.hor	100%
cycl.ver	100%
rowing	100%
jumping	100%
play.bb	76.67%

(a) Accuracy on training dataset

activity	accuracy
sitting	100%
standing	100%
lying.ba	100%
lying.ri	100%
asc.sta	93.33%
desc.sta	100%
stand.elev	63.33%
mov.elev	0%
walk.park	100%
walk.4.fl	96.67%
walk.4.15	100%
run.8	100%
exer.step	100%
exer.train	100%
cycl.hor	100%
cycl.ver	100%
rowing	100%
jumping	100%
play.bb	66.67%

(b) Accuracy on test dataset

Table 3: Accuracy

$$Accuracy = \sum_{i=1}^{19} P(i|i)P(i) \tag{2}$$

"Moving around in an elevator" has never been detected, and it has been confused with "standing in an elevator still," "descending stairs," "exercising on a cross trainer," "ascending stairs," and "exercising on a stepper", this is because as shown in figure 2b the mean distance from points to the centroid is higher than minimum centroid distance, and it is possible that one or more signals useful for detecting such activity were discarded during feature selection.

7 Conclusions

This paper shows that the K-Means-based system was able to achieve a high level of performance in recognizing different human activities.

The use of the K-Means clustering technique is a simple and effective approach to human activity recognition. It can be applied to large datasets and classify activities with high accuracy. Additionally, the K-Means algorithm is computationally efficient and can be easily implemented in real-time systems.

References

[1] https://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities