

# Machine Learning Capstone Project

Human Activity Recognition (HAR) is a research that aims to develop systems to realize automatic recognition of physical activities to extract information about the user-behavior so that these systems can proactively improve user's experience and interaction with the computer. This is usually done by utilizing external sensors (e.g. environmental cameras), sensors on the user (e.g. wearables, body-worn sensors), or sensors on the objects that we interact with (e.g. smartphones). HAR systems successfully took place in products like Nintendo Wii for entertainment and Nike+ running shoes for fitness.

One challenge with the activity recognition that doesn't necessarily exist in object recognition or speech recognition is that HAR offers more degrees of freedom in terms of system design and implementation. Due to its temporal nature, it is not very clear what starts/ends and when. In other words, there is no common clear definition, grammar or structure of human activities that we can use to make a clear and generic problem statement (**Bulling et al., 2014**).

## Project Overview

The focus of this project is to recognize activities of daily living based on the motion related data acquired through a waist-mounted smartphone with embedded motion-sensitive sensors.

Dataset is consisted of the motion related data sampled from the activities of 30 people performing walking, walking-upstairs, walking-downstairs, sitting, standing, and laying and the activity labels given by the experimented based on the recorded video of the performed activities.

Source of the sampled data is the embedded sensor streaming data regarding 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz.

Sampled data were already pre-processed by applying noise filters and re-sampled with fixed-width sliding windows of length 128 (2.56 sec/window) and 50% overlap. In addition, linear acceleration data was filtered from gravitational acceleration as it is almost constant and carries no information regarding the activities. A Butterworth low-pass filter was applied to remove the acceleration data in frequency domain corresponding to a frequency of less than 0.3Hz.

Filtered and re-sampled data was then converted into a feature vector of size 561. Features are subtracted from the properties in temporal and frequency domain. These include first order statistical properties like minimum, maximum, mean, standard deviation, and other variability measures like mean absolute deviation, interquartile range, and some other statistical properties like auto-regression coefficients, and correlation.

## Problem Statement

In this project, the main goal is to classify the 561 dimensional feature space representing the daily life activities into the categories corresponding to the activities labeled as walking, walking-upstairs, walking-downstairs, sitting, standing, and laying. To achieve this goal, various machine learning methods are trained and their learning and prediction performances are compared.

## Metrics

To measure the classification performance of different methods, I used k-fold cross-validation and extracted precision, recall, f-score metrics for every single fold. I considered the weighted average of the metrics based on the number of class labels. This way, obtained metrics are more robust against the imbalance between different class labels.

I took the mean of these metrics for every single folding to obtain final metrics regarding the performance of the classification methods.

## Analysis

When I was having a closer look the distribution of the features, I plotted the correlation between the features and calculated the skewness of each feature. This helped me to decide on which features are the most relevant for the classification tasks.

## Algorithms and Techniques

As the dataset already includes labels, I decided to apply supervised machine learning methods. I applied various methods at first to decide on the baseline classification performance and to pick the one that performs the best to carry on with. I chose SVM and SGD as they returned the best default performance.

For the feature selection, I used KBest feature selection and PCA and finally I combined the features to see how the classification performance changes.

I had to decide on the KBest and PCA features and for that I utilized an exhaustive approach to see the classification performance for different KBest and nPCA component pairs.

## Discussion

### [TO BE ADDED]

(kbest 5)(pca_n 50)	precision: 0.94	recall: 0.93	fscore: 0.93	t_proc: 1.58	t_train: 1.16	t_test: 0.25
(kbest 5)(pca_n100)	precision: 0.95	recall: 0.94	fscore: 0.94	t_proc: 1.50	t_train: 1.81	t_test: 0.42
(kbest 5)(pca_n200)	precision: 0.95	recall: 0.95	fscore: 0.95	t_proc: 1.54	t_train: 3.65	t_test: 0.90
(kbest 10)(pca_n 50)	precision: 0.94	recall: 0.93	fscore: 0.93	t_proc: 1.51	t_train: 1.23	t_test: 0.27
(kbest 10)(pca_n100)	precision: 0.95	recall: 0.94	fscore: 0.94	t_proc: 1.52	t_train: 1.93	t_test: 0.45
(kbest 10)(pca_n200)	precision: 0.95	recall: 0.95	fscore: 0.95	t_proc: 1.58	t_train: 3.72	t_test: 0.92
(kbest 15)(pca_n100)	precision: 0.95	recall: 0.94	fscore: 0.94	t_proc: 1.56	t_train: 1.97	t_test: 0.47
(kbest 15)(pca_n200)	precision: 0.95	recall: 0.95	fscore: 0.95	t_proc: 1.69	t_train: 3.88	t_test: 0.94
(kbest 20)(pca_n100)	precision: 0.95	recall: 0.94	fscore: 0.94	t_proc: 1.75	t_train: 2.03	t_test: 0.48
(kbest 20)(pca_n200)	precision: 0.95	recall: 0.95	fscore: 0.94	t_proc: 1.67	t_train: 3.85	t_test: 0.96
(kbest 50)(pca_n100)	precision: 0.95	recall: 0.94	fscore: 0.94	t_proc: 2.14	t_train: 2.50	t_test: 0.61
(kbest 50)(pca_n200)	precision: 0.95	recall: 0.94	fscore: 0.94	t_proc: 2.26	t_train: 4.43	t_test: 1.11
(kbest100)(pca_n 40)	precision: 0.94	recall: 0.93	fscore: 0.93	t_proc: 3.14	t_train: 2.42	t_test: 0.60
(kbest100)(pca_n 50)	precision: 0.94	recall: 0.93	fscore: 0.93	t_proc: 3.13	t_train: 2.56	t_test: 0.63
(kbest100)(pca_n100)	precision: 0.95	recall: 0.95	fscore: 0.94	t_proc: 3.14	t_train: 3.34	t_test: 0.83
(kbest100)(pca_n200)	precision: 0.95	recall: 0.95	fscore: 0.95	t_proc: 3.24	t_train: 5.36	t_test: 1.37
(kbest200)(pca_n 30)	precision: 0.94	recall: 0.93	fscore: 0.93	t_proc: 5.47	t_train: 4.01	t_test: 1.03
(kbest200)(pca_n 40)	precision: 0.94	recall: 0.93	fscore: 0.93	t_proc: 5.50	t_train: 4.14	t_test: 1.06

(kbest200)(pca_n 50)	precision: 0.94 recall: 0.94 fscore: 0.93	t_proc: 5.52 t_train: 4.24 t_test: 1.08
(kbest200)(pca_n100)	precision: 0.95 recall: 0.94 fscore: 0.94	t_proc: 5.54 t_train: 5.04 t_test: 1.29
(kbest200)(pca_n200)	precision: 0.95 recall: 0.95 fscore: 0.94	t_proc: 5.56 t_train: 7.13 t_test: 1.85
(kbest561)(pca_n 2)	precision: 0.94 recall: 0.94 fscore: 0.94	t_proc: 17.95 t_train: 10.98 t_test: 2.77
(kbest561)(pca_n 5)	precision: 0.94 recall: 0.94 fscore: 0.94	t_proc: 17.98 t_train: 10.31 t_test: 2.64
(kbest561)(pca_n 10)	precision: 0.94 recall: 0.94 fscore: 0.94	t_proc: 17.98 t_train: 10.26 t_test: 2.59
(kbest561)(pca_n 15)	precision: 0.94 recall: 0.93 fscore: 0.93	t_proc: 17.97 t_train: 10.15 t_test: 2.56
(kbest561)(pca_n 20)	precision: 0.94 recall: 0.94 fscore: 0.93	t_proc: 17.96 t_train: 10.31 t_test: 2.60

## References

Bulling, Andreas, Ulf Blanke, and Bernt Schiele. "A tutorial on human activity recognition using body-worn inertial sensors." *ACM Computing Surveys (CSUR)* 46.3 (2014): 33.