## **İDENTİFY EMPLOYEES ACTİVİTİES**

## Motivation

One of the most important resources to manage in our company are employees. Human talent moves around the company’s facilities carrying out different tasks. Introducing mechanisms to facilitate those tasks for our employees will save time and at the end will save money.

## Description

We want to make use of some data gathered by each employee’s mobile device (in an autonomous way, non-invasive, no user interaction required) in order to automatically identify each employee while walking around. We also want to assess what the user is doing using the same data. This information could be the basis to optimize the way the user performs tasks and automatically granting him or her some permissions based on this implicit user recognition.

For this project we are going to use the following dataset:

<https://archive.ics.uci.edu/ml/machine-learning-databases/00240/UCI%20HAR%20Dataset.zip>

The time series for some sensors in the user’s mobile device (worn in the waist) have been windowed in 2.56 second (50% overlapping) and converted into 561 different features. 30 users have participated in the recording of the dataset and they have performed 6 different activities: WALKING, WALKING\_UPSTAIRS, WALKING\_DOWNSTAIRS, SITTING, STANDING, LAYING.

Download the dataset and read the README.txt file. This file contains a description of the data gathered and the contents of the different files in the dataset. The information for some users are contained in a training folder while other users are located in a testing folder. There are 2 files describing the different features based on the recorded sensor data. We do not need to understand how the features have been calculated. We can use these features as the starting point for our calculations.

**What each user is doing at each time window?**

Remove outliers from the information in the training set for WALKING and WALKING\_UPSTAIRS for all users in the training set. Use a 10% outlier factor (you can assess results with other values if desired to compare results).

Xtrain walking up outliers: 215

Xtrain walking up filtered: 858

Xtrain walking outliers: 184

Xtrain walking filtered:1042

Train a classifier to distinguish between WALKING and WALKING\_UPSTAIRS using the 561 features for the information in the training set without outliers. Although not required for this assignment, more activities could be added in order to have a more detailed assessment of what each user is doing each 2.56 second window.

according to code best hidden layer size = 80 , score = 1

Training set score data without outlier: 1.000000

1. Calculate the score for the training samples. Explain the results. Why is the score so high?

Training set score walking data with outlier: 0.997553

Training set score walkingup data with outlier: 0.994408

Training test score is so high because we train the model with this data set.

1. Remove the outliers for the WALKING and WALKING\_UPSTAIRS samples in the testing set and use the previously trained classifier to assess the score. Optionally add more activities if you decide to. What happens? Explain the results?

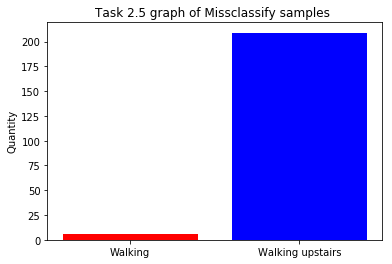
test set score walking data without outlier: 0.997625

test set score walkingup data without outlier: 0.946950

Compare with previous train set score, test score is lower because we calculate the score with the data that is not known by the model.

**Detect information of one particular activity inside the data labeled as belonging to a different activity**

Use the information of the training samples detected as outliers for the WALKING\_UPSTAIRS class in the previous classifiers and assess how many of these samples are classified as WALKING segments. This could be due to flat segments while climbing up stairs.



**Reducing the dimensionality for the input and controlling the complexity of the model (overfitting)**

Repeat the previous calculations using only the first 10 features. If using a classifier based on the MLP architecture, you could try to vary the number of neurons in the hidden layers as well. Explain the results.

Training set score data without outlier: 0.991579

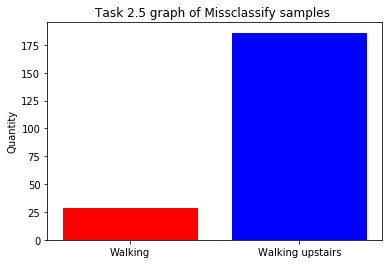
Training set score walking data with outlier: 0.982055

Training set score walkingup data with outlier: 0.963653

test set score walking data without outlier: 0.619952

test set score walkingup data without outlier: 0.920424

the result is similar . we used PCA in order to find the best 10 features. This Features represent all the feature in the dataset.

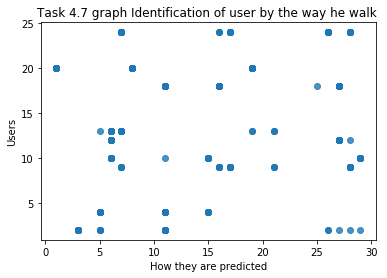


**User identification based on the way each user walks**

Concatenate all the walking samples in the training and test files for all the 561 features. Concatenate the subject\_train and subject\_test information for the same walking samples that we are going to use as labels to automatically identify each user based on the way he or she walks.

Test set score input walking data output subject data with outlier: 1.000000

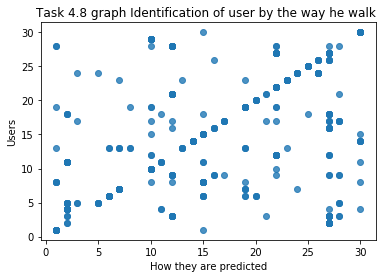
Test set score input walking data output subject data with outlier: 0.000000



random 70/30 train/test split

Train set score : 0.510373

Test set score : 0.446809



1. Train several classifiers on the training samples and validate their accuracy to identify users using the test samples. Explain the results.

Epoch 1/20

964/964 [==============================] - 2s 2ms/sample - loss: 3.0143 - acc: 0.2033

Epoch 2/20

964/964 [==============================] - 0s 118us/sample - loss: 2.2981 - acc: 0.4429

Epoch 3/20

964/964 [==============================] - 0s 116us/sample - loss: 1.6294 - acc: 0.6732

Epoch 4/20

964/964 [==============================] - 0s 112us/sample - loss: 1.1000 - acc: 0.8361

Epoch 5/20

964/964 [==============================] - 0s 119us/sample - loss: 0.7414 - acc: 0.9004

Epoch 6/20

964/964 [==============================] - 0s 115us/sample - loss: 0.5209 - acc: 0.9367

Epoch 7/20

964/964 [==============================] - 0s 115us/sample - loss: 0.3836 - acc: 0.9564

Epoch 8/20

964/964 [==============================] - 0s 114us/sample - loss: 0.2940 - acc: 0.9637

Epoch 9/20

964/964 [==============================] - 0s 113us/sample - loss: 0.2348 - acc: 0.9699

Epoch 10/20

964/964 [==============================] - 0s 118us/sample - loss: 0.1937 - acc: 0.9751

Epoch 11/20

964/964 [==============================] - 0s 113us/sample - loss: 0.1665 - acc: 0.9803

Epoch 12/20

964/964 [==============================] - 0s 187us/sample - loss: 0.1421 - acc: 0.9834

Epoch 13/20

964/964 [==============================] - 0s 118us/sample - loss: 0.1290 - acc: 0.9834

Epoch 14/20

964/964 [==============================] - 0s 117us/sample - loss: 0.1134 - acc: 0.9855

Epoch 15/20

964/964 [==============================] - 0s 117us/sample - loss: 0.1005 - acc: 0.9834

Epoch 16/20

964/964 [==============================] - 0s 112us/sample - loss: 0.0902 - acc: 0.9865

Epoch 17/20

964/964 [==============================] - 0s 116us/sample - loss: 0.0828 - acc: 0.9876

Epoch 18/20

964/964 [==============================] - 0s 117us/sample - loss: 0.0751 - acc: 0.9865

Epoch 19/20

964/964 [==============================] - 0s 113us/sample - loss: 0.0704 - acc: 0.9886

Epoch 20/20

964/964 [==============================] - 0s 111us/sample - loss: 0.0646 - acc: 0.9896

test loss : 0.056181

test accuracy : 0.990315