
```

%{
This function takes in the arrayPerActivity struct containing a
single table for each activity.
It labels this data by assigning a unique label to each class, which
will
eventually be the targets for SVMs.

Arguments
- `arrayPerActivity` -> struct containing a single array per
activity without any magnetometer data.

Returns
- `unlabelledInputs` -> just the input data
- `classLabels` -> [Nx1] class labels with a unique string for
each class
%}

function [unlabelledInputs, classLabels] =
getUnlabelledData(arrayPerActivity)
    % array containing the names of the activities.
    % These names will match the field names in the struct
    sets = ["LGW", "RA", "RD", "SiS", "StS"];
    % redefining the sets in a way that is compatible with SVM class
labelling
    sets_for_labels = [{'LGW'} {'RA'} {'RD'} {'SiS'} {'StS'}];

    % -----
    % Label the data by having a single column dedicated for string
class labels
    % -----
    for ff = 1 : length(sets)
        % obtain the number of rows in the current dataset
        sample = arrayPerActivity(1).(sets{ff});
        temp_labels = cell(length(sample),1);

        % we want one column of labels, each row labelled using a
string
        % according to the target class
        temp_labels(:,1) = sets_for_labels(ff);

        % store all the targets as one long 1D array
        if ff == 1
            classLabels = temp_labels;
            unlabelledInputs = sample;
        else
            classLabels = vertcat(classLabels, temp_labels);
            unlabelledInputs = vertcat(unlabelledInputs, sample);
        end
    end
end
end

```