# Task

To develop a software system for recognition of activities of daily living (ALDs) by applying computational models. You will develop your recognition system using a dataset with signals from accelerometer and gyroscope collected from six activities (1: Walking, 2: Walking upstairs, 3: Walking downstairs, 4: Sitting, 5: Standing, 6: Laying).

For this task, you will use the Human Activity Recognition datasets (humanADLs.zip file) available in the Moodle page of the EE52036. The datasets are built from the recordings of 30 subjects performing ADLs while carrying a waist-mounted smartphone (Samsung Galaxy S II) with embedded inertial sensors.

# Data Preprocessing

* Inspect data
  + Inspect skewness (histograms)
  + Identify anomalies (box and whiskers plots and scatter plots) X
  + Check class balances (bar charts/pie charts) X
  + Identify missing data and it’s type X
  + Identify other noisy data (See If there is any precedence in literature)
* Preprocess data
  + Remove anomalies/Ignore anomalies/impute anomalies?
  + Balance classes (Classes already balanced) X
  + Deal with missing data (None present) X
  + Deal with noisy data
  + Standardize and normalize (Features already normalized and standardized) X
  + Encode actions (Already encoded) X
  + Data smoothing?
* Feature Engineering
  + Apply transformation (Already Normalized so no need?)
  + Polynomial features?
  + Check for interaction terms and combine if necessary, X Not necessary if applying PCA because that is done in the dimensionality reduction
  + Window techniques for time series features?
  + Build features X Not necessary with PCA
  + Dimensionality reduction (PCA/SVD) X Applied PCA
  + Feature Selection X Not necessary, applied PCA
  + Correlation heatmaps and Pearsons correlation coefficient X Not necessary, Applied PCA

# Training

* Train/test/val split with shuffle X
* Can we include Ensemble learning with NN? E.g. boosting

Model 1: Random Forrest X

Model 2: Basic NN X

Model 3: TBC. Probably something like a RNN/transformer

# Evaluation and visualization

* Confusion matrix X
* Learning curves X
* Precision, Recall, F1 Score X
* Cross validation. (Leave one Out? Random forrest takes too long so thinking to stick to cross validation)

# Optimization

* Hyperparameter tuning of best model
* Fix any over/under fitting
* Bayesian Optimization backed up by grid search?
* Add early stopping
* Bias and variance graphs
* Regularization
* Dropout