# Assessment 3: ML for Human Activity Recognition

MATH5836: Data and Machine Learning

## Project Goal

The goal of this project is to train a model of choice on the *Human Activity Recognition* (HAR) dataset to predict a person's activity based on high-dimensional sensor data (561 features). The project also demonstrates the application of two methods that achieve significant dimensionality reduction while preserving model performance.

### Theoretical Tasks

1. Let the data matrix $X \in \mathbb{R}^{n\times d}$ be standardized. Show that
$$\widehat{\Sigma} = \frac{1}{n-1} X^\top X$$
is the sample correlation matrix of the data.

2. Show that $\widehat{\Sigma}$ is a positive semi-definite matrix, and that its singular value decomposition (SVD) must be of the form
$$\widehat{\Sigma} = \sum_{j=1}^{d} \lambda_j \mathbf{u}_j \mathbf{u}_j^\top, \tag{1}$$
where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$ are the eigenvalues of $\widehat{\Sigma}$ and $\mathbf{u_j}$ is an orthonormal eigenvector corresponds to the $j$-th eigenvalue $\lambda_i$.

### Modeling Tasks

1. **Data Loading:** The Human Activity Recognition (HAR) dataset (commonly referred to as the "UCI HAR Dataset" on OpenML) contains sensor measurements from smartphones (accelerometer and gyroscope) worn by participants performing six activities:

   - Class 1: Walking
   - Class 2: Walking Upstairs
   - Class 3: Walking Downstairs
   - Class 4: Sitting
   - Class 5: Standing
   - Class 6: Laying

   Key Characteristics:

   - Samples: $10,299$ observations
   - Features: 561 pre-engineered features derived from raw sensor signals (e.g., time-domain statistics, frequency-domain Fast Fourier Transform coefficients).
   - Classes: 6 (activities listed above).
   - Purpose: Classify human activities based on sensor data.

   You can load this dataset using:

```python
from sklearn.datasets import fetch_openml

har = fetch_openml(name="har", version=1, as_frame=True)
X, y = har.data, har.target.astype(int)
```

   **References:** (1) Paper; (2) UCI Page;

2. **Preprocessing & EDA:** Preprocess the data by inspecting and handling missing values, outliers, and categorical variables (if any of them exists). Check class-wise characteristics of the data. By computing the correlation between the target variable and the features, identify, analyse and visualise at least 10 key features.

3. **Basic Modeling:** Consider the following two models to train:

   - A dense neural network with at least two hidden layers, each containing at least 10 neurons.
   - A random forest classifier.

   You may fine-tune hyperparameters such as the network size or the maximum tree depth based on your available computational resources.

   Run at least 5 independent experiments using different random seeds, and report the mean $\pm$ standard deviation of Accuracy, F1-score, and AUC-ROC. In each experiment, split the data into 80% training and 20% testing. Be sure to standardize the data separately for each experiment.

4. **Dimensionality Reduction using Correlation:** For each $k \leq 561$, let $\widetilde{X}_k \in \mathbb{R}^{n \times k}$ denote a new data matrix formed by selecting the $k$ features (i.e., columns of $X$) that are most strongly correlated (positively or negatively) with the response $y$. Repeat the modeling task above (using your chosen model) on the reduced dataset $(\widetilde{X}_k, y)$, varying $k$ over the set $[100, 200, 300, 400, 500]$. Plot the average performance (e.g., accuracy, F1-score, AUC-ROC) versus $k$, and briefly discuss your observations.

5. **Dimensionality Reduction using SVD:** This task is similar to the one above, but uses SVD instead of correlation for dimensionality reduction. In each experiment, after standardizing $X$, compute the SVD of the correlation matrix $\widehat{\Sigma}$ as defined in (1):

$$\widehat{\Sigma} = \sum_{j=1}^{d} \lambda_j \mathbf{u}_j \mathbf{u}_j^\top.$$

   For each $k \leq 561$, let $U_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{d \times k}$ denote the matrix formed by the top $k$ eigenvectors corresponding to the largest $k$ eigenvalues of $\widehat{\Sigma}$. Construct the reduced data matrix as

$$\widetilde{X}_k = X U_k U_k^\top.$$

   Fit your chosen model on the transformed dataset $(\widetilde{X}_k, y)$, varying $k$ over $[100, 200, 300, 400, 500]$. Plot the average performance versus $k$, and discuss your findings.

6. **Summary:** Provide a detailed summary of your approach and your findings, and make a conclusion.