```
clear;      % clear the workspace
clc;        % clear the command window
close all;  % close all popup windows

% ======================================
% ================ TASK 1 ==============
% ======================================

fprintf("\n--------\n Importing the data...\n--------\n")
rawData = importDatasets();

% [OPTIONAL] Try out different params for the low pass filter
% experimentFilters(rawData, 7, 100)

fprintf("\n--------\n Filtering the data...\n--------\n")
filteredRawData = filterData(rawData);

fprintf("\n--------\n Extracting the time-domain features...\n--------
\n")
processedData = extractFeatures(filteredRawData);

fprintf("\n--------\n Organising the features...\n--------\n")
arrayPerActivity = organiseFeatures(processedData);

fprintf("\n--------\n Labelling the data...\n--------\n")
[unlabelledData, classLabels] = getUnlabelledData(arrayPerActivity);
[labelledData] = getLabelledData(arrayPerActivity);

% [OPTIONAL] Plot the 7 extracted time domain features of a section
% plotTimeDomain("foot_l_gyro_x", labelledData)

fprintf("\n--------\n Training the ANN using all the features...
\n--------\n")
allFeaturesNNAccuracy = crossValidateNN(labelledData, 5,
 201, "trainscg");
close all

fprintf("\n--------\n Training the SVM using all the features...
\n--------\n")
allFeaturesSVMAccuracy = svmPosterior(labelledData,
 classLabels, "polynomial", 1);


% ======================================
% ================ TASK 2 ==============
% ======================================

% delete vars no longer needed to save memory
clearvars -except labelledData unlabelledData classLabels

fprintf("\n--------\n Finding the top 15 features...\n--------\n")
```

```
fifteenFeaturesLabelled = find15Features(labelledData, unlabelledData,
 classLabels);

% [OPTIONAL] Train a NN for the top 15 only
% fprintf("\n--------\n Training the ANN using 15 features...
\n--------\n")
% fifteenFeaturesNNAccuracy = crossValidateNN(fifteenFeaturesLabelled,
 5, 10, "trainscg");

% [OPTIONAL] Train an SVM for the top 15 only
% fprintf("\n--------\n Training the SVM using 15 features...
\n--------\n")
% fifteenFeaturesSVMAccuracy = svmPosterior(fifteenFeaturesLabelled,
 classLabels, "polynomial", 1);

% ======================================
% ================ TASK 3 ==============
% ======================================

fprintf("\n--------\n Finding features from a single segment...
\n--------\n")
[segmentFeaturesLabelled] = extractSegment("thigh_r", labelledData);

% ======================================
% ================ TASK 4 ==============
% ======================================

fprintf("\n--------\n Training the ANN using features from a single
 segment...\n--------\n")
singleSegmentNNAccuracy = crossValidateNN(segmentFeaturesLabelled, 5,
 35, "trainscg");

fprintf("\n--------\n Training the SVM using features from a single
 segment...\n--------\n")
singleSegmentSVMAccuracy = svmPosterior(segmentFeaturesLabelled,
 classLabels, "polynomial", 1);
```

Published with MATLAB® R2020a