**Protocol for Big Data Science Examination Project: Classification - Activity Recognition in Goats**

We aim to classify the activities of goats that wore devices that recorded their movements. The dataset has already been pre-processed.

Step 1: Data Preparation

The first step of the data preparation is to remove duplicates and irrelevant or empty features. Next, the capitalisation of text is standardised to prevent wrongful classifications. The data is formatted to avoid numbers being interpreted as text or decimal points being wrongly interpreted, and categorical variables are converted into numerical values. Afterwards, outliers are detected and removed (taking precautions not to remove valid but extreme values) using multivariate methods (e.g., one-class SVM or Elliptic Envelope) as the data is highly dimensional. If the data is noisy, binning or clustering methods can be explored to improve data quality. (It must be remembered that the data has already been binned during pre-processing.) Missing data is handled next, accounting for MCAR, MAR, or MNAR instances, deletion or single imputation methods can be explored. Following this, the data is partitioned. Remembering that the explored classifiers utilise hyperparameters, a nested cross-validation approach will be explored to optimise the hyperparameters. Lastly, the training data sets are normalised, and the resulting normalisation parameters are used to normalise the corresponding validation and test data sets.

Step 2: Classification

Next, we will compare at least three different prediction models using several performance measures to evaluate the models. We will also report the runtimes of training the models. Care will be taken not to use irrelevant features to train our models. Some classifier models we can compare are:

1. Random Forest: this model operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees.
2. Support Vector Machines (SVM): SVM tries to find the best decision boundary to segregate n-dimensional space into classes. Different kernels can be explored, and to use an SVM model as a multi-class classifier, the one-vs-one and one-vs-all approaches can be compared.
3. Quadratic Discriminant Analysis (QDA): assumes normality but relaxes the assumption that the covariance matrix is the same in each class. It models the probability distribution of each class using a quadratic decision boundary to classify new data into one of the predefined classes.
4. k-Nearest Neighbours (k-NN): a classification technique that considers k-nearest neighbours from the training set and predicts the class for the new data point based on the class of most of its k-nearest neighbours.

Step 3: Dimensionality Reduction (DR)

We will explore at least three DR mechanisms and compare the model's performance with the baseline models. Some methods we can explore are:

* Feature Selection: In this approach, we select a subset of the original features based on some criterion. There are several (un)supervised techniques available for feature selection, such as:
  + - Wrapper techniques such as Forward and Backward feature selection.
    - Embedded techniques such as Decision tree and SVM weight feature selection.
* Feature Extraction: This approach transforms the original features into a new set. Some of the commonly used feature extraction techniques are:
  + Linear such as Linear Discriminant Analysis (LDA, supervised) and Principal Component Analysis (PCA, unsupervised)
  + Non-linear, such as Locally Linear Embedding (LLE, unsupervised)

Step 4: Model Evaluation

We will explore the following performance measures to evaluate our models:

* Accuracy: the ratio of correctly classified instances to the total number of predictions.
* Precision: the ratio of true positives to the sum of true and false positives.
* Recall: the ratio of true positives to the sum of true positives and false negatives.
* F-score: a measure of a test's accuracy calculated from the precision and recall.