Sprint 1 – action classification through key points

In the first sprint of the Machine Learning project, you are tasked with training an action recognition model on key point data. The dataset for this sprint has been constructed out of the student recordings at iGent.

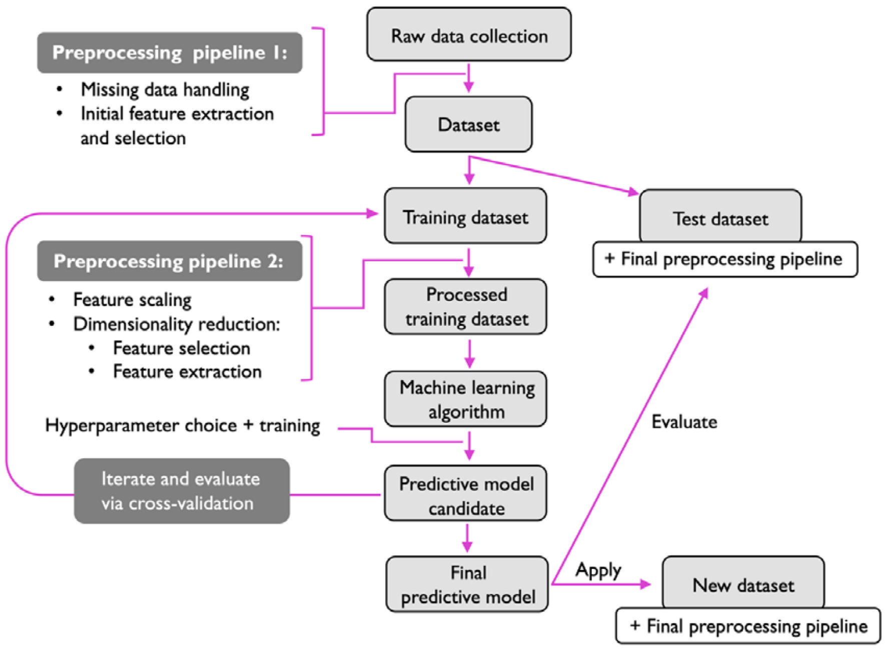
Submission and grading

Your work is graded according to the soundness of the methodology, your understanding of the procedure and your ability to critically evaluate your obtained results. We do not grade for the accuracy on the classifier.

You submit a **Jupyter notebook** which contains all code and output, and that covers the different questions below. You also submit a **PDF version** of the notebook with output. Make sure to deliver a notebook that is organized in subsections that reflect the different steps of the workflow, and that contains sufficient explanations and discussions. If the notebook is too large, you can submit different notebooks.

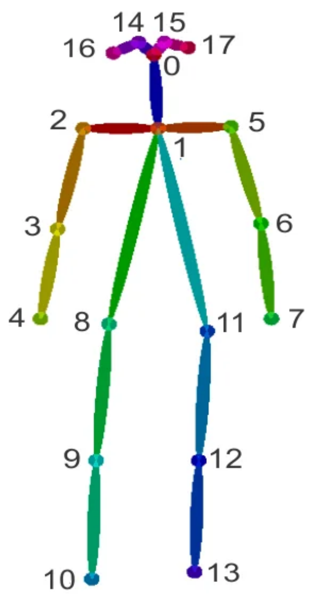
Workflow

Your notebook should follow the workflow that was discussed during theory class, and for which a schematic overview is shown below. Since the dataset was already curated by the authors, some steps of the workflow need not be executed.


Raw data collection
Preprocessing pipeline 1:
Missing data handling
Initial feature extraction and selection
Dataset
Training dataset
Test dataset
+ Final preprocessing pipeline
Preprocessing pipeline 2:
Feature scaling
Dimensionality reduction:
Feature selection
Feature extraction
Processed training dataset
Machine learning algorithm
Hyperparameter choice + training
Evaluate
Iterate and evaluate via cross-validation
Predictive model candidate
Final predictive model
Apply
New dataset
+ Final preprocessing pipeline

Dataset

All recordings at iGent, each consisting of 450 frames, have been processed into arrays of size (450, 18, 3). For each frame, 18 key points have been generated, made up of three values: the x-coordinate, the y-coordinate, and a confidence value. The following figure gives an overview of the ordering of key points.



0. Nose
1. Neck
2. Right Shoulder
3. Right Elbow
4. Right Wrist
5. Left Shoulder
6. Left Elbow
7. Left Wrist
8. Right Hip
9. Right Knee
10. Right Ankle
11. Left Hip
12. Left Knee
13. Left Ankle
14. Right eye
15. Left Eye
16. Right Ear
17. Left Ear

Exploratory data analysis

As the raw data already has been pre-processed into standardized files, this step can be rather short.

Some questions you can explore:

- What is contained in the name of each file?
- How can you filter data by action, or by group?
- Is the data balanced?
- What are the ranges for each feature?
- Do all (values for all) features make sense? Are there any outliers?

Split your data in a train set and a hold-out test set

To assess the generalization capabilities of your **final** model, you can use a hold-out test set. Note that the test set is used **only at the end** of your training pipeline, and not during iterations (i.e. model selection and tuning).

A straightforward approach to split the dataset in a train set and a test set is by random split — stack all frames and randomly divide these into two subsets. However, by applying this strategy to the recording data, the frames of a single action by a single student may (will probably) end up in both subsets. This *leaks* information from the train to the test set, and might give the false impression that your model generalizes well on unseen data. To mitigate this, ensure all data from a single recording is contained in one subset (train set or test set).

Preprocessing pipeline 2

Again, due to the standardized format of the dataset, this step can be rather short. However, some common techniques can be explored, such as normalization or min-max scaling.

Some additional steps you can explore:

- Are there big (noticeable) differences between recordings of the same action?
- Can you reduce the variability between recordings?

Machine learning algorithm

Choose a classifier

In the sprint workshop, you trained an SVM on a subset of the data. Select three additional models (preferably from different model *families*). Elaborate on the choice for each model.

Hyperparameter choice + training

Train each model, and search for the optimal hyperparameters. As stated before, do this using **only the train set**. Optimizing on the test set would also give a false impression of generalization.

Optimal hyperparameters should be found using cross-validation. There exist multiple forms of cross-validation. Note that the "default" form (which randomizes samples) cannot be used here. Apply an appropriate form of cross-validation, and elaborate on your choice.

Select a number of metrics for classification performance, and explain.

Predictive model candidate

Analysis

Examine the best performing variant of each model. How do these stack against each other?

Feature importance

Not all features are equally important. In this case, one could argue that the "face" key points are less important than the "limb" key points, for example. By removing less important features, the dimensionality of the dataset can be reduced, possibly resulting in the training of a better model.

For your single best model, investigate which features are most important.

Re-iterate with selected features

- Re-train your best model using only the important features,
- using cross-validation to find the optimal hyperparameters.
- Compare the results to the initial model that used all features. What do you observe?

Final evaluation

The final step is the first step wherein you use the test set. Apply the appropriate pre-processing steps and evaluate your best model (either the top model using all features, or using the most important features). Compare the results to the cross-validation results in training.

Manual evaluation

Inspect the results:
- Construct a confusion matrix.
- Visualize some misclassifications.
- Can you explain these?
- Are there any disadvantages to using a frame-by-frame model?

Can you come up with additional steps to further explore the results?