TÉCNICO LISBOA

Machine Learning
MMAC, MECD
Project Report
2025/2026 – 1st Semester, P1

Authors: André Antunes Amaro (106419), Catarina Sofia da Silva Gonçalves (103989)
Group: 18

1 Introduction

The purpose of this project was to identify the most suitable models for each task, balancing accuracy, simplicity, and computational efficiency to prevent overfitting. The aim was to develop models that generalize effectively to unseen data rather than performing well only on the training set. This report summarizes the approaches adopted and analyzes the corresponding results.

2 Regression with Non-linear Models

2.1 Objective

We sought to replace an expensive and high-precision sensor with a software model that estimated its output from a set of 6 cheap sensors. Their measurements are related non-linearly with the target measurement. The dataset we were given includes 700 entries, where X of shape $(700, 6)$ represents the measurements from the low-cost sensors, and Y of shape $(700,)$ contains the associated values from the reference sensor.

2.2 Data analysis

In order to analyze the dataset, we began by exploring the order of magnitude of the values from the sensors and the target Y, and observing the feature correlation matrix.

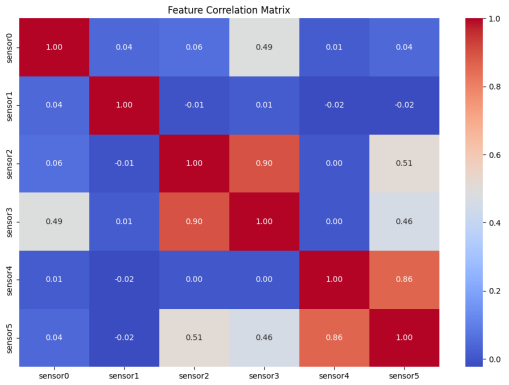

Figure 1: Feature correlation matrix.

From there, we concluded that most sensors presented small correlations. For instance, sensor 1 appeared to not correlate to any other sensor. On the other hand, sensors 2 and 3 showed high association, akin to sensors 4 and 5. We opted to lean more on the conservative side of not removing features, as we worried this tendency would not reproduce in the test set, and the feature set was rather small.

Following that, we plotted both the feature-target correlation relationship, and the scatterplot of Y in function of each sensor.

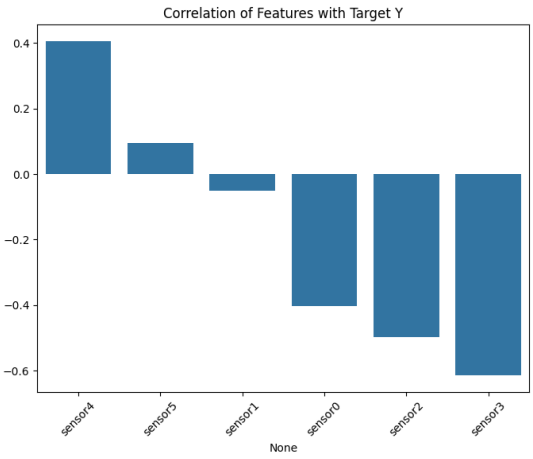

Figure 2: Feature-target correlations.

Sensors 0, 2, 3 and 4 stood out as the most influential in explaining the target.

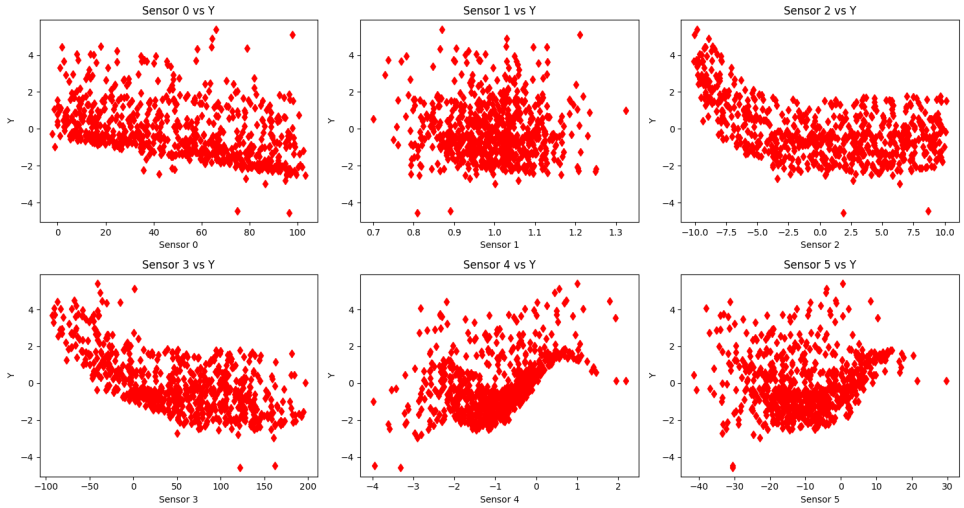

Figure 3: Y scatterplot in function of each sensor.

These scatterplots portray the kind of relationships one would expect following the previous graph. It is both sensor 1 and 5 that show the least modeling suitability, whereas the other sensors suggest Y is linked to them by some type of curve-like behavior.

Afterwards, we proceeded to the training and validation splitting through a standard 70% - 30% proportion. Then, we made sure that both sets contained the more extreme-valued Y observations (around -4 and 5) that stand out in Figure 3. Note that they were not interpreted as true outliers since it was announced that there were no outliers in the dataset.

2.3 Models

The challenge of input scale imbalance was addressed with the inclusion of `StandardScaler` in all model pipelines. This increased R^2 and decreased MSE among every model, which were measured in the validation set to ensure overfitting was being avoided. The mostly high magnitudes of R^2 hinted that low-signal-to-noise ratios were not predominant in the dataset, likely due to its synthetic origin.

To determine the optimal hyperparameters for each model, we performed a manual iterative search over usual value choices for several combinations, after sequentially finding the best order of magnitude for each. This was done through a 5-fold cross-validation in the 70% training portion, to ensure robustness. This methodology was later improved upon in the classification tasks by the use of the package `Optuna` for intelligent searches.

We now display the scores on the validation set for each of the 6 tested models.

Model	R^2	MSE
Polynomial regression with ridge regularization	0.939791	0.137066
Polynomial regression with lasso regularization	0.815377	0.420293
Polynomial kernel ridge regression	0.945102	0.124976
RBF kernel ridge regression	0.984935	0.034296
Polynomial support vector regression	0.979667	0.046288
RBF support vector regression	0.984847	0.034496

Table 1: R^2 and MSE scores on the validation set for several models

Among the linear models, polynomial regression with ridge regularization performed substantially better than the lasso variant, hinting that the data benefited from coefficient shrinkage without feature sparsity. This reinforced our decision of not removing sensor 1.

The polynomial kernel models also performed well ($0.945 < R^2 < 0.980$), though slightly below their RBF counterparts, implying that the latter provided a better bias–variance trade-off for this dataset. Both the RBF kernel ridge and RBF support vector models reached the highest R^2 (≈ 0.985) and the lowest MSE (≈ 0.034). The following plot displays the prediction effectiveness of the former, which achieved the slightly better scoring.

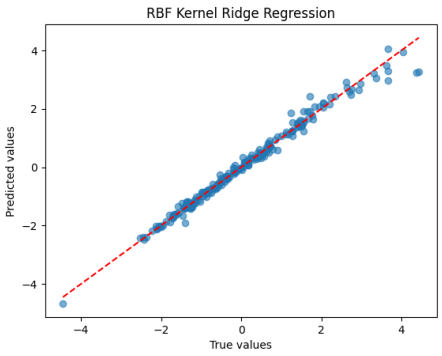

Figure 4: Predicted vs. True Values plot for the RBF kernel ridge regression.

It appears that the underlying relationship between the sensors and the target is indeed non-linear, and that the RBF kernel effectively captured this complexity while maintaining generalization. Hence, we opted for the RBF kernel ridge regression as our final model.

The main improvement we propose to this section of our work is the inclusion of the whole dataset in the training of the final model, instead of the 70% portion from the training-validation split. Additionally, it could also be relevant to check if the removal of features would improve scoring on the test set.

3 Classification of Physical Therapy Sessions

This part of the project focused on analyzing video recordings from physical therapy sessions involving 18 stroke survivors. Each participant was filmed performing various functional exercises designed to replicate everyday activities. In total, 33 keypoints across the entire body were extracted through MediaPipe. The training samples were collected from 14 out of the 18 patients.

3.1 Exercise Recognition

3.1.1 Objective

The first classification problem was a multiclass task aimed at developing a model capable of distinguishing among three exercises: $E1$ (brushing hair), $E2$ (brushing teeth), and $E5$ (hip flexion). We were given access to 700 observations, each described by features representing the mean position and standard deviation of the 33 keypoints recorded during the exercise. The patient numerical identification was also provided for each observation.

3.1.2 Data analysis

Much like in the previous task, our first course of action was to inspect and investigate our data. We started by plotting the mean positions of most key points for some observations, in order to obtain a skeleton-like plot.

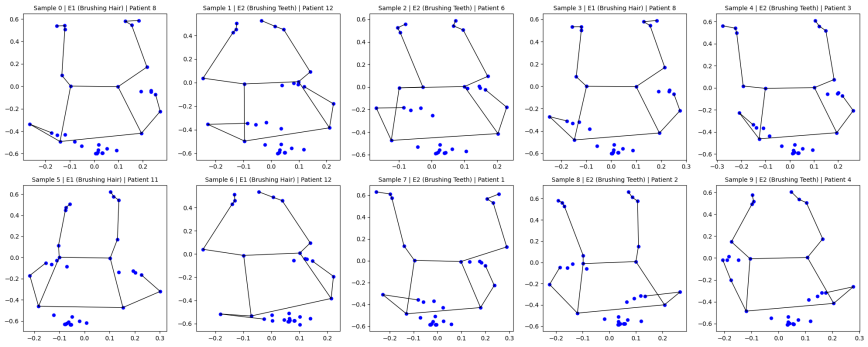

Figure 5: Mean position of key points in observations 0-9 (before pre-processing).

Aiming to standardize body size among patients, we normalized the data through the function `Normalize`. It divides the mean position and standard deviation of every keypoint, by the mean shoulder width to mean torso length ratio of the patient, on a per-observation basis. To reduce the amount of different body poses, all the right-sided movements were mirrored into left-sided movements with the function `Flip`.

The t-SNE between patients and exercises was also plotted, which allowed us to notice that $E1$ and $E2$ had extremely similar observations in our dataset. For instance, patients 1, 9 and 10 have overlapping observations in these exercises. On the contrary, most observations of $E5$ are distinguishable from the other exercises, as shown in the top of the plot.

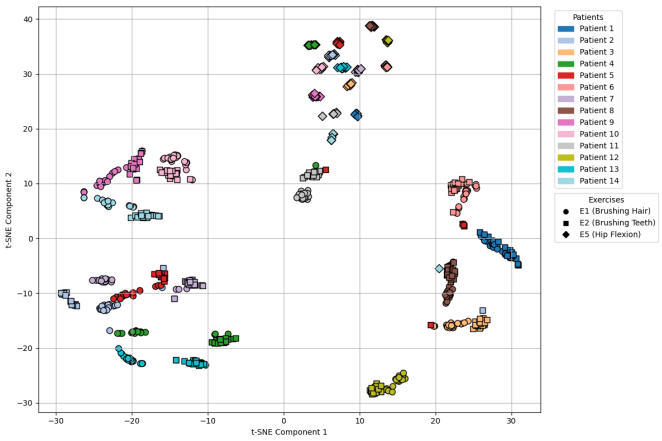

Figure 6: t-SNE across patients and exercises.

In order to improve the distinction between $E1$ and $E2$, we tested various new feature sets. The function `SaySide` indicates the side with most movement (thus, where the motor pattern is being executed). `PositiveOnly` removes all key points from the negative (right) side. `NoNegativeLeg` removes the key points associated with the right leg. `Distance` computes the euclidean distance between two key points, while `AngleAtb` calculates the angle formed by three points. Different combinations of these features were integrated in the models.

3.1.3 Models

Although the data exhibited some class imbalance, specifically in class $E5$, which contained fewer samples, this class was easily distinguishable from the others. Therefore, the imbalance was not considered a major concern.

Since the dataset included only 14 patients, we chose not to create separate training and validating sets. Instead, we applied cross-validation, as it provides a reliable estimate of model performance while maximizing the use of available data. To avoid training and validating on data from the same patients, which could lead to overfitting, we employed `GroupKFold` to split the dataset by patient into seven folds, each containing 12 training patients and 2 validation patients. The same approach was followed in the binary classification task.

The performance of each model was evaluated through the mean macro $F1$ score and Accuracy across all folds. These scores were consistently lower in folds that contained patients with problematic observations regarding distinction between the first two exercises, which was to be expected and in accordance with the t-SNE plot. Nevertheless, we opted to still include all patients (and observations) in the final model training. The best hyperparameters for each model were found through `Optuna`'s intelligent search.

The following results summarize the average cross-validation performance across the 4 tested models.

Model	Mean macro $F1$	Mean Accuracy
Random Forest	0.822	0.800
Logistic Regression	0.851	0.833
Support Vector Machine with RBF kernel	0.917	0.903
Neural Networks (Multilayer perceptron)	0.824	0.807

Table 2: Mean macro $F1$ and Accuracy average cross-validation scores for several models

Not all models benefited equally from the inclusion of extra feature sets: for example, Logistic Regression achieved its best results using only the baseline features, whereas the SVM with RBF kernel showed a clear performance gain when combining the `Normalize`, `Flip`, and `SaySide` transformations ($0.887 \rightarrow 0.917$ macro mean $F1$). In contrast, the Random Forest not only underperformed comparatively, but was also slower to train. This suggests that the SVM model was better able to exploit the richer feature representations introduced by these preprocessing steps. Consequently, we settled with this model for submission.

The confusion matrices produced by the SVM with RBF kernel for each train-validation patient fold are shown below.

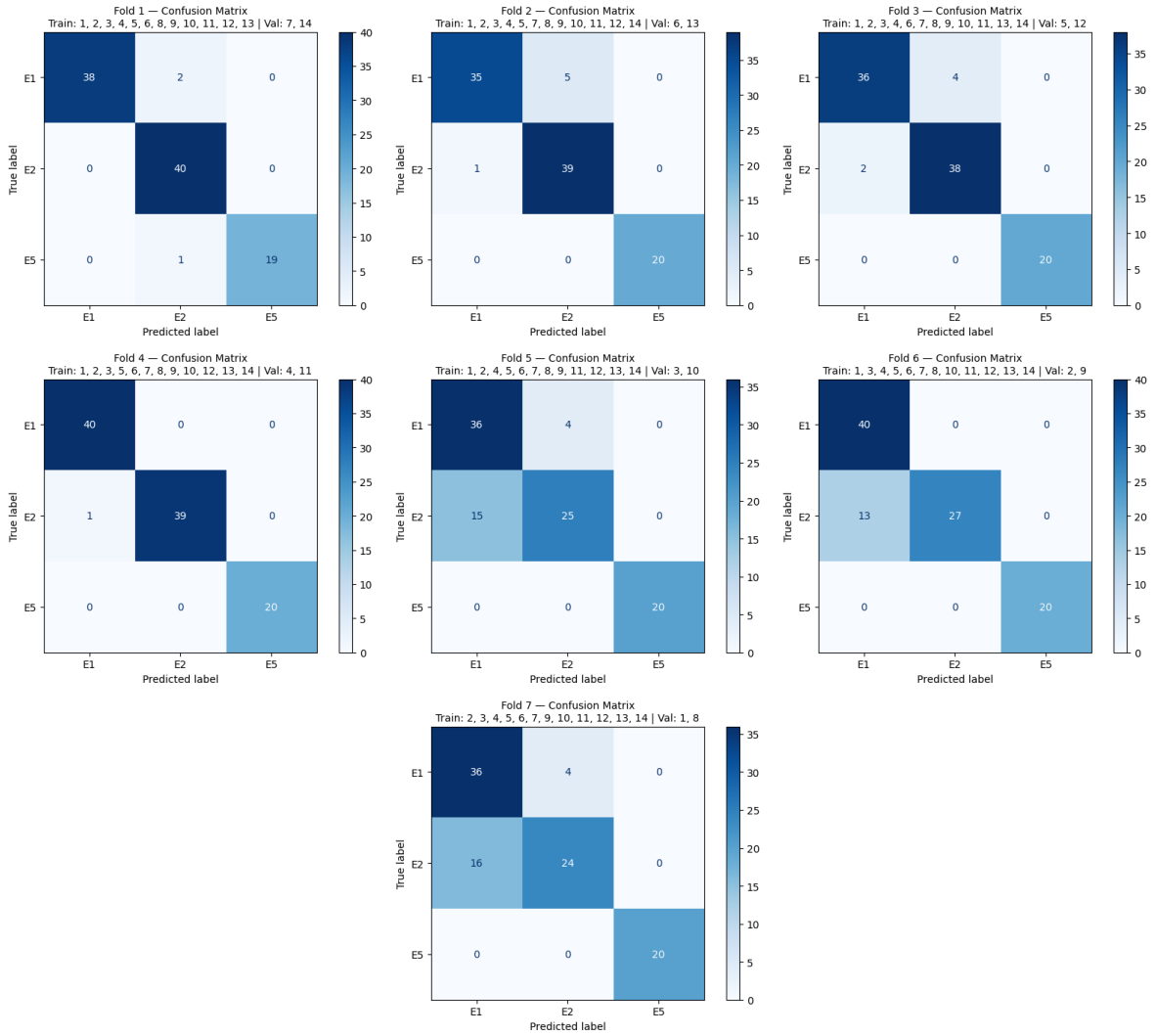

Figure 7: Confusion matrices produced by the SVM with RBF kernel model.

For most folds, the model was able to classify each exercise mostly accurately. With a single misclassification of $E5$ across all folds, it is likely that this class helped keep the $F1$ score fairly high, as it is consistently greater than Accuracy for every model. The folds that yielded the weakest classification performance corresponded to validations on patients 10, 9, and 1 — an outcome consistent with the clusters observed in Figure 6.

The considerable amount of wrong class attributions implies that the preprocessing could have been improved further, namely through feature selection, i.e., removing noisy and/or redundant points, such as those from the face, hands and feet. Implementing a more robust data splitting scheme, such as one based on all possible combinations of `GroupKFold`'s folds, could further strengthen our approach.

3.2 Identification of the impaired side

3.2.1 Objective

The second classification task was binary, focusing on predicting which side of the patient's body was affected by the stroke—left (label 0) or right (label 1). This time, our data frame did not contain a specific data matrix with features. Instead, it was comprised of 444 observations, each including the patient and exercise identification, as well as a sequence of frames (the recording), each displaying the position of the 33 keypoints.

3.2.2 Data analysis

In order to fully understand the data frame, we started by examining its structure and checking the number of observations for each exercise, as well as the number of affected patients per side. This revealed that the data was imbalanced with respect to both of these aspects. To better visualize each recording sequence, we animated a reduced selection of the main key points, and colored healthy and affected sides in opposing tones.

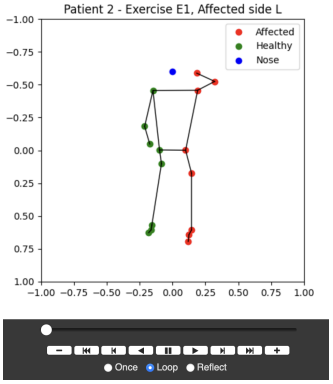

Figure 8: Screen capture of the animation setup, for a given observation.

The class imbalance issue was tackled by flipping the recordings and corresponding labels of patients 2 and 8, resulting in a balanced dataset with 7 patients affected on each side. Initially, the models were trained on the imbalanced data to verify whether any "hard to predict" observations existed on the left side. Since all left-sided cases were correctly classified, we randomly selected two right side affected patients and flipped their data to achieve balance. Patient 1 was deliberately excluded from this process, as visual inspection of the animations showed that they performed the exercises with the affected side almost as well as most others did with their healthy side.

Since skeleton sequences from different observations had different lengths, we applied cubic spline interpolation to get n equally spaced frames across all exercises. Similarly to the previous classification task, the `Normalize` function was applied to the keypoint's coordinates, after modifying it accordingly.

Concerning feature engineering, we explored two different approaches:

- Setting $n = 5$ for the spline interpolation and selecting the main (4 - 6) keypoints for each exercise, adding one or two additional features (such as `SaySide`) to improve performance;
- Creating a comprehensive set of features based on relevant distances, angles, velocities, moving averages, asymmetries, and ranges of motion.

3.2.3 Models

Since we held a high amount of features—approximately half of the amount of observations—we primarily trained tree-based models. SVM with RBF kernel was also trained, since it was the model with the best results from the previous task.

To get a patient-level label from multiple observations over time, we trained an individual classifier for each exercise, and aggregated their outputs using a weighted-voting system, in order to integrate complementary information from different exercises. This allowed us to limit feature dimensionality and simplify the models. Given the skewed sample counts ($E1$: 51, $E2$: 54, $E3$: 111, $E4$: 115, $E5$: 113), we merged the first two exercises into a single model to mitigate imbalance, since they exhibited high similarity (as shown in the preceding task).

In order to set up the weighted-voting system, each prediction of each model was combined using a weighted soft voting approach. Each model's weight was determined based on its mean balanced accuracy across validation folds, and converted into normalized weights with the softmax function. For each patient, the predicted probabilities from all models were aggregated as a weighted sum, and the final prediction was determined by thresholding at 0.5.

As in the previous task, to prevent training and testing on data from the same patients—thereby avoiding overfitting—we used `GroupKFold` to split the dataset by patient into 14 folds. Each fold included 13 patients for training and one for validation, following a leave-one-out strategy. The hyperparamenters that yielded the best Balanced Accuracy were again determined by `Optuna`'s intelligent search.

The first feature engineering approach granted better cross-validation scores across all models, which are displayed in the following table.

Model	$E1 + E2$	$E3$	$E4$	$E5$	Ensemble
Random Forest	0.286	0.500	0.286	0.571	0.571
Logistic Regression	0.500	0.571	0.500	0.571	0.500
CatBoost	0.714	0.571	0.429	0.500	0.714
XGBoost	0.643	0.714	0.643	0.500	0.571
SVM with RBF kernel	0.929	0.714	0.857	0.929	0.929

Table 3: Mean Balanced Accuracy cross-validation scores for several models

The SVM with RBF kernel achieved the best performance across all exercises. In contrast, tree-based methods showed higher variability across exercises, likely due to the limited number of samples and the relatively high feature dimensionality, which can increase the risk of overfitting. They were also considerably slower to train. Logistic Regression, being a linear model, yielded near-baseline scores, confirming the need for non-linear classifiers.

The ensemble approach produced balanced accuracies comparable to or slightly lower than the best individual model. By assigning larger weights to models with higher cross-validation performance, the ensemble improved robustness and reduced the impact of individual model errors, leading to more stable and reliable patient-level predictions.

The only misclassified patient in the SVM with RBF kernel model was patient 1, which aligns with our previous observation that this patient performed the movements with the affected side better than the other patients.

The results on the evaluation testing set showed that our model could be improved. One possible enhancement would be to increase the size of the training data by applying data augmentation techniques, such as adding flipped versions of each patient's data to effectively double the dataset. Additionally, exploring different combinations of features or integrating new feature types could help the model capture more relevant information and improve its generalization ability. Further experimentation with other ensemble strategies, such as stacking, could also enhance performance and robustness on unseen patients.

4 Conclusion

The methods chosen from this work for submission produced effective regression and classification results, demonstrating low overfitting and strong computational efficiency, while keeping the approach simple and robust. In the regression task, the RBF kernel ridge regression model effectively captured the underlying non-linear relationships between features and the target, outperforming all other models. For the classification tasks, the SVM with RBF kernel consistently yielded the best results, demonstrating the suitability of kernel-based approaches for complex human motion data.

The main challenges were inter-patient variability and data imbalance in the final task, which required careful preprocessing and validation to ensure fair evaluation. The results were insightful yet indicated room for improvement, highlighting the potential of these machine learning approaches.

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.