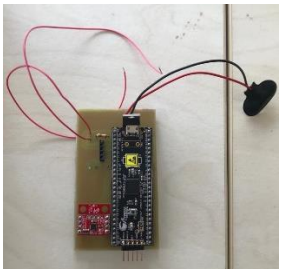
ACQUISITION PROTOCOL

code used for this purpose: `acquisitions_training_classifier.ipynb`

Prerequisites

- Right-handed players only.
- Position of the PCB within the case for acquisitions as shown in the picture below (IMU facing the game net in the forehand grip).



Training phase

Number of subjects: 4

Duration of each acquisition: about 5 seconds

Gestures acquired and related labels set:

- Forehand : ‘0’
- Backhand: ‘1’
- Serve: ‘2’
- No shot: ‘3’

Number of acquisitions for each shot (‘forehand’, ‘backhand’, ‘serve’) → 10 for each subject.
Number of acquisitions for the ‘no shot’ gesture → 20 acquisitions for 1 subject only.
Number of acquisition files → 140 (in the “acquisitions_for_TRAINING” folder of the GitHub repository)

Features engineering

For each `acquisition_file.csv`, containing 400 rows and 7 columns (6 IMU measurement + 1 label), we computed the most significant statistic parameters of every IMU measurement:

- 1° quantile (0.25)
- Median
- 3° quantile (0.75)

In this way, we summarized each `acquisition_file.csv` in 19-dimensional array.

Having 140 `acquisition_file.csv`, the resulting training dataset is a (140, 19) dataframe.

Trained classifiers

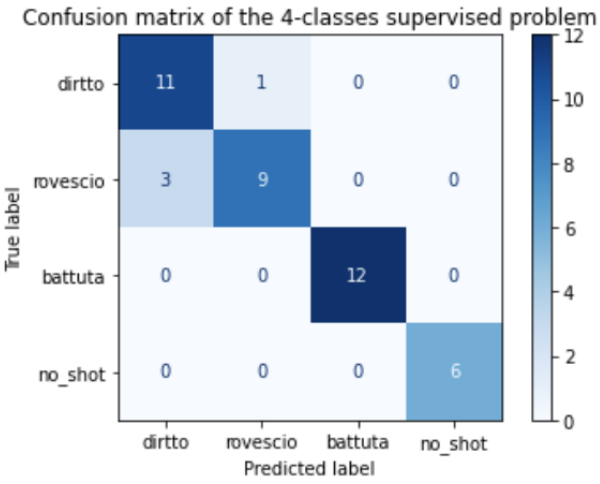
- Decision Tree:

```
f1_train: 0.847222 using {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 8, 'min_samples_split': 10}
f1_test:  0.9047619047619048
[[11  1  0  0]
 [ 3  9  0  0]
 [ 0  0 12  0]
 [ 0  0  0  6]]
              precision    recall  f1-score   support

         0.0       0.79      0.92      0.85        12
         1.0       0.90      0.75      0.82        12
         2.0       1.00      1.00      1.00        12
         3.0       1.00      1.00      1.00         6

    accuracy                           0.90        42
   macro avg       0.92      0.92      0.92        42
weighted avg       0.91      0.90      0.90        42

CPU times: user 306 ms, sys: 23.8 ms, total: 330 ms
Wall time: 2.03 s
```

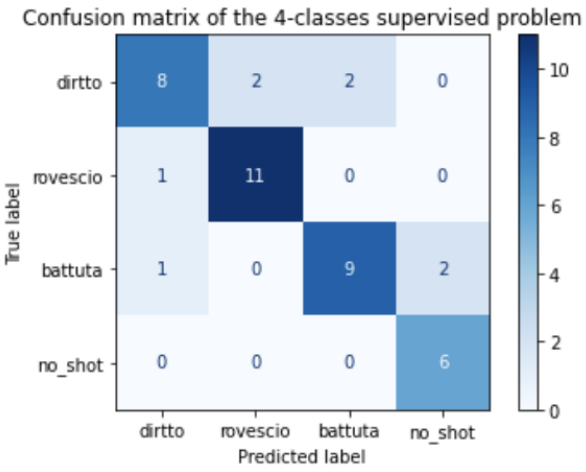


- Logistic Regression:

```
f1_train: 0.806818 using {'C': 10, 'max_iter': 2000}
f1_test:  0.8095238095238095
[[ 8  2  2  0]
 [ 1 11  0  0]
 [ 1  0  9  2]
 [ 0  0  0  6]]
              precision    recall  f1-score   support

         0.0       0.80      0.67      0.73        12
         1.0       0.85      0.92      0.88        12
         2.0       0.82      0.75      0.78        12
         3.0       0.75      1.00      0.86         6

    accuracy                           0.81        42
   macro avg       0.80      0.83      0.81        42
weighted avg       0.81      0.81      0.81        42
```

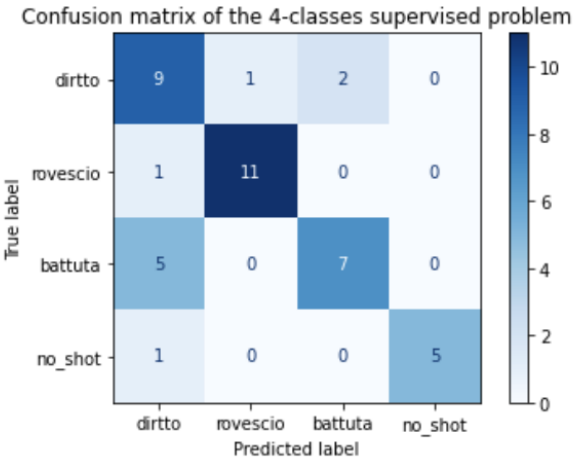


- KNN:

```
f1_train: 0.704861 using {'n_neighbors': 10}
f1_test:  0.7619047619047619
[[ 9  1  2  0]
 [ 1 11  0  0]
 [ 5  0  7  0]
 [ 1  0  0  5]]
              precision    recall  f1-score   support

         0.0       0.56      0.75      0.64        12
         1.0       0.92      0.92      0.92        12
         2.0       0.78      0.58      0.67        12
         3.0       1.00      0.83      0.91         6

    accuracy                           0.76        42
   macro avg       0.81      0.77      0.78        42
weighted avg       0.79      0.76      0.77        42
```



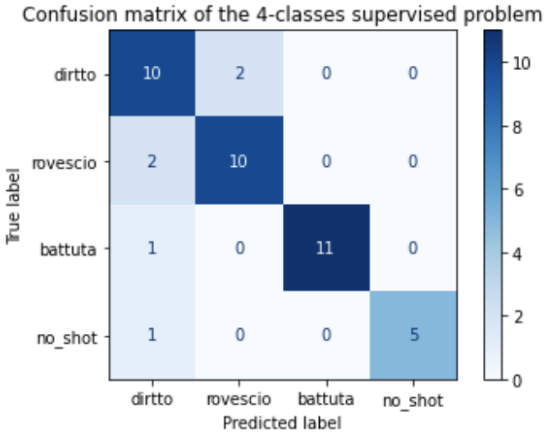
- Random Forest:

```
f1_train: 0.969381 using {'criterion': 'gini', 'max_depth': 15, 'min_samples_leaf': 2, 'min_samples_split': 3, 'n_estimators': 10}
f1_test:  0.9523809523809523
[[10  2  0  0]
 [ 0 12  0  0]
 [ 0  0 12  0]
 [ 0  0  0  6]]

              precision    recall  f1-score   support

         0.0       1.00      0.83      0.91        12
         1.0       0.86      1.00      0.92        12
         2.0       1.00      1.00      1.00        12
         3.0       1.00      1.00      1.00         6

    accuracy                           0.95        42
   macro avg       0.96      0.96      0.96        42
weighted avg       0.96      0.95      0.95        42
```

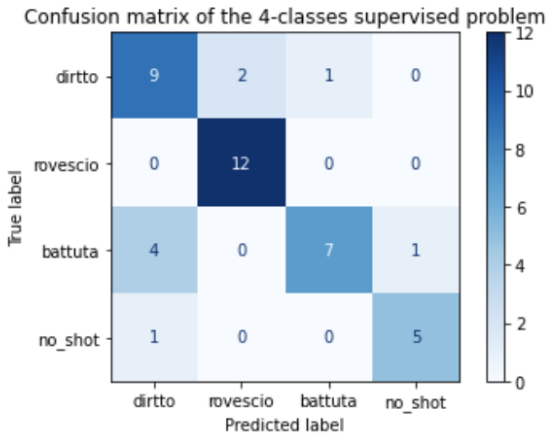


- SVM

```
f1_train: 0.755366 using {'C': 100, 'kernel': 'rbf'}
f1_test:  0.7857142857142857
[[ 9  2  1  0]
 [ 0 12  0  0]
 [ 4  0  7  1]
 [ 1  0  0  5]]
              precision    recall  f1-score   support

         0.0       0.64      0.75      0.69        12
         1.0       0.86      1.00      0.92        12
         2.0       0.88      0.58      0.70        12
         3.0       0.83      0.83      0.83         6

    accuracy                           0.79        42
   macro avg       0.80      0.79      0.79        42
weighted avg       0.80      0.79      0.78        42
```



Scaling the data did not lead to significant improvement in the performance of the classifiers, despite improving over fitting in Logistic Regression and KNN; thus, we decided to use the unscaled dataset.

Among the classifiers, we chose **Random Forest** because of:

✓ High F1 score for both training set and test set.
✓ Similar F1 scores for training sets and test sets, meaning no overfitting problems.
✓ Interpretability of the classifier.