

AdaBoost testen

December 26, 2020

```
[1]: import sys
sys.path.append('../')
from ortho_lib3_Copy2 import *
import pandas as pd
import numpy as np
import copy
import pandas as pd
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

```
[2]: exercises = Exercises.load('../Pickle/
↳def_exercises_sliced_transformed_data_all_categories.pickle')
# exercises = exercises.drop_category(1)
exercises.df
```

```
[2]:      angle_left_shoulder_xz_max_AF  angle_left_shoulder_xz_max_RF  \
0                2.705811                1.200405
1                2.757520                1.381570
2                2.691818                1.372206
3                2.524043                0.687781
4                2.640875                1.189572
..                ...                ...
102               2.420555                1.532106
103               1.201943                0.428302
104               2.186121                0.580735
105               1.882206                0.808175
106               2.602110                1.114584

      angle_right_shoulder_xz_max_AF  angle_right_shoulder_xz_max_RF  \
0                2.880950                1.022868
1                2.693907                1.300333
2                2.597640                1.352112
3                2.581135                0.931032
4                2.606757                1.312657
..                ...                ...
102               2.390046                1.290075
103               2.347702                0.429440
```

104	2.082403	0.712122
105	2.064096	0.987518
106	2.614965	1.054871

	angle_left_shoulder_yz_max_AB	angle_right_shoulder_yz_max_AB	\
0	2.536825	2.785053	
1	2.729818	2.894459	
2	2.312296	2.432454	
3	2.606495	2.639872	
4	2.666926	2.706126	
..	
102	2.591977	2.675785	
103	0.897281	2.104353	
104	2.649783	2.220724	
105	2.459425	2.348511	
106	2.574107	2.385383	

	diff_x_wrist_std_EL	diff_x_wrist_std_AF	diff_x_wrist_std_RF	\
0	0.051140	0.045686	0.047964	
1	0.046985	0.032754	0.022710	
2	0.018752	0.043444	0.059608	
3	0.045662	0.032816	0.036892	
4	0.042752	0.048538	0.042671	
..	
102	0.067242	0.053520	0.045416	
103	0.131204	0.461456	0.041716	
104	0.022393	0.089273	0.157484	
105	0.055195	0.085915	0.043581	
106	0.037779	0.128872	0.047050	

	diff_x_elbow_std_EL	...	angular_acc_xz_elbow_r_mean_AF	\
0	0.066650	...	0.013454	
1	0.035020	...	0.009875	
2	0.035862	...	0.012321	
3	0.030882	...	0.008630	
4	0.015893	...	0.007462	
..	
102	0.028010	...	0.021898	
103	0.038488	...	0.030021	
104	0.014435	...	0.036877	
105	0.025565	...	0.026531	
106	0.029085	...	0.032568	

	angular_acc_xz_elbow_r_std_AF	angular_acc_xz_elbow_r_mean_RF	\
0	0.009997	0.009142	
1	0.007717	0.009636	
2	0.009420	0.010026	

3	0.007913	0.007300
4	0.006183	0.007197
..
102	0.022046	0.029374
103	0.039972	0.009998
104	0.036002	0.044232
105	0.030848	0.030147
106	0.019488	0.023850

	angular_acc_xz_elbow_r_std_RF	angular_vel_yz_elbow_l_std_AB \
0	0.008593	0.035317
1	0.008744	0.017855
2	0.008610	0.028611
3	0.007061	0.021368
4	0.006843	0.026717
..
102	0.035492	0.069536
103	0.007187	0.013514
104	0.048294	0.066856
105	0.026745	0.042285
106	0.021694	0.057128

	angular_vel_yz_elbow_r_std_AB	angular_acc_yz_elbow_l_mean_AB \
0	0.034019	0.010569
1	0.019760	0.009327
2	0.027344	0.009416
3	0.021825	0.006564
4	0.023068	0.007782
..
102	0.066380	0.033381
103	0.040844	0.009843
104	0.062815	0.036638
105	0.048514	0.019478
106	0.066880	0.029559

	angular_acc_yz_elbow_l_std_AB	angular_acc_yz_elbow_r_mean_AB \
0	0.008540	0.009179
1	0.008234	0.008376
2	0.007159	0.008765
3	0.007230	0.007175
4	0.011197	0.007356
..
102	0.031042	0.032047
103	0.008446	0.017797
104	0.035517	0.042551
105	0.018303	0.019738
106	0.022851	0.030719

	angular_acc_yz_elbow_r_std_AB
0	0.008611
1	0.006633
2	0.007664
3	0.007337
4	0.005814
..	...
102	0.036677
103	0.019753
104	0.033384
105	0.018876
106	0.026552

[107 rows x 78 columns]

```
[3]: exp = Experiment(exercises, y_condition= lambda y: y != 'Category_2')
      columns = exp.df.columns.to_numpy()
```

```
[4]: from sklearn.ensemble import AdaBoostRegressor
      from sklearn.datasets import make_regression
      from sklearn.model_selection import cross_val_score, KFold, StratifiedKFold
      from sklearn.model_selection import train_test_split
      from sklearn.model_selection import StratifiedKFold
      from sklearn.metrics import mean_squared_error
```

```
[5]: X = exp.df.values
      y = exp.y
```

0.1 testset ophalen??

```
[6]: dffs_test_1_234 = create_dfframes(['Category_1', 'Category_2', 'Category_3', 'Category_4'],
                                     extype = ['AB', 'AF', 'RF', 'EL'],
                                     data_dir = '../sliced_transformed_testdata',
                                     print_errors=True)
      exercises_test_1_234 = dffs_to_exercises(dffs_test_1_234)

      exercises_test_2_34 = exercises_test_1_234.drop_category(1)

      exercises_test_2_34.df
```

```
VBox(children=(HTML(value=''), IntProgress(value=0, max=198)))
```

```

[6]:      angle_left_shoulder_xz_max_AF  angle_left_shoulder_xz_max_RF  \
0          2.375600          0.515718
1          1.832651          0.571748
2          2.279437          0.892147
3          2.853551          1.391743
4          2.409568          1.093059
5          3.141593          1.113963
6          2.564573          1.060095
7          2.695787          1.893329
8          2.531666          0.982060
9          2.598072          0.780502
10         2.532649          1.405655
11         1.712433          0.510527
12         2.818956          1.538357
13         1.589591          0.785398
14         2.527191          0.967928
15         2.306109          1.069718
16         3.012543          1.297603
17         1.237203          0.822418
18         2.576012          1.108928

      angle_right_shoulder_xz_max_AF  angle_right_shoulder_xz_max_RF  \
0          2.307687          0.415887
1          1.948294          0.650395
2          2.200986          0.913145
3          1.874065          0.996709
4          2.426248          1.013376
5          3.139295          1.253438
6          2.260703          0.883332
7          2.351426          1.709610
8          1.303295          0.926478
9          2.495837          0.821747
10         2.571669          1.468946
11         2.753829          0.566612
12         2.365642          0.952572
13         2.774457          1.408955
14         2.572233          0.955154
15         2.588732          1.337642
16         3.068278          1.408069
17         2.171692          0.780059
18         2.072947          1.087017

      angle_left_shoulder_yz_max_AB  angle_right_shoulder_yz_max_AB  \
0          2.272950          2.120081
1          1.363127          1.292968
2          2.312750          2.271171
3          2.618332          2.024207

```

4	2.583189	2.687324
5	2.482531	2.750692
6	2.673822	2.334830
7	2.485588	1.984378
8	1.608480	0.833220
9	2.333954	2.315521
10	2.399539	1.556696
11	1.663674	3.041914
12	2.728522	2.566286
13	1.330913	2.481952
14	2.399797	2.418613
15	2.333508	2.780054
16	2.813693	2.590100
17	1.519678	2.371818
18	2.574280	2.044648

	diff_x_wrist_std_EL	diff_x_wrist_std_AF	diff_x_wrist_std_RF	\
0	0.022516	0.123650	0.066065	
1	0.046210	0.054191	0.073046	
2	0.062190	0.110703	0.037969	
3	0.034831	0.754436	0.105197	
4	0.121865	0.061636	0.043011	
5	0.119332	0.045775	0.062743	
6	0.052364	0.071169	0.046293	
7	0.141243	0.322232	0.048486	
8	0.056350	0.753814	0.046766	
9	0.046412	0.075304	0.041178	
10	0.069697	0.075780	0.076076	
11	0.042222	0.432843	0.049091	
12	0.080415	0.242830	0.136882	
13	0.141904	0.641800	0.316515	
14	0.019926	0.054665	0.098079	
15	0.238379	0.044046	0.175252	
16	0.207649	0.093720	0.073282	
17	0.113686	0.182995	0.066054	
18	0.038098	0.238048	0.062802	

	diff_x_elbow_std_EL	...	angular_acc_xz_elbow_r_mean_AF	\
0	0.030044	...	0.005253	
1	0.039214	...	0.038027	
2	0.032260	...	0.016534	
3	0.026717	...	0.004213	
4	0.059019	...	0.024748	
5	0.034795	...	0.004806	
6	0.035378	...	0.006766	
7	0.036942	...	0.009332	
8	0.029941	...	0.021308	

9	0.022483	...	0.008440
10	0.115790	...	0.011655
11	0.037710	...	0.012845
12	0.037511	...	0.007610
13	0.043694	...	0.010392
14	0.021813	...	0.013922
15	0.077931	...	0.020187
16	0.028185	...	0.035963
17	0.041324	...	0.055863
18	0.012775	...	0.029496

	angular_acc_xz_elbow_r_std_AF	angular_acc_xz_elbow_r_mean_RF \
0	0.005825	0.005341
1	0.042545	0.028577
2	0.024424	0.017195
3	0.004100	0.004242
4	0.030197	0.017554
5	0.004547	0.007481
6	0.006339	0.008618
7	0.009368	0.026822
8	0.034503	0.013319
9	0.009668	0.012224
10	0.010952	0.012228
11	0.011820	0.008859
12	0.006633	0.007474
13	0.012070	0.013671
14	0.011704	0.023424
15	0.019269	0.033001
16	0.026881	0.069154
17	0.051294	0.031281
18	0.027702	0.038873

	angular_acc_xz_elbow_r_std_RF	angular_vel_yz_elbow_l_std_AB \
0	0.005245	0.044149
1	0.037112	0.075874
2	0.019296	0.024688
3	0.003969	0.023464
4	0.016981	0.080747
5	0.006636	0.027039
6	0.009020	0.030327
7	0.018314	0.037603
8	0.013228	0.030278
9	0.009866	0.033245
10	0.009263	0.068260
11	0.007660	0.038511
12	0.006700	0.032489
13	0.011276	0.018208

14	0.031715	0.015827
15	0.030522	0.058416
16	0.072434	0.067100
17	0.030357	0.047952
18	0.033683	0.089861

	angular_vel_yz_elbow_r_std_AB	angular_acc_yz_elbow_l_mean_AB \
0	0.038997	0.008397
1	0.077269	0.032440
2	0.024603	0.007676
3	0.015371	0.005916
4	0.092344	0.014680
5	0.032143	0.008069
6	0.028319	0.006566
7	0.024352	0.012050
8	0.013986	0.011730
9	0.030319	0.007741
10	0.050020	0.016117
11	0.080184	0.012062
12	0.028551	0.007851
13	0.053140	0.007542
14	0.020158	0.007464
15	0.092668	0.018463
16	0.062882	0.029627
17	0.058858	0.028998
18	0.084309	0.032859

	angular_acc_yz_elbow_l_std_AB	angular_acc_yz_elbow_r_mean_AB \
0	0.009601	0.009458
1	0.029207	0.029826
2	0.008410	0.008046
3	0.007493	0.006083
4	0.021800	0.016240
5	0.005797	0.008373
6	0.005273	0.006544
7	0.017120	0.010629
8	0.011056	0.007357
9	0.008933	0.008987
10	0.015967	0.012980
11	0.014230	0.019489
12	0.008183	0.008377
13	0.008003	0.013570
14	0.007306	0.007421
15	0.016758	0.025989
16	0.028091	0.031143
17	0.029493	0.032410
18	0.030546	0.034543

	angular_acc_yz_elbow_r_std_AB
0	0.010961
1	0.038971
2	0.010082
3	0.006979
4	0.024133
5	0.006443
6	0.006436
7	0.010221
8	0.006739
9	0.008939
10	0.015511
11	0.016729
12	0.006823
13	0.015099
14	0.007320
15	0.028522
16	0.027868
17	0.037222
18	0.030614

[19 rows x 78 columns]

```
[7]: exp_test_1_234 = Experiment(exercises_test_1_234, y_condition= lambda y: y != 'Category_1')
      columns = exp_test_1_234.df.columns.to_numpy()

      exp_test_2_34 = Experiment(exercises_test_2_34, y_condition= lambda y: y != 'Category_2')
      columns = exp_test_2_34.df.columns.to_numpy()
```

```
[8]: X_test_1_234 = exp_test_1_234.df.values
      y_test_1_234 = exp_test_1_234.y

      X_test_2_34 = exp_test_2_34.df.values
      y_test_2_34 = exp_test_2_34.y
```

```
[9]: regr_1_234 = AdaBoostRegressor(n_estimators=370, learning_rate = 1.9)
      regr_1_234.fit(X, y)

      regr_2_34 = AdaBoostRegressor(n_estimators=400, learning_rate = 1.9)
      regr_2_34.fit(X, y)

      ypred_1_234 = regr_1_234.predict(X_test_1_234)
      report_1_234 = classification_report(y_test_1_234, ypred_1_234.round(),
      output_dict=True)
```

```
ypred_2_34 = regr_2_34.predict(X_test_2_34)
report_2_34 = classification_report(y_test_2_34, ypred_2_34.round(),
    ↳output_dict=True)
```

```
[10]: report_1_234, report_2_34
```

```
[10]: ({'0.0': {'precision': 0.1,
    'recall': 0.16666666666666666,
    'f1-score': 0.125,
    'support': 6},
    '1.0': {'precision': 0.6666666666666666,
    'recall': 0.5263157894736842,
    'f1-score': 0.5882352941176471,
    'support': 19},
    'accuracy': 0.44,
    'macro avg': {'precision': 0.3833333333333333,
    'recall': 0.3464912280701754,
    'f1-score': 0.35661764705882354,
    'support': 25},
    'weighted avg': {'precision': 0.5306666666666666,
    'recall': 0.44,
    'f1-score': 0.4770588235294118,
    'support': 25}},
    {'0.0': {'precision': 0.625,
    'recall': 0.7142857142857143,
    'f1-score': 0.6666666666666666,
    'support': 7},
    '1.0': {'precision': 0.8181818181818182,
    'recall': 0.75,
    'f1-score': 0.7826086956521738,
    'support': 12},
    'accuracy': 0.7368421052631579,
    'macro avg': {'precision': 0.7215909090909092,
    'recall': 0.7321428571428572,
    'f1-score': 0.7246376811594202,
    'support': 19},
    'weighted avg': {'precision': 0.7470095693779905,
    'recall': 0.7368421052631579,
    'f1-score': 0.7398932112890922,
    'support': 19}}})
```

```
[ ]:
```