AdaBoost met boom

December 26, 2020

[1]:
```
import sys
sys.path.append('../')
from ortho_lib3 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
```

1 DataFrame maken

2 experiment maken

[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                        2.436920                       1.006761
1                        2.506438                       1.004394
2                        1.750021                       0.846239
3                        2.129875                       0.592590
4                        2.408378                       0.969648
..                            …                              …
79                       2.420555                       1.532106
80                       1.201943                       0.428302
81                       2.186121                       0.580735
82                       1.882206                       0.808175
83                       2.602110                       1.114584

    angle_right_shoulder_xz_max_AF  angle_right_shoulder_xz_max_RF  \
0                         2.854781                        1.492353
1                         2.430216                        0.942692
```

```
2                          1.693882                          0.926566
3                          2.596719                          0.774593
4                          2.435794                          1.015687
..                              …                                 …
79                         2.390046                          1.290075
80                         2.347702                          0.429440
81                         2.082403                          0.712122
82                         2.064096                          0.987518
83                         2.614965                          1.054871

    angle_left_shoulder_yz_max_AB  angle_right_shoulder_yz_max_AB  \
0                        2.449418                         2.567337
1                        2.363443                         2.545257
2                        2.539127                         2.631044
3                        2.041966                         2.719333
4                        2.185690                         2.483305
..                            …                                …
79                       2.591977                         2.675785
80                       0.897281                         2.104353
81                       2.649783                         2.220724
82                       2.459425                         2.348511
83                       2.574107                         2.385383

    diff_x_wrist_std_EL  diff_x_wrist_std_AF  diff_x_wrist_std_RF  \
0              0.106902             0.186194             0.122632
1              0.124113             0.041740             0.031973
2              0.028811             0.026540             0.075991
3              0.186479             0.158048             0.267008
4              0.235828             0.057160             0.086497
..                  …                    …                    …
79             0.067242             0.053520             0.045416
80             0.131204             0.461456             0.041716
81             0.022393             0.089273             0.157484
82             0.055195             0.085915             0.043581
83             0.037779             0.128872             0.047050

    diff_x_elbow_std_EL  …  angular_acc_xz_elbow_r_mean_AF  \
0              0.071650  …                         0.010789
1              0.057727  …                         0.009480
2              0.025086  …                         0.007872
3              0.069187  …                         0.017188
4              0.054271  …                         0.011178
..                  …  …                               …
79             0.028010  …                         0.021898
80             0.038488  …                         0.030021
81             0.014435  …                         0.036877
82             0.025565  …                         0.026531
```

```
83             0.029085  …                      0.032568

    angular_acc_xz_elbow_r_std_AF  angular_acc_xz_elbow_r_mean_RF  \
0                        0.009927                        0.025711
1                        0.011701                        0.010229
2                        0.010929                        0.007723
3                        0.029981                        0.022430
4                        0.013903                        0.009312
..                            …                               …
79                       0.022046                        0.029374
80                       0.039972                        0.009998
81                       0.036002                        0.044232
82                       0.030848                        0.030147
83                       0.019488                        0.023850

    angular_acc_xz_elbow_r_std_RF  angular_vel_yz_elbow_l_std_AB  \
0                        0.025676                       0.043592
1                        0.011686                       0.046195
2                        0.006885                       0.036618
3                        0.020106                       0.060316
4                        0.008978                       0.041339
..                            …                              …
79                       0.035492                       0.069536
80                       0.007187                       0.013514
81                       0.048294                       0.066856
82                       0.026745                       0.042285
83                       0.021694                       0.057128

    angular_vel_yz_elbow_r_std_AB  angular_acc_yz_elbow_l_mean_AB  \
0                        0.045790                        0.015279
1                        0.053882                        0.012215
2                        0.041675                        0.010405
3                        0.098761                        0.018198
4                        0.048522                        0.009418
..                            …                               …
79                       0.066380                        0.033381
80                       0.040844                        0.009843
81                       0.062815                        0.036638
82                       0.048514                        0.019478
83                       0.066880                        0.029559

    angular_acc_yz_elbow_l_std_AB  angular_acc_yz_elbow_r_mean_AB  \
0                        0.012023                        0.012660
1                        0.010071                        0.012145
2                        0.008925                        0.009402
3                        0.026365                        0.023955
4                        0.008248                        0.010349
```

```
..                                ...                                ...
79                           0.031042                           0.032047
80                           0.008446                           0.017797
81                           0.035517                           0.042551
82                           0.018303                           0.019738
83                           0.022851                           0.030719

    angular_acc_yz_elbow_r_std_AB
0                            0.012067
1                            0.012145
2                            0.008356
3                            0.040463
4                            0.010681
..                                ...
79                           0.036677
80                           0.019753
81                           0.033384
82                           0.018876
83                           0.026552

[84 rows x 78 columns]
```

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

[3]:
```
(84, 78)
```

[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
X[1].shape
```

[5]:
```
(78,)
```

[6]:
```
skf = StratifiedKFold(n_splits=5, random_state=None, shuffle=False)

df_scores = pd.DataFrame()

precision_list = []
recall_list = []
```

```
f1_score_list = []
accuracy_list=[]
i=1

for train_index, test_index in skf.split(X, y):
    Xtrain, Xtest = X[train_index], X[test_index]
    ytrain, ytest = y[train_index], y[test_index]
    regr = AdaBoostRegressor(n_estimators=370, learning_rate = 1.9)
    regr.fit(Xtrain, ytrain)
    ypred = regr.predict(Xtest)
    report = classification_report(ytest, ypred.round(), output_dict=True)
    recall = (report.get('weighted avg').get('recall'))
    precision = (report.get('weighted avg').get('precision'))
    f1_score = (report.get('weighted avg').get('f1-score'))
    accuracy = (report.get('accuracy'))

    precision_list.append(precision)
    recall_list.append(recall)
    f1_score_list.append(f1_score)
    accuracy_list.append(accuracy)
    feature_scores = pd.Series(regr.feature_importances_, index=exp.df.columns)
    df_scores['column' + str(i)] = feature_scores.values
    i=i+1

df_scores['average'] = df_scores.mean(axis=1)
df_scores['feature'] = exp.df.columns
print(f'precision: {np.mean(precision_list)} \nrecall: {np.mean(recall_list)}␣
 ↪\nf1_score: {np.mean(f1_score_list)} \naccuracy: {np.mean(accuracy_list)}')
```

```
precision: 0.5287747220100161
recall: 0.5227941176470587
f1_score: 0.5210126474832356
accuracy: 0.5227941176470587
```

[7]:
```
df_scores[['feature', 'average']].sort_values(by='average', ascending=False).
 ↪head(10).reset_index(drop=True)
```

[7]:

	feature	average
0	diff_z_shoulder_std_AB	0.021088
1	diff_x_wrist_std_EL	0.018780
2	angle_right_shoulder_xz_max_AF	0.017991
3	x_wrist_max_EL	0.017694
4	diff_y_wrist_std_AF	0.016969
5	acc_wrists_x_l_mean_EL	0.016798
6	angle_right_shoulder_yz_max_AB	0.016198
7	z_elbow_max_RF	0.015518
8	diff_x_elbow_std_EL	0.015484

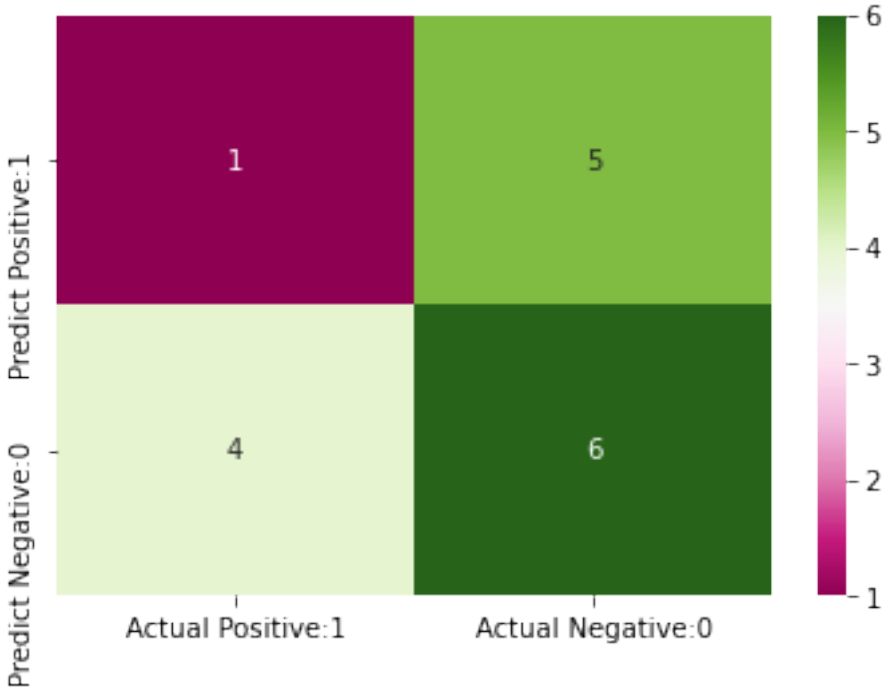
```
9              diff_z_elbow_std_RF  0.015382
```

[8]:
```
from sklearn.metrics import classification_report
report = classification_report(ytest, ypred.round())
print(report)
```

```
              precision    recall  f1-score   support

         0.0       0.20      0.17      0.18         6
         1.0       0.55      0.60      0.57        10

    accuracy                           0.44        16
   macro avg       0.37      0.38      0.38        16
weighted avg       0.42      0.44      0.43        16
```

[9]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(ytest, ypred.round())
cm_matrix = pd.DataFrame(data=cm, columns=['Actual Positive:1', 'Actual␣
↪Negative:0'],
                                 index=['Predict Positive:1', 'Predict Negative:
↪0'])

sns.heatmap(cm_matrix, annot=True, fmt='d', cmap='PiYG')
```

[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc771b4c160>

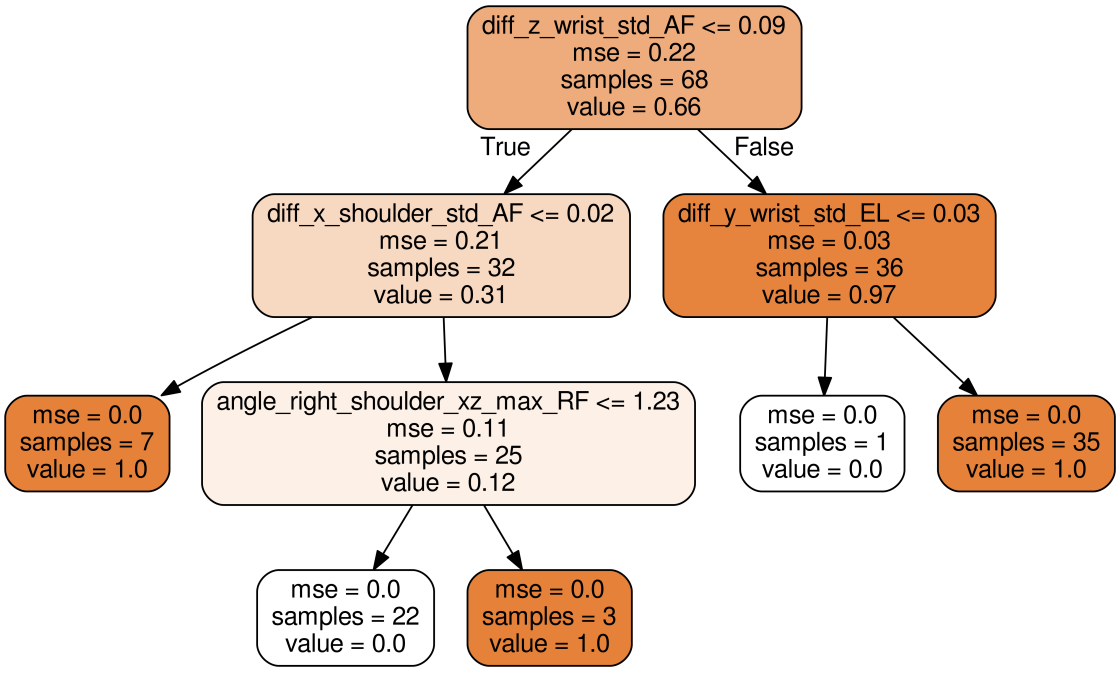


[10]:
```
estimators = regr.estimators_[1]
estimators
```

[10]:
```
DecisionTreeRegressor(max_depth=3, random_state=1314769939)
```

[11]:
```
# Extract single tree
estimator = regr.estimators_[1]

from sklearn.tree import export_graphviz
# Export as dot file
export_graphviz(estimator, out_file='treeADA.dot',
                feature_names = exercises.df.columns,
                class_names = True,
                rounded = True, proportion = False,
                precision = 2, filled = True)

# Convert to png using system command (requires Graphviz)
from subprocess import call
call(['dot', '-Tpng', 'treeADA.dot', '-o', 'treeADA.png', '-Gdpi=600'])

# Display in jupyter notebook
from IPython.display import Image
Image(filename = 'treeADA.png')
```

[11]:



[ ]: