features overzicht

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
```

[2]:
```
def filter_incomplete_patients(exercises, exercises_types = ['AB', 'AF', 'RF',␣
↪'EL']):
    new_exercises = copy.deepcopy(exercises)
    for patient in exercises.patients:
        keys = exercises[patient].keys()
        for ex in exercises_types:
            if ex not in keys:
                new_exercises = new_exercises.drop_patient(patient)
                print('dropped ' + str(patient) + ' because there is (a)␣
↪missing exercise(s): ', list(keys))

    return new_exercises
```

0.1 symmetrie links/rechts

[3]:
```
def symmetrie(df, sl, sr):
    df['verschil'] =  abs(df[sl] - df[sr])
    return sum(df['verschil'])/len(df['verschil'])
```

0.2 afstand (bewegingsrange)

[4]:
```
def afstand(df, sl, sr):
    afstand_links = np.max(df[sl]) - np.min(df[sl])
    afstand_rechts = np.max(df[sr]) - np.min(df[sr])
    return abs(afstand_links, afstand_rechts)
```

0.3 hoogte van de schouder bij maximale ellebooghoogte

[5]:
```
def schouderhoogte(df):
    max_z_4 = np.max(df['z_4'])
    max_z_7 = np.max(df['z_7'])

    max_z_5 = np.max(df['z_5'])
    max_z_8 = np.max(df['z_8'])

    if max_z_4 < max_z_7:
        el =  df.loc[(lambda df: df['z_5'] == max_z_5), 'z_4']
    else:
        el =  df.loc[(lambda df: df['z_8'] == max_z_8), 'z_7']
    return max(el)
```

0.4 maximale hoogte

[6]:
```
def maxhoogte(df, sl, sr):
    return min(np.max(df[sl]), np.max(df[sr]))
```

0.5 maximale hoek schouder

[7]:
```
def hoek(df):
    max_schouder_links = np.max(df['hoek_links'])
    max_schouder_rechts = np.max(df['hoek_rechts'])
    return min(max_schouder_links, max_schouder_rechts)
```

0.6 beweging over de verkeerde as

AB: x-as

AF: y-as

RF: y-as

EL: niks

[8]:
```
def beweging(df, sl, sr):
    min_links = np.min(df[sl])
    max_links = np.max(df[sl])
    verschil_links = max_links-min_links

    min_rechts = np.min(df[sr])
    max_rechts = np.max(df[sr])
    verschil_rechts = max_rechts-min_rechts
```

```
    return max(verschil_links, verschil_rechts)
```

1 DataFrame maken

[9]:
```
class CustomDFFrame(DFFrame):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self['hoek_rechts'] = angle(self, '3', '7', '8')
        self['hoek_links'] = angle(self, '3', '4', '5')
        self['symelleboog'] = symmetrie(self, 'x_5', 'x_8')
        self['sympols'] = symmetrie(self, 'x_6', 'x_9')
        self['symschouder'] = symmetrie(self, 'z_4', 'z_7')
        self['schouderafstand'] = afstand(self, 'z_4', 'z_7')
        self['elleboogafstand'] = afstand(self, 'x_5', 'x_8')
        self['polsafstand'] = afstand(self, 'x_6', 'x_9')
        self['schouderhoogte'] = schouderhoogte(self)
        self['maxelleboog'] = maxhoogte(self, 'z_5', 'z_8')
        self['maxpols'] = maxhoogte(self, 'z_4', 'z_7')
        self['maxhoek'] = hoek(self)
        self['ABxas'] = beweging(self, 'x_5', 'x_8')
        self['AFyas'] = beweging(self, 'y_5', 'y_8')
        self['RFyas'] = beweging(self, 'y_5', 'y_8')
```

[ ]:
```
dff = create_dfframes([1,2,3,4],
                extype= ['AB', 'AF', 'RF', 'EL'],
                dfframe_class = CustomDFFrame,
                data_dir = '../transformed_data',
                print_errors = False,
                show_progress = True)
```

```
VBox(children=(HTML(value=''), IntProgress(value=0, max=1014)))
```

2 features toevoegen

[ ]:
```
Patient.add_features(['maxhoek', 'maxpols',␣
↪'symelleboog','sympols','symschouder','schouderafstand','elleboogafstand','polsafstand',␣
↪'schouderhoogte','maxelleboog'], ['AB', 'AF','EL','RF'], ['min'])
Patient.add_features(['ABxas'], ['AB'], ['min'])
Patient.add_features(['AFyas'], ['AF'], ['min'])
Patient.add_features(['RFyas'], ['RF'], ['min'])
```

3 experiment maken

[ ]:
```
exercises = dffs_to_exercises(dff)
exercises = filter_incomplete_patients(exercises)
exercises.df
```

[ ]:
```
exp = Experiment(exercises, y_condition= lambda cat: cat != 'Category_1')
fss = [
       ['symelleboog_min_AB'], ['sympols_min_AB'],['symschouder_min_AB'],
       ['symelleboog_min_AF'], ['sympols_min_AF'],['symschouder_min_AF'],
       ['symelleboog_min_EL'], ['sympols_min_EL'],['symschouder_min_EL'],
       ['symelleboog_min_RF'], ['sympols_min_RF'],['symschouder_min_RF'],
       ['schouderafstand_min_AB'], ['elleboogafstand_min_AB'],␣
 ↪['polsafstand_min_AB'],
       ['schouderafstand_min_AF'], ['elleboogafstand_min_AF'],␣
 ↪['polsafstand_min_AF'],
       ['schouderafstand_min_EL'], ['elleboogafstand_min_EL'],␣
 ↪['polsafstand_min_EL'],
       ['schouderafstand_min_RF'], ['elleboogafstand_min_RF'],␣
 ↪['polsafstand_min_RF'],
       ['schouderhoogte_min_AB'], ['schouderhoogte_min_AF'],␣
 ↪['schouderhoogte_min_EL'], ['schouderhoogte_min_RF'],
       ['maxelleboog_min_AB'], ['maxelleboog_min_AF'], ['maxelleboog_min_EL'],␣
 ↪['maxelleboog_min_RF'],
       ['maxpols_min_AB'], ['maxpols_min_AF'], ['maxpols_min_EL'],␣
 ↪['maxpols_min_RF'],
       ['maxhoek_min_AB'], ['maxhoek_min_AF'], ['maxhoek_min_EL'],␣
 ↪['maxhoek_min_RF'],
       ['ABxas_min_AB'], ['AFyas_min_AF'], ['RFyas_min_RF']
   ]

results = Results(exp)
exp.cols
```

[ ]:
```
for fs in fss:
    exp.fit_inliers_ensemble(fs, results = results, factor=1.4)
df = results.report()
df
```

[ ]:
```
df = df[df['precision']==1]
df
```

[ ]:
```
goeie = df['features'].tolist()
exp = Experiment(exercises, y_condition= lambda cat: cat != 'Category_1')
fs = goeie
results = Results(exp)
```

```
exp.fit_inliers_ensemble(fs, results = results, factor=1.4)
df = results.report()
df
```

[ ]:
```
print(
    set(results.ids),
    set(exercises.patients),
    set(results.ids) - set(exercises.patients)) # empty set
```

[ ]:
```

```

[ ]:
```

```