AB schouder elleboog hoogte

December 26, 2020

[1]:
```
import sys
sys.path.append("../")
import pandas as pd
from ortho_lib import *
import os
import matplotlib.pyplot as plt
import numpy as np
```

[2]:
```
path_cats = ['..//transformed_data/Category_1/', '..//transformed_data/
 ↪Category_2/', '..//transformed_data/Category_3/', '..//transformed_data/
 ↪Category_4/']
exercise = '/AB1'
df = pd.DataFrame()

def schouderhoogte(path_cat, df = pd.DataFrame()): #bij het aanroepen van de␣
 ↪functie het indexnummer voor de categorie uit path_cats
    global patientID
    patientID = os.listdir(path_cats[path_cat])

    for patient in patientID:
        path = path_cats[path_cat] + patient + exercise + '.txt'
        df_patient = exercise_to_df(path)
        df_patient['patientID'] = patient
        df = df.append([df_patient])
        del df['x']
        del df['y']


    shoulder_df = df[df['sensor'] != '2'] #anker verwijderen uit de dataframe,␣
 ↪dit datapunt is nooit nodig
    shoulder_df = shoulder_df.set_index( ['patientID', 'frame'], drop=True,␣
 ↪inplace=False, verify_integrity=False)
    shoulder_df = shoulder_df[shoulder_df['sensor'] != '3'] #sensoren␣
 ↪verwijderen die niet van belang zijn. Alleen de sensoren bewaren die␣
 ↪vergeleken moeten worden.
    shoulder_df = shoulder_df[shoulder_df['sensor'] != '6']
    shoulder_df = shoulder_df[shoulder_df['sensor'] != '9']
```

```
slechte_arm =[]
schouder_links_list = []
schouder_rechts_list=[]
elleboog_links_list = []
elleboog_rechts_list = []
for patient in patientID:
    dfpatient = df[df['patientID']==str(patient)]

    schouder_links = dfpatient[dfpatient['sensor'] == '4']
    z_schouder_links = list(schouder_links['z'])
    schouder_links_list.append(z_schouder_links)

    schouder_rechts = dfpatient[dfpatient['sensor'] == '7']
    z_schouder_rechts = list(schouder_rechts['z'])
    schouder_rechts_list.append(z_schouder_rechts)

    elleboog_links = dfpatient[dfpatient['sensor'] == '5']
    z_elleboog_links = list(elleboog_links['z'])
    elleboog_links_list.append(z_elleboog_links)

    elleboog_rechts = dfpatient[dfpatient['sensor'] == '8']
    z_elleboog_rechts = list(elleboog_links['z'])
    elleboog_rechts_list.append(z_elleboog_rechts)


    max_s_l = max(schouder_links['z'])
    max_s_r = max(schouder_rechts['z'])

    if max_s_l > max_s_r: #als links hogere waarde heeft dan rechts, dan is␣
↪de linkerarm de slechte arm
        slechte_arm.append('links')
    else:
        slechte_arm.append('rechts')

global shoulder_elbow_df

shoulder_elbow_df = pd.DataFrame()
shoulder_elbow_df['patientID'] = patientID
shoulder_elbow_df.set_index(['patientID'], drop = True, inplace = True)
shoulder_elbow_df['schouder links'] = schouder_links_list
shoulder_elbow_df['schouder rechts'] = schouder_rechts_list
shoulder_elbow_df['elleboog links'] = elleboog_links_list
shoulder_elbow_df['elleboog rechts'] = elleboog_rechts_list
shoulder_elbow_df['slechte arm'] = slechte_arm
shoulder_elbow_df['category'] = path_cat + 1
```

```
    slechte_schouder = []
    slechte_elleboog = []
    for patient in patientID:
        if shoulder_elbow_df.loc[patient]['slechte arm'] == 'rechts':
            slechte_schouder.append(shoulder_elbow_df.loc[patient]['schouder␣
↪rechts'])
            slechte_elleboog.append(shoulder_elbow_df.loc[patient]['elleboog␣
↪rechts'])
        else:
            slechte_schouder.append(shoulder_elbow_df.loc[patient]['schouder␣
↪links'])
            slechte_elleboog.append(shoulder_elbow_df.loc[patient]['elleboog␣
↪links'])

    shoulder_elbow_df['elleboog'] = slechte_elleboog
    shoulder_elbow_df['schouder'] = slechte_schouder

    shoulder_elbow_df = shoulder_elbow_df[['category', 'elleboog', 'schouder']]

    max_elleboog = []
    max_el_schouder = []
    for patient in patientID:
        max_el = np.max(shoulder_elbow_df.loc[patient]['elleboog'])
        index = shoulder_elbow_df.loc[patient]['elleboog'].index(max_el)
        el_schouder = shoulder_elbow_df.loc[patient]['schouder'][index]

        max_elleboog.append(max_el)
        max_el_schouder.append(el_schouder)

    max_el_schouder_df = pd.DataFrame()
    max_el_schouder_df['category'] = shoulder_elbow_df['category']
    max_el_schouder_df['max elleboghoogte'] = max_elleboog
    max_el_schouder_df['schouderhoogte'] = max_el_schouder

    #return shoulder_elbow_df
    return max_el_schouder_df
```

[3]:
```
schouderhoogte_df =pd.concat([schouderhoogte(0), schouderhoogte(1),␣
↪schouderhoogte(2), schouderhoogte(3)])
```

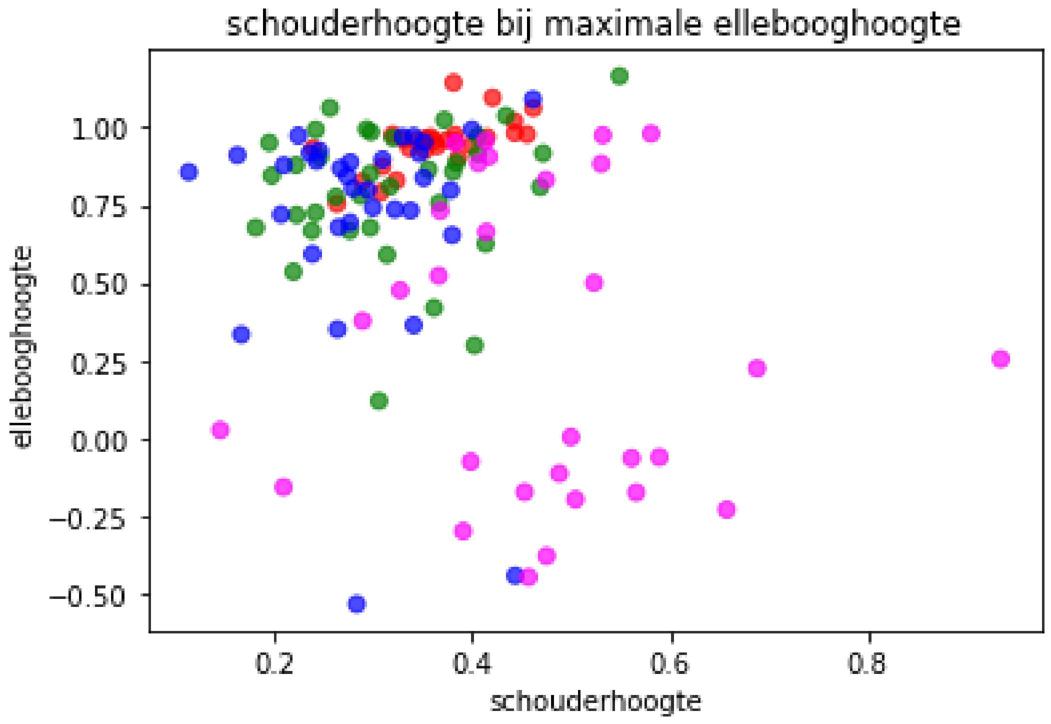
[4]:
```
schouderhoogte_df
```

[4]:
```
           category  max elleboghoogte  schouderhoogte
patientID
8                 1           0.874375        0.309388
3                 1           0.791086        0.307439
```

```
1                 1         0.968847        0.358060
22                1         0.902799        0.386102
17                1         0.955169        0.383486
…             …             …           …
27                4        -0.112915        0.488138
5                 4        -0.228941        0.657194
2                 4        -0.060507        0.589145
4                 4         0.224927        0.687858
24                4         0.884682        0.406214

[127 rows x 3 columns]
```

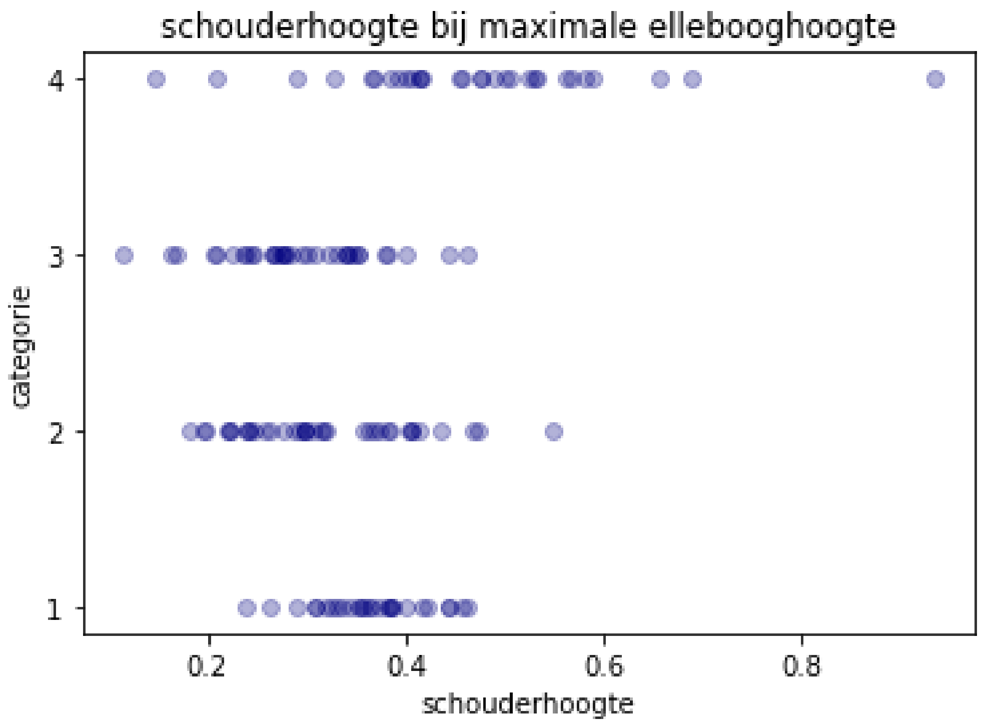
[5]:
```
colors = {1:'red', 2:'green', 3:'blue', 4:'magenta'}

plt.scatter(schouderhoogte_df['schouderhoogte'], schouderhoogte_df['max␣
 ↪elleboghoogte'], alpha = 0.7, c=schouderhoogte_df['category'].map(colors))
plt.title('schouderhoogte bij maximale elleboghoogte')
# plt.yticks([1,2,3,4])
plt.xlabel('schouderhoogte')
plt.ylabel('elleboghoogte')
```

[5]:
```
Text(0, 0.5, 'elleboghoogte')
```



```
[6]: plt.scatter(schouderhoogte_df['schouderhoogte'], schouderhoogte_df['category'],␣
      ↪alpha=0.3, c='navy')
     plt.title('schouderhoogte bij maximale elleboghoogte')
     plt.yticks([1,2,3,4])
     plt.xlabel('schouderhoogte')
     plt.ylabel('categorie')
```

```
[6]: Text(0, 0.5, 'categorie')
```



```
[7]: schouderhoogte_df_subset = schouderhoogte_df[(schouderhoogte_df['category'] ==␣
      ↪1)|(schouderhoogte_df['category'] == 4)]
     schouderhoogte_df_subset
```

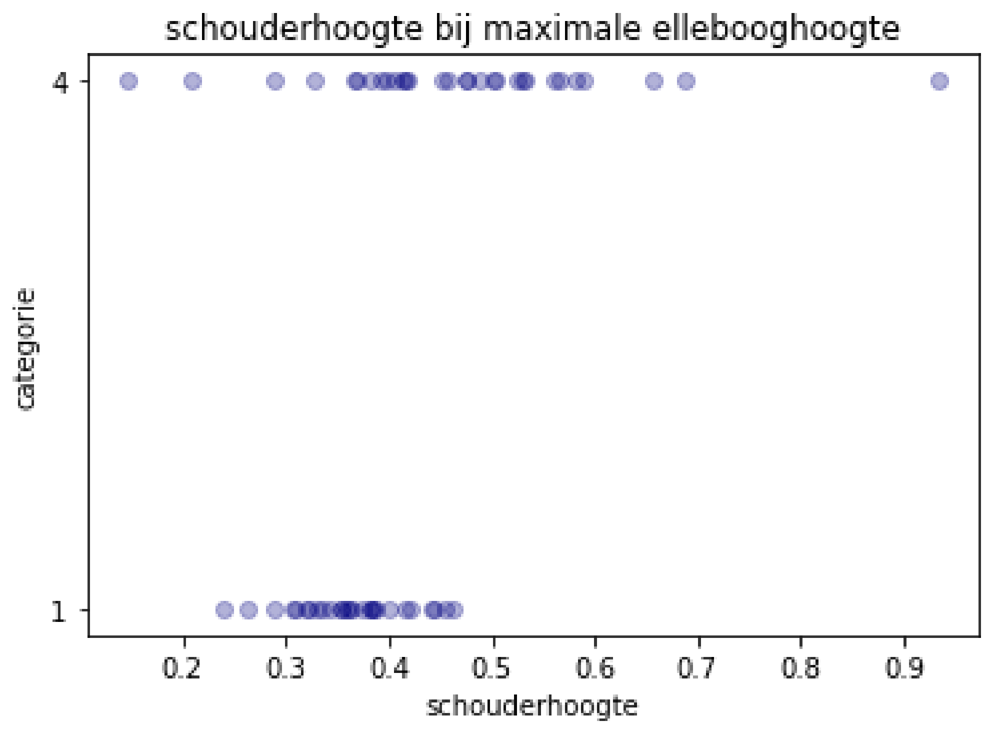
```
[7]:            category  max elleboghoogte  schouderhoogte
     patientID
     8                 1           0.874375        0.309388
     3                 1           0.791086        0.307439
     1                 1           0.968847        0.358060
     22                1           0.902799        0.386102
     17                1           0.955169        0.383486
     15                1           0.939635        0.364358
     6                 1           0.964969        0.353481
     11                1           0.957720        0.343624
```

13	1	1.095390	0.420626
7	1	0.976534	0.319702
29	1	0.977238	0.455504
23	1	0.968325	0.415028
21	1	0.956031	0.362277
9	1	0.960043	0.331118
20	1	0.932946	0.239097
19	1	0.981771	0.443059
26	1	1.143378	0.381151
12	1	0.753785	0.263476
10	1	1.061813	0.461762
30	1	0.825390	0.288977
16	1	0.975518	0.382626
27	1	0.829090	0.323521
5	1	0.932510	0.336521
2	1	1.019917	0.442558
4	1	0.926292	0.352480
28	1	0.958440	0.374242
24	1	0.941978	0.399557
35	4	-0.156847	0.209408
8	4	-0.378071	0.475214
3	4	0.974471	0.531953
1	4	0.255151	0.933827
36	4	0.377861	0.289206
14	4	-0.173864	0.565777
34	4	0.731316	0.367777
22	4	0.882636	0.530356
33	4	-0.063890	0.561118
38	4	0.949839	0.383062
31	4	0.905426	0.416932
6	4	-0.074987	0.398668
41	4	-0.196054	0.504358
11	4	0.003941	0.499731
7	4	-0.446249	0.457002
29	4	0.663402	0.414295
23	4	0.499985	0.523077
21	4	0.830854	0.474490
9	4	0.979532	0.580697
25	4	-0.172479	0.452984
26	4	-0.298293	0.390977
39	4	0.522693	0.366249
12	4	0.956342	0.413487
10	4	0.476003	0.327305
30	4	0.026412	0.145721
27	4	-0.112915	0.488138
5	4	-0.228941	0.657194
2	4	-0.060507	0.589145

```
4                  4          0.224927        0.687858
24                 4          0.884682        0.406214
```

[8]:
```
plt.scatter(schouderhoogte_df_subset['schouderhoogte'],␣
 ↪schouderhoogte_df_subset['category'], alpha=0.3, c='navy')
plt.title('schouderhoogte bij maximale elleboghoogte')
plt.yticks([1,4])
plt.xlabel('schouderhoogte')
plt.ylabel('categorie')
```

[8]:
```
Text(0, 0.5, 'categorie')
```



[9]:
```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
import numpy as np
from sklearn.linear_model import LogisticRegression

#splitten test en train set
X = np.asarray(schouderhoogte_df_subset[['schouderhoogte']])
y = np.asarray(schouderhoogte_df_subset[['category']])

scores=[]
```

```python
skf = StratifiedKFold(n_splits=6)
for train, test in skf.split(X, y):
    X_train, X_test = X[train], X[test]
    y_train, y_test = y[train], y[test]
    logistic_reg = LogisticRegression()
    logistic_reg.fit(X_train,y_train)
    y_predict = logistic_reg.predict(X_test)
    score = logistic_reg.score(X_test, y_test)
    print(y_predict, score)
    scores.append(score)

print(np.mean(scores))
```

```
[4 4 4 4 4 1 4 4 4 1] 0.3
[4 4 4 4 4 4 4 4 4 4] 0.5
[4 4 4 4 1 4 4 4 4 4] 0.6
[4 4 1 4 4 4 4 4 4] 0.6666666666666666
[1 4 1 1 4 1 4 1 1] 0.5555555555555556
[4 4 4 4 4 4 4 4 4] 0.5555555555555556
0.5296296296296296
```

```
/opt/jupyterhub/anaconda/lib/python3.6/site-
packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  return f(**kwargs)
/opt/jupyterhub/anaconda/lib/python3.6/site-
packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  return f(**kwargs)
/opt/jupyterhub/anaconda/lib/python3.6/site-
packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  return f(**kwargs)
/opt/jupyterhub/anaconda/lib/python3.6/site-
packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  return f(**kwargs)
/opt/jupyterhub/anaconda/lib/python3.6/site-
packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  return f(**kwargs)
/opt/jupyterhub/anaconda/lib/python3.6/site-
packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vector
```

```
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  return f(**kwargs)
```

[10]:
```
# schouderhoogte(0)
# for patient in patientID:
#     if patient == '8':
#         plt.plot(shoulder_elbow_df.loc[patient]['elleboog'],␣
↪shoulder_elbow_df.loc[patient]['schouder'], alpha = 0.3, c = 'magenta')
#         plt.ylabel('hoogte van de elleboog')
#         plt.xlabel('hoogte van de schouder')
```

[ ]:
```

```