

# EL polsafstand 4 onderscheiden

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 = '/EL1'
df = pd.DataFrame()

def polsafstand(path_cat, df = pd.DataFrame()): #bij het aanroepen van de
↳functie het indexnummer voor de categorie uit path_cats
    patientID = os.listdir(path_cats[path_cat])
    if path_cat == 3:
        patientID.remove('23')
        patientID.remove('21')

    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['z']
        del df['y']

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

```

wrist_df = wrist_df[wrist_df['sensor'] != '4']
wrist_df = wrist_df[wrist_df['sensor'] != '5']
wrist_df = wrist_df[wrist_df['sensor'] != '7']
wrist_df = wrist_df[wrist_df['sensor'] != '8']

minpolsafstand_list = []
for patient in patientID:
    dfpatient = df[df['patientID']==str(patient)]
    per_patient_6 = dfpatient[dfpatient['sensor'] == '6']
    per_patient_9 = dfpatient[dfpatient['sensor'] == '9']
    max_6 = max(per_patient_6['x'])
    min_6 = min(per_patient_6['x'])
    verschil_6 = max_6 - min_6
    max_9 = max(per_patient_9['x'])
    min_9 = min(per_patient_9['x'])
    verschil_9 = max_9 - min_9
    minpolsafstand = min(verschil_6, verschil_9)
    minpolsafstand_list.append(minpolsafstand)

wrist_distance_df = pd.DataFrame()
wrist_distance_df['patientID'] = patientID
wrist_distance_df.set_index(['patientID'], drop = True, inplace = True)
wrist_distance_df['wrist distance'] = minpolsafstand_list
wrist_distance_df['category'] = path_cat + 1

return wrist_distance_df

```

```

[3]: df_polsen = pd.concat([polsafstand(0), polsafstand(1), polsafstand(2),
↪polsafstand(3)])

```

```

[4]: df_polsen
df_wel_4 = df_polsen[df_polsen['category']==4]
df_niet_4 = df_polsen[df_polsen['category']!=4]
df_niet_4['category'] = 0
df_polsen = pd.concat([df_wel_4, df_niet_4])
df_polsen

```

/opt/jupyterhub/anaconda/lib/python3.6/site-packages/ipykernel\_launcher.py:4:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
after removing the cwd from sys.path.

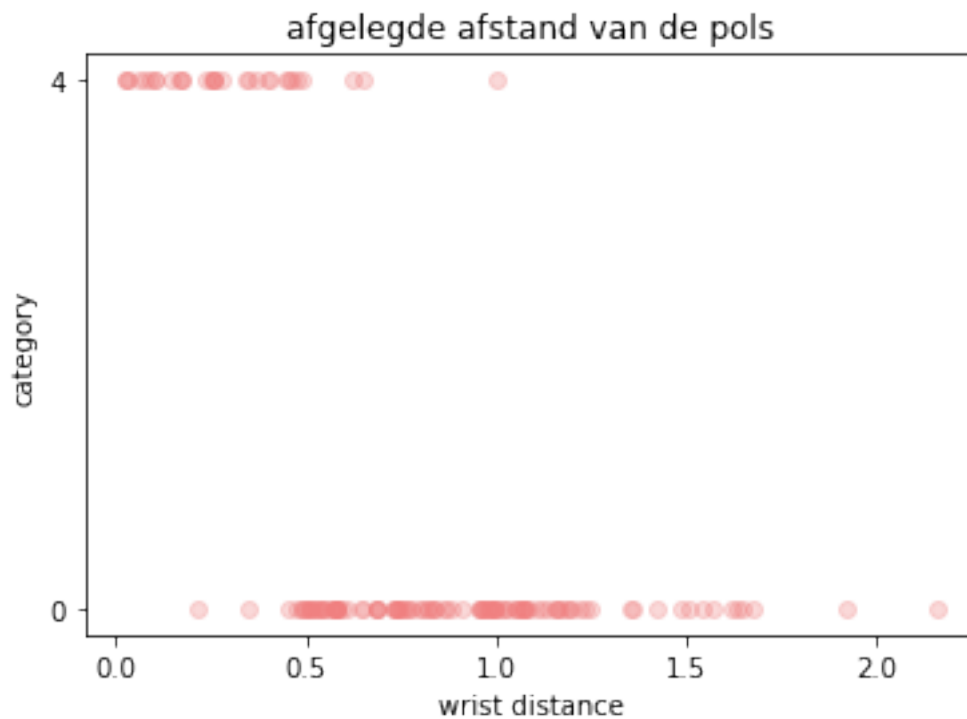
```
[4]:
```

|           | wrist distance | category |
|-----------|----------------|----------|
| patientID |                |          |
| 35        | 0.071632       | 4        |
| 8         | 0.086545       | 4        |
| 3         | 0.274643       | 4        |
| 1         | 0.647750       | 4        |
| 36        | 0.394949       | 4        |
| ...       | ...            | ...      |
| 5         | 0.534417       | 0        |
| 18        | 0.766089       | 0        |
| 4         | 0.476574       | 0        |
| 28        | 0.738165       | 0        |
| 24        | 0.486092       | 0        |

[137 rows x 2 columns]

```
[11]: plt.scatter(df_polsen['wrist distance'], df_polsen['category'], alpha = 0.3,
    ↪color = 'lightcoral')
plt.title('afgelegde afstand van de pols')
plt.yticks([0,4])
plt.xlabel('wrist distance')
plt.ylabel('category')
```

```
[11]: Text(0, 0.5, 'category')
```



```
[12]: 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(df_polsen[['wrist distance']])
y = np.asarray(df_polsen[['category']])

scores=[]

skf = StratifiedKFold(n_splits=10)
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 0 0 0 0 0 0 0 0 0 0 0] 0.9285714285714286
[0 4 4 0 0 0 0 0 0 0 0 0 0 0] 0.9285714285714286
[4 4 4 0 0 0 0 0 0 0 0 0 0 0] 1.0
[4 4 4 0 0 0 0 0 0 0 0 0 0 0] 1.0
[4 4 4 0 0 4 0 0 0 0 0 0 0 0] 0.9285714285714286
[0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.7857142857142857
[0 4 4 0 0 0 0 0 0 0 0 0 0 0] 0.9285714285714286
[0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.7692307692307693
[0 0 0 0 0 0 0 0 0 0 0 0 0 0] 0.7692307692307693
[0 4 4 0 0 0 0 0 0 0 0 0 0 0] 0.9230769230769231
0.8961538461538463
```

```
/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)
/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)

```

[ ]: