EL polsafstand

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
```

[4]:

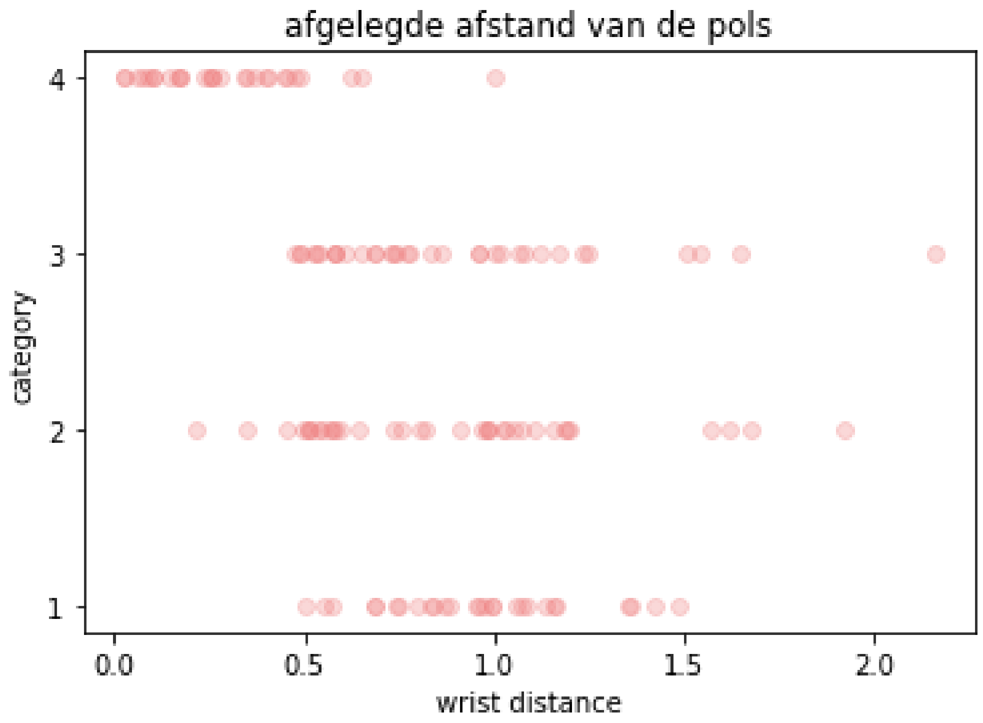
```
           wrist distance  category
patientID
8                0.840174         1
3                1.349328         1
1                1.054837         1
22               0.829084         1
17               0.992387         1
…                   …        …
27               1.003091         4
5                0.487523         4
2                0.369859         4
4                0.445833         4
24               0.257944         4

[125 rows x 2 columns]
```

[5]:

```
plt.scatter(df_polsen['wrist distance'], df_polsen['category'], alpha = 0.3,␣
 ↪color ='lightcoral')
plt.title('afgelegde afstand van de pols')
plt.yticks([1,2,3,4])
plt.xlabel('wrist distance')
plt.ylabel('category')
```

[5]:

```
Text(0, 0.5, 'category')
```



[6]:

```
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=6)
for train, test in skf.split(X, y):
    X_train, X_test = X[train], X[test]
```

```python
    y_train, y_test = y[train], y[test]
    logistic_reg = LogisticRegression(multi_class='multinomial', solver='saga') #multinomial of ovr???
    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))
```

```
[2 2 2 2 2 4 2 2 2 4 2 4 2 4 2 2 2 4 4 4 2] 0.3333333333333333
[2 2 2 2 2 4 1 2 2 2 4 2 2 2 3 2 4 4 4 4 4] 0.42857142857142855
[2 2 2 2 2 2 3 4 2 2 2 2 2 2 2 3 4 4 4 4 4] 0.47619047619047616
[2 2 2 2 2 2 2 2 4 2 2 2 4 2 2 2 4 4 4 4 4] 0.47619047619047616
[2 2 4 2 2 2 4 2 2 2 2 2 2 2 2 2 4 4 4 2 2] 0.38095238095238093
[3 2 3 3 3 3 3 3 4 3 3 2 2 4 2 4 4 4 4 4] 0.25
0.39087301587301587
```

```
/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().
```

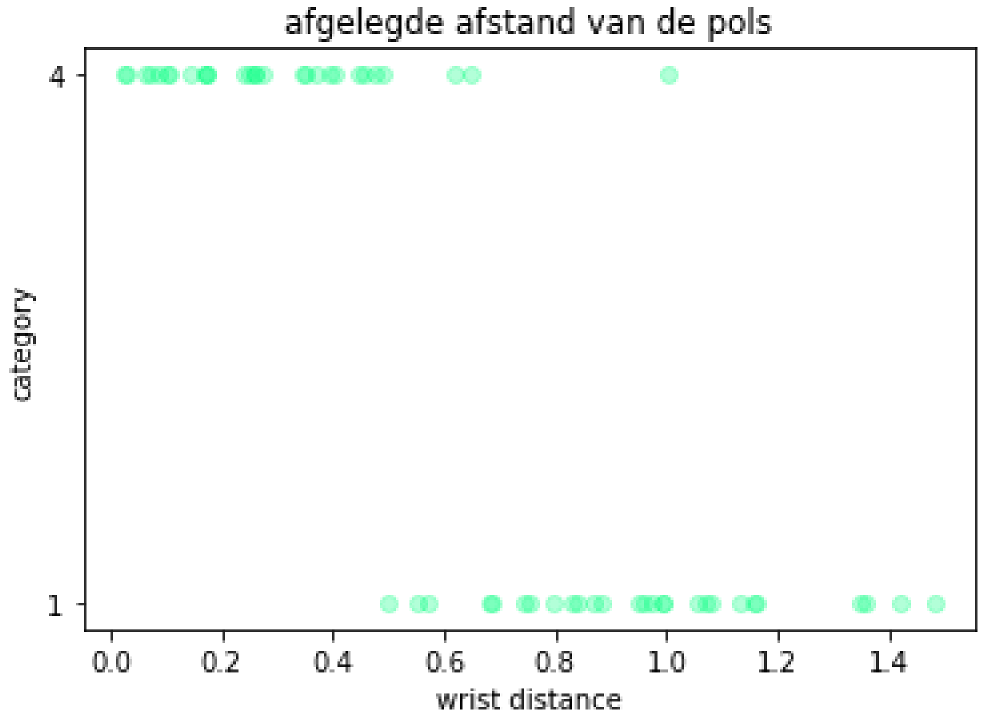
```
 return f(**kwargs)
```

[7]:

```
df_polsen_subset = df_polsen[(df_polsen['category']==␣
 ↪1)|(df_polsen['category']== 4)]
plt.scatter(df_polsen_subset['wrist distance'], df_polsen_subset['category'],␣
 ↪alpha = 0.3, color = 'springgreen')
plt.title('afgelegde afstand van de pols')
plt.yticks([1,4])
plt.xlabel('wrist distance')
plt.ylabel('category')
```

[7]:

```
Text(0, 0.5, 'category')
```



[8]:

```
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_subset[['wrist distance']])
y = np.asarray(df_polsen_subset[['category']])

scores=[]
```

```
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))
```

```
[1 1 1 1 1 4 4 4 1 4] 0.9
[1 1 1 1 1 4 4 4 4] 1.0
[1 1 1 1 1 4 4 4 4] 1.0
[1 1 1 1 4 4 4 4 4] 1.0
[4 1 4 1 4 4 4 4 4] 0.7777777777777778
[1 4 1 1 1 4 4 4 4] 0.7777777777777778
0.9092592592592593
```

```
/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)
```

[ ]:

```
```