EL schouderhoogte

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 schouderhoogte(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['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'] != '5']
    shoulder_df = shoulder_df[shoulder_df['sensor'] != '9']
    shoulder_df = shoulder_df[shoulder_df['sensor'] != '8']

    minschouderafstand_list = []
    for patient in patientID:
        dfpatient = df[df['patientID']==str(patient)]
        per_patient_4 = dfpatient[dfpatient['sensor'] == '4']
        per_patient_7 = dfpatient[dfpatient['sensor'] == '7']
        max_4 = max(per_patient_4['z'])
        min_4 = min(per_patient_4['z'])
        verschil_4 = max_4 - min_4
        max_7 = max(per_patient_7['z'])
        min_7 = min(per_patient_7['z'])
        verschil_7 = max_7 - min_7
        minschouderafstand = max(verschil_4, verschil_7)
        minschouderafstand_list.append(minschouderafstand)

    shoulder_distance_df = pd.DataFrame()
    shoulder_distance_df['patientID'] = patientID
    shoulder_distance_df.set_index(['patientID'], drop = True, inplace = True)
    shoulder_distance_df['shoulder distance'] = minschouderafstand_list
    shoulder_distance_df['category'] = path_cat + 1

    return shoulder_distance_df
```

[3]:

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

[4]:

```
df_schouders
```

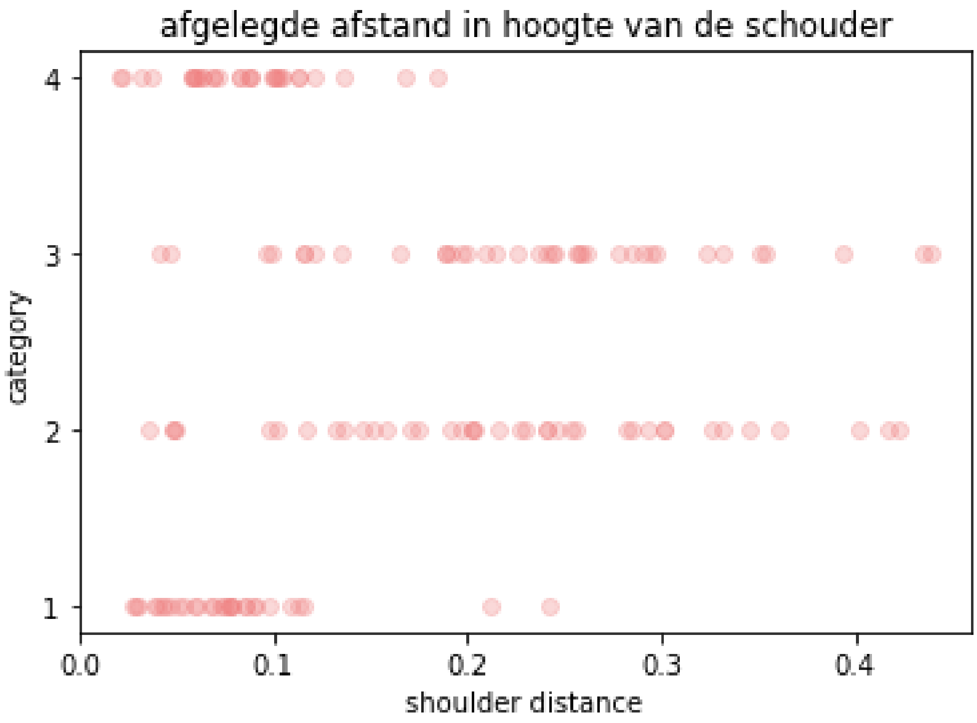
[4]:

```
           shoulder distance  category
patientID
8                   0.027868         1
3                   0.212365         1
1                   0.074604         1
14                  0.042877         1
22                  0.029527         1
…                        …         …
5                   0.102421         4
2                   0.087518         4
4                   0.099942         4
28                  0.104417         4
24                  0.082356         4

[137 rows x 2 columns]
```

[5]:
```
plt.scatter(df_schouders['shoulder distance'], df_schouders['category'], alpha = 0.3, color ='lightcoral')
plt.title('afgelegde afstand in hoogte van de schouder')
plt.yticks([1,2,3,4])
plt.xlabel('shoulder 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_schouders[['shoulder distance']])
y = np.asarray(df_schouders[['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(multi_class='multinomial', solver='saga')
    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 2 4 4 4 4 2 2 2 4 2 2 2 2 2 2 2 2 4 4 4 2 2] 0.34782608695652173
[4 4 4 2 4 3 2 2 3 4 3 3 2 2 2 3 2 2 4 4 4 4 4] 0.34782608695652173
[4 2 4 4 2 2 2 3 2 2 2 3 2 2 3 2 2 2 2 2 2 4 2] 0.30434782608695654
[4 2 2 4 4 2 2 2 2 2 2 4 2 2 2 2 2 2 2 4 4 2 4] 0.391304347826087
[2 2 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2] 0.30434782608695654
[4 2 2 4 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2] 0.2727272727272727
0.32806324110671936
```

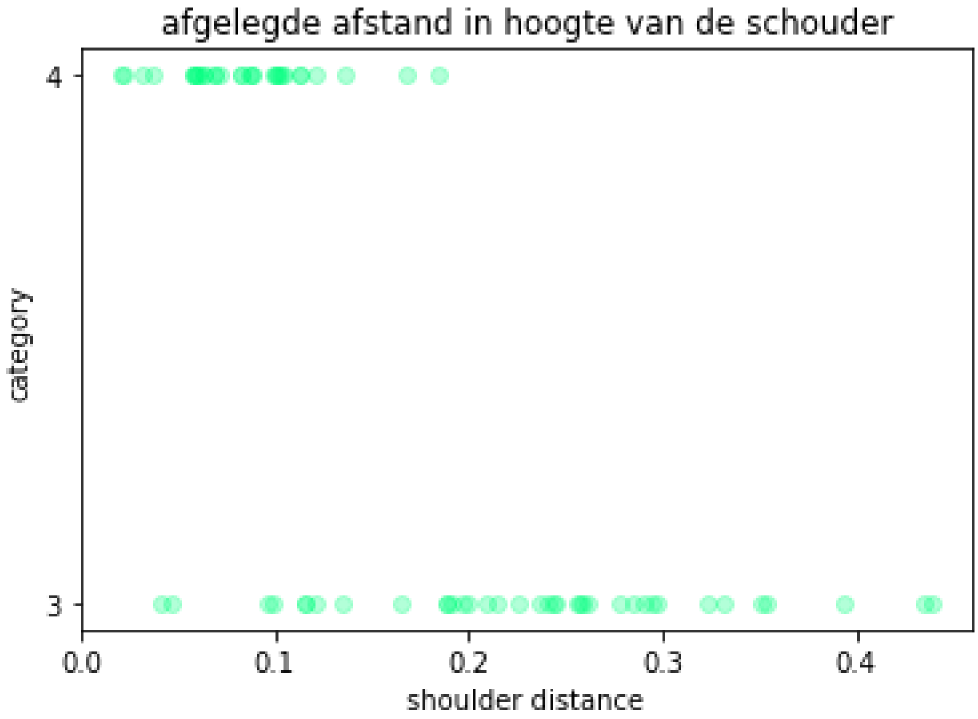
```
/opt/jupyterhub/anaconda/lib/python3.6/site-
packages/sklearn/utils/validation.py:72: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  return f(**kwargs)
/opt/jupyterhub/anaconda/lib/python3.6/site-
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_schouder_subset = df_schouders[(df_schouders['category']==␣
↪3)|(df_schouders['category']== 4)]
plt.scatter(df_schouder_subset['shoulder distance'],␣
↪df_schouder_subset['category'], alpha = 0.3, color = 'springgreen')
plt.title('afgelegde afstand in hoogte van de schouder')
plt.yticks([3,4])
plt.xlabel('shoulder 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_schouder_subset[['shoulder distance']])
y = np.asarray(df_schouder_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))
```

```
[3 3 3 3 3 3 3 4 4 4 3 3] 0.8333333333333334
[3 3 3 3 3 3 3 4 4 4 3 3] 0.75
[3 3 3 3 3 4 3 3 4 3 3] 0.5454545454545454
[3 3 3 3 3 3 4 3 3 3 4] 0.7272727272727273
[3 3 3 3 3 3 4 3 3 3 3] 0.6363636363636364
[3 4 3 3 3 3 3 3 3 3 3] 0.45454545454545453
0.6578282828282829
```

```
/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)
```

[ ]:

```
```