

In [141]...

```
# Imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
import seaborn as sns #data visualization library
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import normalize
import glob
```

In [142]...

```
# Helper functions:
def print_confusion_matrix(y, y_pred):
    cm = confusion_matrix(y, y_pred)
    sns.heatmap(cm, annot=True)
```

In [143]...

```
# Only split data:
def split_data(samples, amount):

    augmented_dataset = []
    for sample in samples:
        new_a1 = np.array_split(sample[0][:80000], amount)
        new_a2 = np.array_split(sample[1][:80000], amount)
        new_a3 = np.array_split(sample[2][:80000], amount)
        new_a4 = np.array_split(sample[3][:80000], amount)
        for i in range(0, amount):
            datapoint = np.hstack([new_a1[i], new_a2[i],
                                   new_a3[i], new_a4[i]])

            datapoint = np.delete(datapoint, 1)
            augmented_dataset.append(datapoint)
    return augmented_dataset
```

In [144]...

```
# Split and preprocess:
def split_and_process_data(samples, amount):
    augmented_dataset = []
    for sample in samples:

        new_a1 = np.array_split(sample[0][:80000], amount)
        new_a2 = np.array_split(sample[1][:80000], amount)
        new_a3 = np.array_split(sample[2][:80000], amount)
        new_a4 = np.array_split(sample[3][:80000], amount)

        for i in range(0, amount):
            a1 = abs(np.fft.fft(new_a1[i]).real)
            a2 = abs(np.fft.fft(new_a2[i]).real)
            a3 = abs(np.fft.fft(new_a3[i]).real)
            a4 = abs(np.fft.fft(new_a4[i]).real)

            datapoint = np.hstack([a1, a2, a3, a4])
            datapoint = np.delete(datapoint, 1)
            augmented_dataset.append(datapoint)
    return augmented_dataset
```

In [145]...

```
# Get data:
```

```

Raw_Broken = []
path = "./archive/BrokenTooth"
csv_files = glob.glob(path + "/*.csv", recursive = True)
for file in csv_files:
    df = pd.read_csv(file)
    a1 = np.array(df['a1'])
    a2 = np.array(df['a2'])
    a3 = np.array(df['a3'])
    a4 = np.array(df['a4'])
    datapoint = [a1,a2,a3,a4]
    Raw_Broken.append(datapoint)
Raw_Healthy = []
path = "./archive/Healthy"
csv_files = glob.glob(path + "/*.csv", recursive = True)
for file in csv_files:
    df = pd.read_csv(file)
    a1 = np.array(df['a1'])
    a2 = np.array(df['a2'])
    a3 = np.array(df['a3'])
    a4 = np.array(df['a4'])
    datapoint = [a1,a2,a3,a4]
    Raw_Healthy.append(datapoint)

```

In [146...

```

# Generate samples:
new_broken = split_and_process_data(Raw_Broken,20)
new_healthy = split_and_process_data(Raw_Healthy,20)

samples = new_broken + new_healthy

labels1 = [np.array(0)] * 200
labels2 = [np.array(1)] * 200
labels = labels1 + labels2
#print(labels)
X_train,X_test, y_train, y_test = train_test_split(samples,
                                                    labels,test_size=0.3,random_state = random.randint(0,4000))
print("Train data:{}".format(len(X_train)))
print("Test data:{}".format(len(X_test)))
#print("Validation data:{}".format(len(X_val)))

```

Train data:280

Test data:120

In [147...

```

# Make MLP model:
mlp = MLPClassifier(early_stopping = True, verbose = True,
                    learning_rate_init = 0.01,
                    activation = 'tanh',validation_fraction = 0.1)
scores = cross_val_score(mlp,X_train,y_train,cv=5)
mlp = mlp.fit(X_train,y_train)
print("%0.2f accuracy with a standard deviation of %0.2f" %
      (scores.mean(), scores.std()))
y_pred = mlp.predict(X_test)
print("accuracy score: {}".format(accuracy_score(y_test,y_pred)))

print_confusion_matrix(y_test,y_pred)

```

Iteration 1, loss = 0.97394995

Validation score: 0.478261

Iteration 2, loss = 1.74493874

Validation score: 0.478261

Iteration 3, loss = 1.36161366

Validation score: 0.478261

Iteration 4, loss = 0.78857342

```
Validation score: 1.000000
Iteration 5, loss = 0.53346500
Validation score: 0.826087
Iteration 6, loss = 0.51462885
Validation score: 0.826087
Iteration 7, loss = 0.51773095
Validation score: 0.826087
Iteration 8, loss = 0.48790812
Validation score: 1.000000
Iteration 9, loss = 0.44958388
Validation score: 1.000000
Iteration 10, loss = 0.44659934
Validation score: 0.565217
Iteration 11, loss = 0.46395957
Validation score: 0.782609
Iteration 12, loss = 0.45913916
Validation score: 0.478261
Iteration 13, loss = 0.50526329
Validation score: 1.000000
Iteration 14, loss = 0.43515594
Validation score: 1.000000
Iteration 15, loss = 0.40337683
Validation score: 1.000000
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 1.04852286
Validation score: 0.478261
Iteration 2, loss = 0.94220735
Validation score: 0.478261
Iteration 3, loss = 0.67543862
Validation score: 0.521739
Iteration 4, loss = 0.53871552
Validation score: 0.521739
Iteration 5, loss = 0.54072904
Validation score: 0.521739
Iteration 6, loss = 0.65376638
Validation score: 0.521739
Iteration 7, loss = 0.69865064
Validation score: 0.521739
Iteration 8, loss = 0.66203001
Validation score: 0.521739
Iteration 9, loss = 0.52265199
Validation score: 1.000000
Iteration 10, loss = 0.47042680
Validation score: 0.478261
Iteration 11, loss = 0.55415527
Validation score: 0.478261
Iteration 12, loss = 0.76118524
Validation score: 0.478261
Iteration 13, loss = 0.95453778
Validation score: 0.478261
Iteration 14, loss = 0.85084123
Validation score: 0.478261
Iteration 15, loss = 0.72532165
Validation score: 0.478261
Iteration 16, loss = 0.55337057
Validation score: 1.000000
Iteration 17, loss = 0.43061551
Validation score: 1.000000
Iteration 18, loss = 0.40023386
Validation score: 1.000000
Iteration 19, loss = 0.40047670
Validation score: 1.000000
Iteration 20, loss = 0.39487293
```

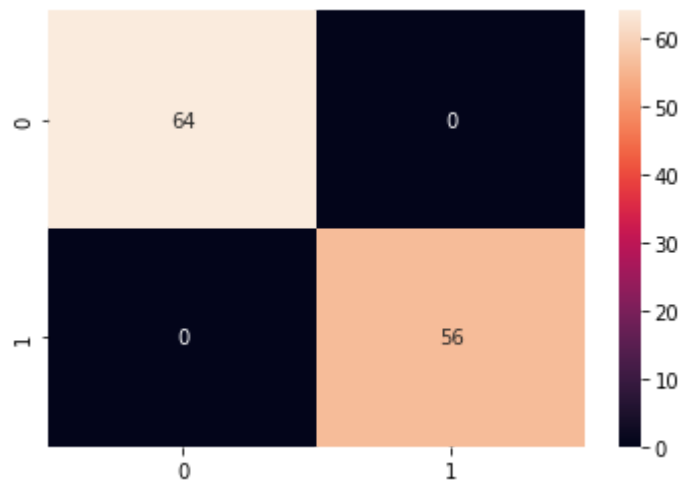
```
Validation score: 1.000000
Validation score did not improve more than tol=0.000100 for 10 consecutive epo
chs. Stopping.
Iteration 1, loss = 1.12718595
Validation score: 0.782609
Iteration 2, loss = 0.61014318
Validation score: 0.478261
Iteration 3, loss = 0.62527822
Validation score: 0.478261
Iteration 4, loss = 0.60372542
Validation score: 0.478261
Iteration 5, loss = 0.73857022
Validation score: 0.478261
Iteration 6, loss = 0.64468637
Validation score: 0.739130
Iteration 7, loss = 0.46726355
Validation score: 1.000000
Iteration 8, loss = 0.42236850
Validation score: 0.913043
Iteration 9, loss = 0.46160517
Validation score: 0.608696
Iteration 10, loss = 0.44888599
Validation score: 1.000000
Iteration 11, loss = 0.40165106
Validation score: 1.000000
Iteration 12, loss = 0.36581394
Validation score: 0.869565
Iteration 13, loss = 0.40216049
Validation score: 0.478261
Iteration 14, loss = 0.51946677
Validation score: 0.869565
Iteration 15, loss = 0.35818022
Validation score: 0.869565
Iteration 16, loss = 0.35576159
Validation score: 1.000000
Iteration 17, loss = 0.29873953
Validation score: 1.000000
Iteration 18, loss = 0.27855491
Validation score: 1.000000
Validation score did not improve more than tol=0.000100 for 10 consecutive epo
chs. Stopping.
Iteration 1, loss = 1.04657871
Validation score: 0.478261
Iteration 2, loss = 1.09227111
Validation score: 0.478261
Iteration 3, loss = 0.86077112
Validation score: 0.478261
Iteration 4, loss = 0.64491334
Validation score: 0.521739
Iteration 5, loss = 0.60897639
Validation score: 0.521739
Iteration 6, loss = 0.72546944
Validation score: 0.521739
Iteration 7, loss = 0.77760005
Validation score: 0.521739
Iteration 8, loss = 0.68206159
Validation score: 0.521739
Iteration 9, loss = 0.60959955
Validation score: 0.521739
Iteration 10, loss = 0.59670278
Validation score: 0.521739
Iteration 11, loss = 0.51499524
Validation score: 1.000000
Iteration 12, loss = 0.44425013
```

```
Validation score: 1.000000
Iteration 13, loss = 0.37415408
Validation score: 1.000000
Iteration 14, loss = 0.41115274
Validation score: 1.000000
Iteration 15, loss = 0.39422410
Validation score: 1.000000
Iteration 16, loss = 0.34550743
Validation score: 1.000000
Iteration 17, loss = 0.28067943
Validation score: 1.000000
Iteration 18, loss = 0.30579344
Validation score: 1.000000
Iteration 19, loss = 0.27295385
Validation score: 1.000000
Iteration 20, loss = 0.28222642
Validation score: 1.000000
Iteration 21, loss = 0.26494401
Validation score: 1.000000
Iteration 22, loss = 0.27320721
Validation score: 1.000000
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 0.80318568
Validation score: 0.565217
Iteration 2, loss = 0.51964660
Validation score: 0.782609
Iteration 3, loss = 0.49085785
Validation score: 0.478261
Iteration 4, loss = 0.69904667
Validation score: 0.478261
Iteration 5, loss = 0.73507186
Validation score: 0.478261
Iteration 6, loss = 0.66769486
Validation score: 1.000000
Iteration 7, loss = 0.41923656
Validation score: 0.956522
Iteration 8, loss = 0.46058042
Validation score: 0.521739
Iteration 9, loss = 0.58123680
Validation score: 0.521739
Iteration 10, loss = 0.78426297
Validation score: 0.521739
Iteration 11, loss = 0.69334983
Validation score: 0.521739
Iteration 12, loss = 0.53935866
Validation score: 0.521739
Iteration 13, loss = 0.53712286
Validation score: 0.913043
Iteration 14, loss = 0.47776565
Validation score: 1.000000
Iteration 15, loss = 0.39407173
Validation score: 0.956522
Iteration 16, loss = 0.45262922
Validation score: 0.478261
Iteration 17, loss = 0.51835128
Validation score: 0.478261
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 0.97297019
Validation score: 0.500000
Iteration 2, loss = 0.45039439
Validation score: 1.000000
Iteration 3, loss = 0.35052394
```

```

Validation score: 1.000000
Iteration 4, loss = 0.33714669
Validation score: 1.000000
Iteration 5, loss = 0.35570365
Validation score: 1.000000
Iteration 6, loss = 0.23548247
Validation score: 1.000000
Iteration 7, loss = 0.24075558
Validation score: 1.000000
Iteration 8, loss = 0.24941599
Validation score: 1.000000
Iteration 9, loss = 0.23632915
Validation score: 1.000000
Iteration 10, loss = 0.23684425
Validation score: 1.000000
Iteration 11, loss = 0.23504069
Validation score: 1.000000
Iteration 12, loss = 0.22606214
Validation score: 1.000000
Iteration 13, loss = 0.20579082
Validation score: 1.000000
Validation score did not improve more than tol=0.000100 for 10 consecutive epo
chs. Stopping.
1.00 accuracy with a standard deviation of 0.00
accuracy score: 1.0

```



In [148...

```

# Generate samples without preprocessing:
new_broken = split_data(Raw_Broken,20)
new_healthy = split_data(Raw_Healthy,20)

samples = new_broken + new_healthy

labels1 = [np.array(0)] * 200
labels2 = [np.array(1)] * 200
labels = labels1 + labels2
X_train_0,X_test_0, y_train_0, y_test_0 = train_test_split(samples,
                                                             labels,test_size=0.3,random_state = random.randint(0,4000))
print("Train data:{}".format(len(X_train_0)))
print("Test data:{}".format(len(X_test_0)))

# Second mlp model:
mlp = MLPClassifier(early_stopping = True, verbose = True,
                    learning_rate_init = 0.01,
                    activation = 'tanh',validation_fraction = 0.1)

scores = cross_val_score(mlp,X_train_0,y_train_0,cv=5)
mlp = mlp.fit(X_train_0,y_train_0)

```

```

print("MLP Without abs or fft:")
print("%0.2f accuracy with a standard deviation of %0.2f" %
      (scores.mean(), scores.std()))

y_pred = mlp.predict(X_test)

print("accuracy score: {}".format(accuracy_score(y_test,y_pred)))

print_confusion_matrix(y_test,y_pred)

```

```

Train data:280
Test data:120
Iteration 1, loss = 0.88650210
Validation score: 0.478261
Iteration 2, loss = 0.74946075
Validation score: 0.521739
Iteration 3, loss = 0.50553361
Validation score: 0.478261
Iteration 4, loss = 0.42716128
Validation score: 0.478261
Iteration 5, loss = 0.37366705
Validation score: 0.391304
Iteration 6, loss = 0.34762853
Validation score: 0.434783
Iteration 7, loss = 0.32213541
Validation score: 0.434783
Iteration 8, loss = 0.31025594
Validation score: 0.478261
Iteration 9, loss = 0.29633030
Validation score: 0.434783
Iteration 10, loss = 0.28673572
Validation score: 0.434783
Iteration 11, loss = 0.27909063
Validation score: 0.434783
Iteration 12, loss = 0.26883696
Validation score: 0.521739
Iteration 13, loss = 0.26587440
Validation score: 0.521739
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 0.87309743
Validation score: 0.652174
Iteration 2, loss = 0.64111220
Validation score: 0.608696
Iteration 3, loss = 0.49602550
Validation score: 0.652174
Iteration 4, loss = 0.42517518
Validation score: 0.782609
Iteration 5, loss = 0.38392374
Validation score: 0.782609
Iteration 6, loss = 0.34959637
Validation score: 0.782609
Iteration 7, loss = 0.32877501
Validation score: 0.782609
Iteration 8, loss = 0.30861360
Validation score: 0.782609
Iteration 9, loss = 0.29618625
Validation score: 0.782609
Iteration 10, loss = 0.28359429
Validation score: 0.782609
Iteration 11, loss = 0.27697575
Validation score: 0.782609
Iteration 12, loss = 0.27069333
Validation score: 0.826087

```

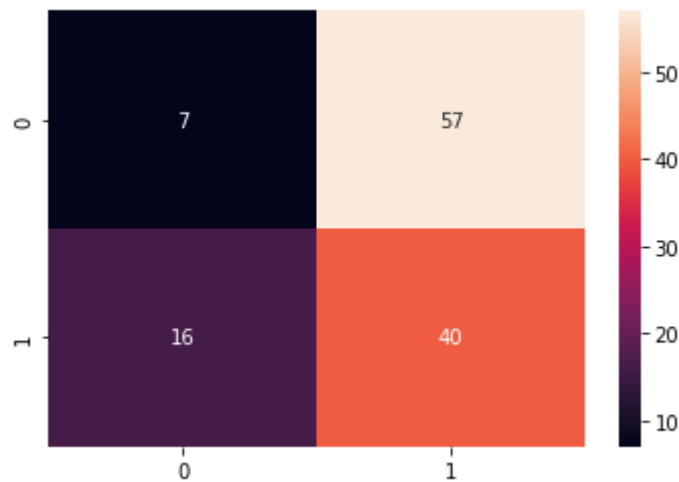
```
Iteration 13, loss = 0.26289494
Validation score: 0.782609
Iteration 14, loss = 0.25801610
Validation score: 0.782609
Iteration 15, loss = 0.24660752
Validation score: 0.826087
Iteration 16, loss = 0.23411525
Validation score: 0.826087
Iteration 17, loss = 0.22547176
Validation score: 0.826087
Iteration 18, loss = 0.21906058
Validation score: 0.782609
Iteration 19, loss = 0.21226829
Validation score: 0.782609
Iteration 20, loss = 0.20750227
Validation score: 0.782609
Iteration 21, loss = 0.20347853
Validation score: 0.782609
Iteration 22, loss = 0.20053214
Validation score: 0.782609
Iteration 23, loss = 0.19924486
Validation score: 0.782609
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 0.85087772
Validation score: 0.608696
Iteration 2, loss = 0.59659868
Validation score: 0.478261
Iteration 3, loss = 0.40004959
Validation score: 0.391304
Iteration 4, loss = 0.33294099
Validation score: 0.391304
Iteration 5, loss = 0.29045958
Validation score: 0.347826
Iteration 6, loss = 0.27331467
Validation score: 0.434783
Iteration 7, loss = 0.26515909
Validation score: 0.434783
Iteration 8, loss = 0.24909859
Validation score: 0.434783
Iteration 9, loss = 0.23798859
Validation score: 0.391304
Iteration 10, loss = 0.22628036
Validation score: 0.391304
Iteration 11, loss = 0.21952888
Validation score: 0.391304
Iteration 12, loss = 0.21229533
Validation score: 0.391304
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 0.92471324
Validation score: 0.260870
Iteration 2, loss = 0.80778793
Validation score: 0.391304
Iteration 3, loss = 0.56470470
Validation score: 0.478261
Iteration 4, loss = 0.45465791
Validation score: 0.521739
Iteration 5, loss = 0.39599225
Validation score: 0.521739
Iteration 6, loss = 0.35537026
Validation score: 0.521739
Iteration 7, loss = 0.31652558
Validation score: 0.521739
```



```
Iteration 8, loss = 0.29469086
Validation score: 0.521739
Iteration 9, loss = 0.27440965
Validation score: 0.521739
Iteration 10, loss = 0.26477047
Validation score: 0.521739
Iteration 11, loss = 0.25608584
Validation score: 0.565217
Iteration 12, loss = 0.24731118
Validation score: 0.521739
Iteration 13, loss = 0.23725408
Validation score: 0.521739
Iteration 14, loss = 0.22986767
Validation score: 0.521739
Iteration 15, loss = 0.22510292
Validation score: 0.521739
Iteration 16, loss = 0.21734517
Validation score: 0.521739
Iteration 17, loss = 0.21122608
Validation score: 0.565217
Iteration 18, loss = 0.20515784
Validation score: 0.608696
Iteration 19, loss = 0.20266695
Validation score: 0.608696
Iteration 20, loss = 0.19784167
Validation score: 0.652174
Iteration 21, loss = 0.19440596
Validation score: 0.695652
Iteration 22, loss = 0.18989724
Validation score: 0.695652
Iteration 23, loss = 0.18360423
Validation score: 0.695652
Iteration 24, loss = 0.18106549
Validation score: 0.695652
Iteration 25, loss = 0.18064160
Validation score: 0.695652
Iteration 26, loss = 0.18107249
Validation score: 0.695652
Iteration 27, loss = 0.17985215
Validation score: 0.695652
Iteration 28, loss = 0.17526525
Validation score: 0.695652
Iteration 29, loss = 0.17041763
Validation score: 0.695652
Iteration 30, loss = 0.16466611
Validation score: 0.652174
Iteration 31, loss = 0.16173558
Validation score: 0.608696
Iteration 32, loss = 0.15583378
Validation score: 0.565217
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 0.85062788
Validation score: 0.434783
Iteration 2, loss = 0.77109176
Validation score: 0.521739
Iteration 3, loss = 0.51035846
Validation score: 0.565217
Iteration 4, loss = 0.39590288
Validation score: 0.565217
Iteration 5, loss = 0.35330772
Validation score: 0.565217
Iteration 6, loss = 0.32769804
Validation score: 0.521739
```

```
Iteration 7, loss = 0.30504560
Validation score: 0.478261
Iteration 8, loss = 0.28932140
Validation score: 0.521739
Iteration 9, loss = 0.26920612
Validation score: 0.521739
Iteration 10, loss = 0.25265334
Validation score: 0.521739
Iteration 11, loss = 0.24219871
Validation score: 0.478261
Iteration 12, loss = 0.23824627
Validation score: 0.478261
Iteration 13, loss = 0.23121207
Validation score: 0.478261
Iteration 14, loss = 0.22162292
Validation score: 0.478261
Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Iteration 1, loss = 0.88934799
Validation score: 0.500000
Iteration 2, loss = 0.46001091
Validation score: 0.464286
Iteration 3, loss = 0.34910265
Validation score: 0.464286
Iteration 4, loss = 0.30224018
Validation score: 0.464286
Iteration 5, loss = 0.27162176
Validation score: 0.571429
Iteration 6, loss = 0.24564441
Validation score: 0.571429
Iteration 7, loss = 0.23501222
Validation score: 0.571429
Iteration 8, loss = 0.22373227
Validation score: 0.571429
Iteration 9, loss = 0.21157392
Validation score: 0.500000
Iteration 10, loss = 0.20067207
Validation score: 0.571429
Iteration 11, loss = 0.19054837
Validation score: 0.571429
Iteration 12, loss = 0.18135655
Validation score: 0.571429
Iteration 13, loss = 0.16957428
Validation score: 0.607143
Iteration 14, loss = 0.16257366
Validation score: 0.571429
Iteration 15, loss = 0.15524738
Validation score: 0.500000
Iteration 16, loss = 0.14529960
Validation score: 0.500000
Iteration 17, loss = 0.13638554
Validation score: 0.535714
Iteration 18, loss = 0.13314434
Validation score: 0.535714
Iteration 19, loss = 0.12285496
Validation score: 0.535714
Iteration 20, loss = 0.11569489
Validation score: 0.500000
Iteration 21, loss = 0.11266977
Validation score: 0.464286
Iteration 22, loss = 0.10777186
Validation score: 0.500000
Iteration 23, loss = 0.10379158
Validation score: 0.500000
```

Iteration 24, loss = 0.10085404  
 Validation score: 0.535714  
 Validation score did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.  
 MLP Without abs or fft:  
 0.54 accuracy with a standard deviation of 0.03  
 accuracy score: 0.39166666666666666



In [149]...

```
# SVM model:
from sklearn.svm import SVC
svm = SVC(kernel = 'sigmoid', verbose = True)
svm = svm.fit(X_train, y_train)
scores = cross_val_score(svm, X_train, y_train, cv=5)
y_pred = svm.predict(X_test)
print("%0.2f accuracy with a standard deviation of %0.2f"
      % (scores.mean(), scores.std()))
print("accuracy score: {}".format(accuracy_score(y_test, y_pred)))

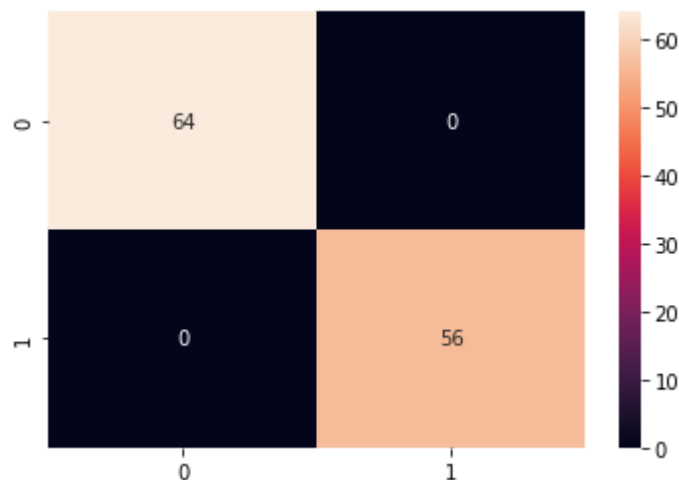
print_confusion_matrix(y_test, y_pred)
```

```
[LibSVM]*
optimization finished, #iter = 79
obj = -15.123307, rho = -1.320310
nSV = 53, nBSV = 17
Total nSV = 53
[LibSVM]*
optimization finished, #iter = 88
obj = -14.861742, rho = -1.345464
nSV = 51, nBSV = 17
Total nSV = 51
[LibSVM]*
optimization finished, #iter = 104
obj = -14.631133, rho = -1.356082
nSV = 51, nBSV = 14
Total nSV = 51
[LibSVM]*
optimization finished, #iter = 119
obj = -14.597557, rho = -1.211054
nSV = 61, nBSV = 13
Total nSV = 61
[LibSVM]*
optimization finished, #iter = 85
obj = -14.661125, rho = -1.297077
nSV = 50, nBSV = 18
Total nSV = 50
[LibSVM]*
optimization finished, #iter = 125
```

```

obj = -14.879209, rho = -1.284382
nSV = 52, nBSV = 16
Total nSV = 52
1.00 accuracy with a standard deviation of 0.00
accuracy score: 1.0

```



In [150]...

```

# Second SVM model:
from sklearn.svm import SVC
svm = SVC(kernel = 'sigmoid', verbose = True)
svm = svm.fit(X_train_0, y_train_0)
scores = cross_val_score(svm, X_train_0, y_train_0, cv=5)
y_pred = svm.predict(X_test)
print("%0.2f accuracy with a standard deviation of %0.2f"
      % (scores.mean(), scores.std()))
print("accuracy score: {}".format(accuracy_score(y_test, y_pred)))

print_confusion_matrix(y_test, y_pred)

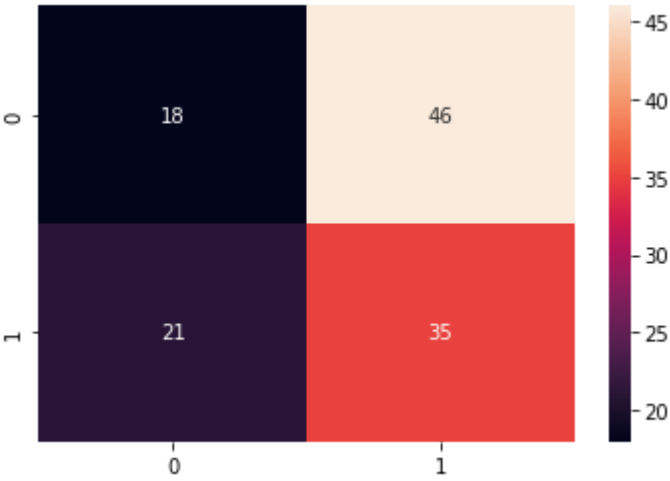
```

```

[LibSVM]*
optimization finished, #iter = 242
obj = -175.581984, rho = -0.137646
nSV = 279, nBSV = 210
Total nSV = 279
[LibSVM]*
optimization finished, #iter = 189
obj = -140.707395, rho = -0.160764
nSV = 224, nBSV = 168
Total nSV = 224
[LibSVM]*
optimization finished, #iter = 173
obj = -140.087052, rho = -0.081102
nSV = 223, nBSV = 175
Total nSV = 223
[LibSVM]*
optimization finished, #iter = 186
obj = -141.729755, rho = -0.128446
nSV = 223, nBSV = 175
Total nSV = 223
[LibSVM]*
optimization finished, #iter = 183
obj = -141.002449, rho = -0.162941
nSV = 222, nBSV = 178
Total nSV = 222
[LibSVM]*
optimization finished, #iter = 182
obj = -140.234395, rho = -0.196554
nSV = 224, nBSV = 172
Total nSV = 224

```

0.59 accuracy with a standard deviation of 0.02  
accuracy score: 0.44166666666666665



```
In [ ]:
```