

dtw

April 15, 2016

```
In [1]: from fastdtw import fastdtw
        from scipy.spatial.distance import euclidean

        #minta szerint
        #szerintem ez a jó
        def timeseries_second():
            last_keydown = ''
            timeseries = []
            row = data_in.readline()
            row = row.split()
            while row:
                if row[0] == 'keydown':

                    if last_keydown :
                        timeseries.append(int(row[5])-int(last_keydown))
                        last_keydown = row[5]
                    row = data_in.readline()
                    row = row.split()
            return timeseries

        data_in = open("C:/Users/valikund/Desktop/Challenge/keystrokes-12users-raw-data.txt","r")
        training_in = open("C:/Users/valikund/Desktop/Challenge/keystrokes-12users-train-labels.txt","r")
        test_in = open("C:/Users/valikund/Desktop/Challenge/keystrokes-12users-test-hypothetic-labels.txt","r")
        data_x = []
        data_y = []
        row = data_in.readline().strip()
        while row:
            if 'TYPING PATTERN' in row:
                data_x.append(timeseries_second())
                row = data_in.readline().strip()

        data_in.close()
        #separating the train and test dataset
        train_x = []
        train_y = []
        test_x = []
        test_y = []
        test_index = []
        row = training_in.readline().strip()
        while row:
            row = row.split(',')
            train_x.append(data_x[int(row[0])-1])
            train_y.append(int(row[1]))
```

```

        row = training_in.readline()
training_in.close()
#the test datasets
row = test_in.readline().strip()
while row:
    row = row.split(',')
    test_x.append(data_x[int(row[0])-1])
    test_index.append(int(row[0]))
    test_y.append(int(row[1]))
    row = test_in.readline()
test_in.close()

data_length = []
for data in data_x:
    data_length.append(len(data))

distance, path = fastdtw(train_x[0], train_x[25], dist=euclidean)
print(distance)

40455.0

In [2]: import itertools
def most_common(L):
    groups = itertools.groupby(sorted(L))
    def _auxfun((item, iterable)):
        return len(list(iterable)), -L.index(item)
    return max(groups, key=_auxfun)[0]

In [3]: def timeseries_first():
    keydown = ''
    timeseries = []
    row = data_in.readline()
    row = row.split()
    while row:
        if row[0] == 'keydown':
            keydown = row[5]
        if keydown and row[0] == 'keyup':
            timeseries.append(int(row[5])-int(keydown))
            keydown = ''
        row = data_in.readline()
        row = row.split()
    return timeseries

data_in = open("C:/Users/valikund/Desktop/Challenge/keystrokes-12users-raw-data.txt", "r")
training_in = open("C:/Users/valikund/Desktop/Challenge/keystrokes-12users-train-labels.txt", "r")
test_in = open("C:/Users/valikund/Desktop/Challenge/keystrokes-12users-test-hypothetic-labels.txt", "r")
data_x_1 = []
data_y_1 = []
row = data_in.readline().strip()
while row:
    if 'TYPING PATTERN' in row:
        data_x_1.append(timeseries_first())
    row = data_in.readline().strip()

data_in.close()

```

```

#separating the train and test dataset
train_x_1 = []
train_y_1 = []
test_x_1 = []
test_y_1 = []
row = training_in.readline().strip()
while row:
    row = row.split(',')
    train_x_1.append(data_x[int(row[0])-1])
    train_y_1.append(int(row[1]))
    row = training_in.readline()
training_in.close()
#the test datasets
row = test_in.readline().strip()
while row:
    row = row.split(',')
    test_x_1.append(data_x[int(row[0])-1])
    test_y_1.append(int(row[1]))
    row = test_in.readline()
test_in.close()

data_length = []
for data in data_x:
    data_length.append(len(data))

distance, path = fastdtw(train_x[0], train_x[25], dist=euclidean)
print(distance)

```

40455.0

```

In [4]: def predict(train_x,train_y,test_x, k):
        index = 0
        knn = []
        knn_label = []
        for x in train_x:
            distance, path = fastdtw(x, test_x, dist=euclidean)
            if len(knn)<k :
                knn.append(distance)
                knn_label.append(train_y[index])
            elif max(knn)>distance:
                remove_item = knn.index(max(knn))
                knn.pop(remove_item)
                knn_label.pop(remove_item)
                knn.append(distance)
                knn_label.append(train_y[index])
            index = index + 1
        return knn_label

```

```

In [7]: train_x_sorted = []
        train_y_arranged = np.array(train_y)
        sort_index = np.argsort(train_y_arranged)
        for position in sort_index:
            train_x_sorted.append(train_x[position])
        train_y.sort()
        print(train_y_arranged, train_y)

```

```
(array([ 2,  2,  2,  2,  1,  3,  2,  3,  3,  3,  3,  1,  1,  4,  4,  4,  1,
        1,  4,  4, 12, 12, 12, 12, 12, 10, 10, 10, 10, 10,  7,  7,  7,  7,
        7, 11, 11, 11, 11, 11,  5,  5,  5,  5,  5,  8,  8,  8,  8,  8,  6,
        6,  6,  6,  9,  9,  9,  9,  9]), [1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4,
```

[0.	57382.	52965.	60451.	67179.	69730.	72022.	58292.	70434.]
	61729.	60162.	73598.	60544.	73872.	68264.	63823.	69766.	67746.
	78625.	70195.	56073.	61667.	63817.	59159.	53302.	71790.	72001.
	66396.	69162.	69485.	66841.	64992.	64891.	61162.	65467.	71017.
	55983.	77792.	63066.	60053.	67497.	69215.	70525.	70038.	70908.
	63564.	62430.	60774.	62301.	70260.	66066.	66128.	79773.	67292.]
	94774.	72362.	65733.	69525.	63772.	69348.]			
[57382.	0.	49334.	58816.	49095.	48691.	50242.	48084.	50099.]
	48521.	55833.	53276.	48955.	54133.	58478.	50310.	48355.	45354.]
	58392.	63470.	44460.	46433.	42105.	42689.	42518.	51718.	49712.]
	50064.	48494.	46960.	45820.	48146.	45911.	52445.	45920.	43812.]
	41027.	50517.	47499.	41151.	51536.	43585.	44309.	46563.	45243.]
	46308.	44618.	44376.	53828.	44890.	50047.	40071.	63919.	53166.]
	61917.	52429.	46598.	47618.	43745.	57126.]			
[52965.	49334.	0.	65467.	66366.	59995.	58880.	56480.	67894.]
	62892.	63284.	64560.	57776.	57498.	63443.	60035.	66932.	66285.]
	56326.	77494.	58649.	56531.	60260.	53069.	50934.	65231.	64040.]
	64250.	61656.	68143.	57885.	63629.	60676.	60440.	59712.	58847.]
	48531.	60506.	59381.	54094.	52094.	60080.	56151.	59867.	56977.]
	58773.	67730.	68474.	64142.	57038.	59676.	55803.	62340.	58145.]
	68757.	61439.	63122.	66888.	62667.	64530.]			
[60451.	58816.	65467.	0.	64320.	61542.	62740.	56816.	63114.]
	68993.	67511.	75259.	67810.	63745.	73851.	68290.	74363.	54880.]
	67001.	75183.	58337.	62540.	62188.	47666.	59910.	65475.	67975.]
	65291.	69414.	67529.	58029.	65521.	60087.	60947.	60991.	58404.]
	53442.	69997.	63962.	57461.	71535.	55809.	68289.	62106.	55863.]
	60708.	59041.	62908.	66771.	60903.	54563.	57010.	81000.	70560.]
	88214.	61194.	63016.	59538.	57889.	72069.]			
[67179.	49095.	66366.	64320.	0.	62302.	74487.	66217.	74726.]
	73272.	70321.	56451.	69566.	60869.	71629.	74180.	66404.	66705.]
	67805.	92268.	62888.	64133.	63717.	61701.	62158.	70412.	70204.]
	69351.	72226.	72199.	66155.	67967.	67228.	67500.	67420.	62546.]
	62252.	75636.	70105.	61224.	63578.	68823.	67136.	67184.	70081.]
	65527.	67440.	69739.	60110.	64765.	64777.	62423.	83536.	61088.]
	85765.	69417.	67299.	70793.	68705.	72477.]			
[69730.	48691.	59995.	61542.	62302.	0.	35893.	28432.	40410.]
	47835.	42741.	41561.	40385.	44653.	54562.	55885.	51758.	42166.]
	67176.	68848.	40697.	49977.	40321.	37911.	39433.	36341.	33680.]
	35341.	33775.	38779.	32331.	33353.	33256.	34524.	33274.	43550.]
	38491.	41638.	38350.	43544.	42832.	40945.	43089.	37500.	34155.]
	36858.	36095.	33588.	38073.	37094.	40483.	39806.	55569.	47894.]

55663.	36714.	36693.	40463.	41294.	36370.]				
[72022.	50242.	58880.	62740.	74487.	35893.	0.	33633.	39162.	
45620.	49284.	45154.	41651.	42517.	55896.	51550.	43854.	43658.	
64542.	71041.	37566.	44425.	37622.	30808.	44624.	38534.	37331.	
36558.	39350.	41516.	35187.	34677.	32791.	33601.	33704.	47489.	
37662.	44785.	41688.	47492.	46732.	37546.	39674.	39051.	41660.	
39282.	37874.	32716.	40455.	40884.	37248.	33720.	50964.	49095.	
53753.	43612.	41334.	37469.	42074.	40029.]				
[58292.	48084.	56480.	56816.	66217.	28432.	33633.	0.	35005.	
40710.	55723.	33199.	37409.	42443.	60312.	51149.	46117.	43412.	
58632.	62231.	43084.	48812.	36410.	28463.	41025.	37258.	33552.	
33892.	44715.	42907.	29095.	33085.	31376.	40951.	33064.	41967.	
42436.	35092.	46920.	42335.	40536.	34670.	42483.	37300.	35661.	
29752.	41307.	38004.	41939.	32191.	41526.	36053.	48867.	48448.	
51769.	41442.	32604.	35698.	38430.	34916.]				
[70434.	50099.	67894.	63114.	74726.	40410.	39162.	35005.	0.	
40541.	50115.	44602.	45803.	44998.	60898.	55380.	50040.	55535.	
70555.	66963.	42193.	49571.	43197.	36761.	41540.	44952.	42397.	
43928.	43834.	43197.	40933.	41293.	44343.	40140.	40881.	48841.	
40305.	39441.	43451.	41200.	49880.	45420.	43248.	44958.	45797.	
36767.	42091.	44234.	46821.	40350.	44598.	41554.	61619.	55980.	
55656.	42206.	41048.	40520.	47856.	40680.]				
[61729.	48521.	62892.	68993.	73272.	47835.	45620.	40710.	40541.	
0.	59861.	46463.	39937.	51466.	62384.	46092.	47536.	37695.	
71326.	64140.	50308.	53668.	37968.	36574.	43576.	46420.	41795.	
45194.	49739.	47615.	38727.	40753.	38394.	39305.	38113.	50833.	
42726.	50262.	47644.	45974.	45847.	42952.	48915.	43999.	44619.	
43082.	42363.	33754.	47480.	37108.	41601.	50044.	50738.	51851.	
45562.	43412.	38180.	36751.	39381.	42787.]				
[60162.	55833.	63284.	67511.	70321.	42741.	49284.	55723.	50115.	
59861.	0.	46222.	49250.	45960.	61898.	55299.	52955.	53343.	
61590.	71962.	43469.	45071.	45542.	44974.	43108.	53334.	50083.	
51293.	54815.	55649.	46532.	49111.	48767.	46902.	49079.	56446.	
49025.	43980.	53322.	50960.	48726.	48436.	57175.	54020.	46720.	
47534.	51454.	46178.	53473.	50494.	49635.	51944.	61602.	55087.	
65672.	51328.	50456.	51135.	49911.	49776.]				
[73598.	53276.	64560.	75259.	56451.	41561.	45154.	33199.	44602.	
46463.	46222.	0.	42259.	41265.	60979.	58523.	48291.	46341.	
65638.	68262.	41278.	43952.	44216.	39348.	40454.	44730.	46193.	
42625.	48888.	48190.	40220.	46811.	42571.	40051.	44062.	48091.	
40278.	48920.	42980.	43001.	42529.	41617.	48113.	41760.	42505.	
44930.	41249.	40526.	43176.	50783.	47115.	43817.	51945.	52038.	
58443.	49818.	49704.	46842.	56869.	47072.]				
[60544.	48955.	57776.	67810.	69566.	40385.	41651.	37409.	45803.	
39937.	49250.	42259.	0.	44235.	59336.	47132.	42363.	35204.	
67321.	62975.	38506.	49224.	39663.	33798.	41596.	44717.	45618.	
41923.	46832.	45313.	40535.	43852.	40605.	41732.	38784.	44971.	
47815.	39057.	40856.	45228.	40130.	40667.	39457.	46094.	39766.	
38899.	42295.	39085.	42613.	40943.	41833.	43339.	59280.	43577.	
44937.	45117.	47481.	40305.	43311.	39765.]				
[73872.	54133.	57498.	63745.	60869.	44653.	42517.	42443.	44998.	
51466.	45960.	41265.	44235.	0.	49464.	49270.	52496.	48393.	
57931.	71905.	41559.	43786.	41396.	38398.	41928.	46652.	44450.	
44027.	44069.	47022.	40767.	42799.	44212.	39084.	43463.	48130.	

43516.	44100.	45976.	47868.	44724.	41222.	46489.	47253.	47149.
45244.	43990.	39508.	45960.	43000.	45436.	44316.	53428.	49571.
67301.	50171.	44188.	45812.	56466.	44514.]			
[68264.	58478.	63443.	73851.	71629.	54562.	55896.	60312.	60898.
62384.	61898.	60979.	59336.	49464.	0.	63098.	67062.	62182.
72681.	83307.	53580.	59698.	55568.	58022.	54517.	63919.	63351.
62743.	63066.	63155.	58361.	62852.	61257.	66883.	59680.	62771.
55083.	62204.	59981.	61902.	63705.	61324.	55295.	54366.	55013.
57916.	58528.	53109.	64223.	59400.	55345.	52236.	65412.	58683.
72271.	60109.	57935.	62041.	64041.	61374.]			
[63823.	50310.	60035.	68290.	74180.	55885.	51550.	51149.	55380.
46092.	55299.	58523.	47132.	49270.	63098.	0.	52100.	46103.
57551.	81064.	42401.	47568.	42698.	53178.	45204.	55396.	53383.
55096.	53177.	54412.	50936.	54213.	49237.	49419.	49972.	48661.
53445.	59432.	52953.	45980.	49920.	51559.	46126.	52498.	52145.
51300.	48534.	43979.	55625.	48780.	45712.	47454.	61595.	57459.
56151.	58819.	52419.	51260.	54720.	55803.]			
[69766.	48355.	66932.	74363.	66404.	51758.	43854.	46117.	50040.
47536.	52955.	48291.	42363.	52496.	67062.	52100.	0.	40492.
62135.	69459.	43377.	48876.	41549.	39989.	50133.	48961.	44721.
50763.	56183.	54908.	46965.	46779.	46960.	46877.	49298.	50972.
44916.	48690.	46434.	43307.	45373.	41265.	43048.	47590.	48472.
40562.	46258.	45432.	48762.	43454.	41222.	54376.	55522.	52461.
61338.	45855.	46834.	50308.	44474.	47013.]			
[67746.	45354.	66285.	54880.	66705.	42166.	43658.	43412.	55535.
37695.	53343.	46341.	35204.	48393.	62182.	46103.	40492.	0.
63306.	57403.	39590.	41304.	42774.	33516.	41201.	37833.	39613.
37566.	40614.	38042.	37159.	41638.	31920.	38094.	34072.	42020.
43531.	48512.	42712.	43892.	43840.	37841.	43119.	42822.	38361.
38949.	38265.	32692.	46268.	38304.	34870.	38619.	48570.	46111.
52038.	45588.	37676.	41828.	42975.	47975.]			
[78625.	58392.	56326.	67001.	67805.	67176.	64542.	58632.	70555.
71326.	61590.	65638.	67321.	57931.	72681.	57551.	62135.	63306.
0.	78140.	56133.	54184.	56707.	57047.	52914.	64645.	60525.
62116.	62622.	65528.	58902.	60879.	63822.	66021.	56807.	53999.
56749.	58555.	53674.	61024.	54028.	55395.	59695.	60358.	61062.
53413.	59434.	55257.	65043.	64813.	55948.	60415.	64853.	61131.
74982.	69461.	63920.	64599.	67202.	60772.]			
[70195.	63470.	77494.	75183.	92268.	68848.	71041.	62231.	66963.
64140.	71962.	68262.	62975.	71905.	83307.	81064.	69459.	57403.
78140.	0.	55799.	54564.	56797.	59041.	51674.	65820.	63718.
62968.	63896.	65601.	62538.	64798.	61765.	56200.	59694.	67370.
60267.	65972.	54753.	67115.	70497.	56285.	59357.	61761.	61116.
62764.	60131.	58105.	63373.	63795.	60391.	65483.	62268.	73177.
73144.	63539.	62361.	58654.	66595.	69671.]			
[56073.	44460.	58649.	58337.	62888.	40697.	37566.	43084.	42193.
50308.	43469.	41278.	38506.	41559.	53580.	42401.	43377.	39590.
56133.	55799.	0.	36048.	33363.	27263.	30014.	36714.	33440.
35847.	38393.	36759.	31165.	32989.	30722.	30714.	30818.	37836.
34654.	42699.	38319.	38928.	37065.	37107.	39932.	34803.	33994.
36000.	33202.	36679.	37211.	31687.	33983.	36916.	42002.	41949.
54412.	38272.	36015.	35201.	44991.	41157.]			
[61667.	46433.	56531.	62540.	64133.	49977.	44425.	48812.	49571.
53668.	45071.	43952.	49224.	43786.	59698.	47568.	48876.	41304.

54184.	54564.	36048.	0.	34468.	33369.	32287.	46008.	43832.
45333.	47302.	45050.	40427.	40589.	40436.	38187.	39180.	43359.
39805.	43900.	37747.	41336.	41565.	38667.	44885.	44663.	37742.
42076.	40900.	37191.	39461.	41132.	44878.	43368.	52635.	55342.
60325.	41758.	45936.	45776.	46031.	43791.]			
[63817.	42105.	60260.	62188.	63717.	40321.	37622.	36410.	43197.
37968.	45542.	44216.	39663.	41396.	55568.	42698.	41549.	42774.
56707.	56797.	33363.	34468.	0.	21152.	31423.	32021.	37114.
32070.	34580.	33226.	34078.	28845.	29210.	31478.	26780.	39199.
32553.	39056.	36765.	38667.	41856.	31417.	36651.	36702.	33182.
34414.	32202.	28013.	43680.	33750.	32024.	33739.	49425.	45523.
50091.	47017.	33656.	34266.	38860.	37391.]			
[59159.	42689.	53069.	47666.	61701.	37911.	30808.	28463.	36761.
36574.	45004.	39348.	33798.	38398.	58022.	53178.	39989.	33516.
57047.	59041.	27263.	33369.	21152.	0.	33390.	32859.	29824.
33972.	32794.	31615.	26308.	28150.	26848.	26811.	27749.	43324.
31302.	33960.	32327.	35678.	34407.	27409.	37312.	35142.	31874.
31028.	30127.	25074.	34713.	26922.	29495.	38159.	39297.	44931.
42164.	40748.	31472.	33229.	39161.	35445.]			
[53302.	42518.	50934.	59910.	62158.	39433.	44624.	41025.	41540.
43576.	43108.	40454.	41596.	41928.	54517.	45204.	50133.	41201.
52914.	51674.	30014.	32401.	31423.	33390.	0.	36729.	34921.
38136.	38351.	40643.	35102.	33908.	33755.	28600.	31354.	40559.
34326.	37047.	39182.	37588.	41573.	36614.	39237.	39866.	34162.
36751.	31736.	36863.	43037.	32829.	34809.	35471.	47811.	48733.
56342.	35373.	33189.	36661.	40127.	35631.]			
[71790.	51718.	65231.	65475.	70412.	36341.	38534.	37258.	44952.
46420.	53334.	44730.	44717.	46652.	63919.	55396.	48961.	37833.
64645.	65820.	36714.	46008.	32021.	32859.	36729.	0.	12845.
13872.	18702.	16035.	18402.	18786.	15515.	27147.	16804.	40994.
41006.	41225.	41741.	43057.	39282.	34115.	37582.	36986.	42254.
25603.	21743.	28137.	39465.	20890.	36859.	40426.	47980.	49362.
48665.	25148.	18471.	27904.	27718.	30031.]			
[72001.	49712.	64040.	67975.	70204.	33680.	37331.	33552.	42397.
41795.	50083.	46193.	45618.	44450.	63351.	53383.	44721.	39613.
60525.	63718.	33440.	43832.	37114.	29824.	34921.	12845.	0.
13592.	18053.	16920.	16374.	16344.	16569.	25821.	15600.	43899.
36966.	34018.	39009.	43327.	36947.	32348.	33809.	33489.	32954.
26115.	26424.	25022.	38422.	17593.	34964.	36605.	44855.	45833.
45472.	24403.	20287.	25271.	26896.	28945.]			
[66396.	50064.	64250.	65291.	69351.	35341.	36558.	33892.	43928.
45194.	51293.	42625.	41923.	44027.	62743.	55096.	50763.	37566.
62116.	62968.	35847.	45333.	32070.	33972.	38136.	13872.	13592.
0.	13242.	20168.	17072.	20576.	17657.	27131.	18812.	41542.
38377.	38236.	41184.	43346.	37327.	32930.	36620.	35535.	34220.
24489.	22717.	27065.	37795.	21435.	35542.	36484.	48086.	49001.
41193.	25505.	22787.	28051.	24642.	30772.]			
[69162.	48494.	61656.	69414.	72226.	33775.	39350.	44715.	43834.
49739.	54815.	48888.	46832.	44069.	63066.	53177.	56183.	40614.
62622.	63896.	38393.	47302.	34580.	32794.	38351.	18702.	18053.
13242.	0.	14900.	23137.	22060.	17705.	27928.	18850.	42014.
41938.	38132.	37778.	40356.	48689.	33045.	35653.	38928.	35133.
29728.	30292.	27381.	42629.	21744.	36512.	39076.	42661.	50404.
45882.	30389.	22923.	25849.	26952.	30024.]			

[69485.	46960.	68143.	67529.	72199.	38779.	41516.	42907.	43197.
	47615.	55649.	48190.	45313.	47022.	63155.	54412.	54908.	38042.
	65528.	65601.	36759.	45050.	33226.	31615.	40643.	16035.	16920.
	20168.	14900.	0.	19377.	20810.	16743.	25791.	17344.	40900.
	39477.	40731.	38655.	41198.	46543.	36203.	35602.	36607.	34432.
	33177.	23887.	27186.	44135.	21776.	36396.	38726.	43037.	48401.
	44815.	26120.	21393.	26461.	32404.	30717.]			
[66841.	45820.	57885.	58029.	66155.	32331.	35187.	29095.	40933.
	38727.	46532.	40220.	40535.	40767.	58361.	50936.	46965.	37159.
	58902.	62538.	31165.	40427.	34078.	26308.	35102.	18402.	16374.
	17072.	23137.	19377.	0.	17330.	16660.	22821.	13975.	38484.
	35093.	36220.	36791.	36996.	37689.	29239.	35762.	33694.	33671.
	27478.	20306.	23056.	39167.	20318.	31940.	33239.	42201.	44351.
	48015.	28364.	26964.	26964.	30882.	29317.]			
[64992.	48146.	63629.	65521.	67967.	33353.	34914.	33085.	41293.
	40753.	49111.	46811.	43852.	42799.	62852.	54213.	46779.	41638.
	60879.	64798.	32989.	40589.	28845.	28150.	33908.	18786.	16344.
	20576.	22060.	20810.	17330.	0.	17816.	24936.	18561.	42810.
	37506.	31775.	42457.	40431.	37884.	33497.	35468.	34551.	33265.
	27844.	23183.	25698.	36557.	20089.	35519.	34059.	43791.	47328.
	45828.	28108.	22967.	29809.	29367.	28856.]			
[64891.	45911.	60676.	60087.	67228.	33256.	32791.	31376.	44343.
	38394.	48767.	42571.	40605.	44212.	61257.	49237.	46960.	31920.
	63822.	61765.	30722.	40436.	29210.	26848.	33755.	15515.	16569.
	17657.	17705.	16743.	16660.	17816.	0.	21706.	15895.	38549.
	36407.	32397.	37565.	35943.	34292.	34113.	36842.	37619.	36264.
	26566.	22016.	21643.	35835.	17372.	30234.	31043.	46252.	43712.
	46485.	22550.	21752.	26735.	28986.	28229.]			
[61162.	52445.	60440.	60947.	67500.	34798.	33601.	40951.	40140.
	39305.	46902.	40051.	41732.	39084.	66883.	49419.	46877.	38094.
	66021.	56200.	30714.	38187.	31478.	26811.	28600.	27147.	25821.
	27131.	27928.	25791.	22821.	24936.	21706.	0.	22526.	38294.
	39306.	34609.	38935.	40928.	34012.	31568.	33462.	31734.	33659.
	32610.	27956.	27989.	36447.	25326.	35906.	32235.	50534.	41076.
	51576.	29658.	28130.	31985.	39694.	34815.]			
[65467.	45920.	59712.	60991.	67420.	33274.	33704.	33064.	40881.
	38113.	49079.	44062.	38784.	43463.	59680.	49972.	49298.	34072.
	56807.	59694.	30818.	39180.	26780.	27749.	31354.	16804.	15600.
	18812.	18850.	17344.	13975.	18561.	15895.	22526.	0.	37982.
	31140.	33518.	36068.	40145.	34188.	29732.	34168.	32998.	30966.
	27674.	22046.	23554.	34804.	18822.	30655.	33969.	41919.	43514.
	48464.	27880.	26006.	25748.	30983.	29505.]			
[71017.	43812.	58847.	58404.	62546.	43550.	47489.	41967.	48841.
	50833.	56446.	48091.	44971.	48130.	62771.	48661.	50972.	42020.
	53999.	67370.	37836.	43359.	39199.	43324.	40559.	40994.	43899.
	41542.	42014.	40900.	38484.	42810.	38549.	38294.	37982.	0.
	43619.	47345.	41254.	41386.	39184.	39982.	42688.	41477.	39384.
	37709.	39293.	33331.	46329.	39261.	38818.	37304.	52590.	49456.
	52957.	48192.	40315.	42767.	46083.	48793.]			
[55983.	41027.	48531.	53442.	62252.	38491.	37662.	42436.	40305.
	42726.	49025.	40278.	47815.	43516.	55083.	53445.	44916.	43531.
	56749.	60267.	34654.	39805.	32553.	31302.	34326.	41006.	36966.
	38377.	41938.	39477.	35093.	37506.	36407.	39306.	31140.	43619.
	0.	37792.	39703.	38417.	38891.	34561.	38459.	38264.	35587.

35747.	40788.	35226.	41385.	37395.	35463.	38924.	46889.	51224.
53195.	40719.	43609.	43619.	39634.	48684.]			
[77792.	50517.	60506.	69997.	75636.	41638.	44785.	35092.	39441.
50262.	43980.	48920.	39057.	44100.	62204.	59432.	48690.	48512.
58555.	65972.	42699.	43900.	39056.	33960.	37047.	41225.	34018.
38236.	38132.	40731.	36220.	31775.	32397.	34609.	33518.	47345.
37792.	0.	45744.	46640.	43630.	39866.	37933.	37955.	39570.
42021.	35435.	29126.	41862.	34134.	40741.	35753.	45805.	47487.
51921.	37070.	34876.	38966.	34258.	38463.]			
[63066.	47499.	59381.	63962.	70105.	38350.	41688.	46920.	43451.
47644.	53322.	42980.	40856.	45976.	59981.	52953.	46434.	42712.
53674.	54753.	38319.	37747.	36765.	32327.	39182.	41741.	38995.
41184.	37778.	38655.	36791.	42457.	37565.	38935.	36068.	41254.
39703.	45744.	0.	42133.	37945.	34792.	43225.	38026.	36523.
36667.	39669.	36693.	36468.	40499.	41680.	39255.	53505.	50498.
52995.	47850.	45784.	40912.	49233.	43595.]			
[60053.	41151.	54094.	57461.	61224.	43544.	47492.	42335.	41200.
45974.	50960.	43001.	45228.	47868.	61902.	45980.	43307.	43892.
61024.	67115.	38928.	41336.	38667.	35678.	37588.	43057.	43327.
43346.	40356.	41198.	36996.	40431.	35943.	40928.	40145.	41386.
38417.	46640.	42133.	0.	45833.	37808.	42391.	41281.	38008.
37557.	42768.	37038.	39810.	45191.	42647.	44426.	54083.	52265.
55471.	50119.	42664.	48330.	47336.	46445.]			
[67497.	51536.	52094.	71535.	63578.	42832.	46732.	40536.	49880.
45847.	48726.	42529.	40130.	44724.	63705.	49920.	45373.	43840.
54028.	70497.	37065.	41565.	41856.	34407.	41573.	39282.	36947.
37327.	48689.	46543.	37689.	37884.	34292.	34012.	34188.	39184.
38891.	43630.	37945.	45833.	0.	34583.	36908.	37507.	32003.
40977.	48004.	34682.	41649.	40637.	36295.	41871.	52721.	46295.
49776.	42704.	37747.	43483.	39939.	40684.]			
[69215.	43585.	60080.	55809.	68823.	40945.	37546.	35597.	45420.
42952.	48436.	41617.	40667.	41222.	61324.	51559.	40674.	37841.
55395.	56285.	37107.	38667.	31417.	27409.	36614.	34115.	32348.
32930.	33045.	36203.	29239.	33497.	34113.	31568.	29732.	39982.
34561.	39866.	34792.	37808.	34583.	0.	36940.	34134.	36549.
29387.	29513.	32034.	42960.	32229.	33359.	39877.	43680.	45462.
49430.	37659.	32498.	34204.	34370.	35547.]			
[70525.	44309.	56151.	68289.	67136.	43089.	39674.	42483.	43248.
48915.	57175.	48113.	39457.	46489.	55295.	46126.	43048.	43119.
59695.	59357.	39932.	44885.	36651.	37312.	39237.	37582.	33809.
36620.	35653.	35602.	35762.	35468.	36842.	33462.	34168.	42688.
38459.	37933.	43225.	42391.	36908.	36940.	0.	38471.	35911.
38710.	35222.	31156.	41479.	35828.	37658.	36048.	43243.	47731.
54020.	35016.	39648.	37579.	44317.	38945.]			
[70038.	46563.	59867.	62106.	67184.	37500.	39051.	37300.	44958.
43999.	54020.	41760.	46094.	47253.	54366.	52498.	47590.	42822.
60358.	61761.	34803.	44663.	36702.	35142.	39866.	36986.	33489.
35535.	38928.	36607.	33694.	34551.	37619.	31734.	32998.	41477.
38264.	37955.	38026.	41281.	37507.	34134.	38471.	0.	38059.
34360.	34409.	32786.	48960.	32224.	38965.	33831.	47967.	41206.
57642.	38274.	36192.	38746.	41560.	37967.]			
[70908.	45243.	56977.	55863.	70081.	34155.	41660.	35661.	45797.
44619.	46720.	42505.	39766.	47149.	55013.	52145.	48472.	38361.
61062.	61116.	33994.	37742.	33182.	31874.	34162.	42254.	32954.

34220.	35133.	34432.	33671.	33265.	36264.	33659.	30966.	39384.
35587.	39570.	36523.	38008.	32003.	36549.	35911.	38059.	0.
34353.	33264.	34010.	43461.	30804.	36681.	32207.	45252.	44766.
49741.	37509.	33353.	39095.	44216.	39237.]			
[63564.	46308.	58773.	60708.	65527.	36858.	39282.	29752.	36767.
43082.	47534.	44930.	38899.	45244.	57916.	51300.	40562.	38949.
53413.	62764.	36000.	42076.	34414.	31028.	36751.	25603.	26115.
24489.	29728.	33177.	27478.	27844.	26566.	32610.	27674.	37709.
35747.	42021.	36667.	37557.	40977.	29387.	38710.	34360.	34353.
0.	27090.	28113.	41226.	25814.	34761.	34788.	40817.	43026.
47243.	31600.	28161.	29325.	32443.	30493.]			
[62430.	44618.	67730.	59041.	67440.	36095.	37874.	41307.	42091.
42363.	51454.	41249.	42295.	43990.	58528.	48534.	46258.	38265.
59434.	60131.	33202.	40900.	32202.	30127.	31736.	21743.	26424.
22717.	29293.	23887.	20306.	23183.	22016.	27956.	22046.	39293.
40788.	35435.	39669.	42768.	48004.	29513.	35222.	34409.	33264.
27090.	0.	24357.	43134.	21548.	35187.	31080.	40834.	41643.
51112.	27997.	33369.	34811.	27273.	29180.]			
[60774.	44376.	68474.	62908.	69739.	33588.	32716.	38004.	44234.
33754.	46178.	40526.	39085.	39508.	53109.	43979.	45432.	32692.
55257.	58105.	36679.	37191.	28013.	25074.	36863.	28137.	25022.
27065.	27381.	27186.	23056.	25698.	21643.	27989.	23554.	33331.
35226.	29126.	36693.	37038.	34682.	32034.	31156.	32786.	34010.
28113.	24357.	0.	35301.	22968.	30497.	31249.	41123.	45347.
45161.	35562.	24753.	32508.	34967.	36037.]			
[62301.	53828.	64142.	66771.	60110.	38073.	40455.	41939.	46821.
47480.	53473.	43176.	42613.	45960.	64223.	55625.	48762.	46268.
65043.	63373.	37211.	39461.	43680.	34713.	43037.	39465.	38422.
37795.	42629.	44135.	39167.	36557.	35835.	36447.	34804.	46329.
41385.	41862.	36468.	39810.	41649.	42960.	41479.	48960.	43461.
41226.	43134.	35301.	0.	42667.	44643.	49172.	50194.	60763.
54172.	46869.	39602.	42169.	44394.	44711.]			
[70260.	44890.	57038.	60903.	64765.	37094.	40884.	32191.	40350.
37108.	50494.	50783.	40943.	43000.	59400.	48780.	43454.	38304.
64813.	63795.	31687.	41132.	33750.	26922.	32829.	20890.	17593.
21435.	21744.	21776.	20318.	20089.	17372.	25326.	18822.	39261.
37395.	34134.	40499.	45191.	40637.	32229.	35828.	32224.	30804.
25814.	21548.	22968.	42667.	0.	38477.	30186.	40948.	44252.
45652.	28254.	29791.	30677.	36286.	31431.]			
[66066.	50047.	59676.	54563.	64777.	40483.	37248.	41526.	44598.
41601.	49635.	47115.	41833.	45436.	55345.	45712.	41222.	34870.
55948.	60391.	33983.	44878.	32024.	29495.	34809.	36859.	34964.
35542.	36512.	36396.	31940.	35519.	30234.	35906.	30655.	38818.
35463.	40741.	41680.	42647.	36295.	33359.	37658.	38965.	36681.
34761.	35187.	30497.	44643.	38477.	0.	36975.	46536.	43879.
49915.	44972.	36440.	36420.	36750.	34191.]			
[66128.	40071.	55803.	57010.	62423.	39806.	33720.	36053.	41554.
50044.	51944.	43817.	43339.	44316.	52236.	47454.	54376.	38619.
60415.	65483.	36916.	43368.	33739.	38159.	35471.	40426.	36605.
36484.	39076.	38726.	33239.	34059.	31043.	32235.	33969.	37304.
38924.	35753.	39255.	44426.	41871.	39877.	36048.	33831.	32207.
34788.	31080.	31249.	49172.	30186.	36975.	0.	42788.	43789.
46448.	38415.	33232.	39666.	39812.	37774.]			
[79773.	63919.	62340.	81000.	83536.	55569.	50964.	48867.	61619.

50738.	61602.	51945.	59280.	53428.	65412.	61595.	55522.	48570.
64853.	62268.	42002.	52635.	49425.	39297.	47811.	47980.	44855.
48086.	42661.	43037.	42201.	43791.	46252.	50534.	41919.	52590.
46889.	45805.	53505.	54083.	52721.	43680.	43243.	47967.	45252.
40817.	40834.	41123.	50194.	40948.	46536.	42788.	0.	55458.
60802.	46883.	43932.	46360.	51733.	45815.]			
[67292.	53166.	58145.	70560.	61088.	47894.	49095.	48448.	55980.
51851.	55087.	52038.	43577.	49571.	58683.	57459.	52461.	46111.
61131.	73177.	41949.	55342.	45523.	44931.	48733.	49362.	45833.
49001.	50404.	48401.	44351.	47328.	43712.	41076.	43514.	49456.
51224.	47487.	50498.	52265.	46295.	45462.	47731.	41206.	44766.
43026.	41643.	45347.	60763.	44252.	43879.	43789.	55458.	0.
63805.	49336.	47704.	55833.	50312.	53103.]			
[94774.	61917.	68757.	88214.	85765.	55663.	53753.	51769.	55656.
45562.	65672.	58443.	44937.	67301.	72271.	56151.	61338.	52038.
74982.	73144.	54412.	60325.	50091.	42164.	56342.	48665.	45472.
41193.	45882.	44815.	48015.	45828.	46485.	51576.	48464.	52957.
53195.	51921.	52995.	55471.	49776.	49430.	54020.	57642.	49741.
47243.	51112.	45161.	54172.	45652.	49915.	46448.	60802.	63805.
0.	45643.	53286.	51072.	46024.	50660.]			
[72362.	52429.	61439.	61194.	69417.	36714.	43612.	41442.	42206.
43412.	51328.	49818.	45117.	50171.	60109.	58819.	45855.	45588.
69461.	63539.	38272.	41758.	47017.	40748.	35373.	25148.	24403.
25505.	30389.	26120.	28364.	28108.	22550.	29658.	27880.	48192.
40719.	37070.	47850.	50119.	42704.	37659.	35016.	38274.	37509.
31600.	27997.	35562.	46869.	28254.	44972.	38415.	46883.	49336.
45643.	0.	27301.	32969.	27185.	32882.]			
[65733.	46598.	63122.	63016.	67299.	36693.	41334.	32604.	41048.
38180.	50456.	49704.	47481.	44188.	57935.	52419.	46834.	37676.
63920.	62361.	36015.	45936.	33656.	31472.	33189.	18471.	20287.
22787.	22923.	21393.	26964.	22967.	21752.	28130.	26006.	40315.
43609.	34876.	45784.	42664.	37747.	32498.	39648.	36192.	33353.
28161.	33369.	24753.	39602.	29791.	36440.	33232.	43932.	47704.
53286.	27301.	0.	29465.	29884.	28933.]			
[69525.	47618.	66888.	59538.	70793.	40463.	37469.	35698.	40520.
36751.	51135.	46842.	40305.	45812.	62041.	51260.	50308.	41828.
64599.	58654.	35201.	45776.	34266.	33229.	36661.	27904.	25271.
28051.	25849.	26461.	26964.	29809.	26735.	31985.	25748.	42767.
43619.	38966.	40912.	48330.	43483.	34204.	37579.	38746.	39095.
29325.	34811.	32508.	42169.	30677.	36420.	39666.	46360.	55833.
51072.	32969.	29465.	0.	29882.	37381.]			
[63772.	43745.	62667.	57889.	68705.	41294.	42074.	38430.	47856.
39381.	49911.	56869.	43311.	56466.	64041.	54720.	44474.	42975.
67202.	66595.	44991.	46031.	38860.	39161.	40127.	27718.	26896.
24642.	26952.	32404.	30882.	29367.	28986.	39694.	30983.	46083.
39634.	34258.	49233.	47336.	39939.	34370.	44317.	41560.	44216.
32443.	27273.	34967.	44394.	36286.	36750.	39812.	51733.	50312.
46024.	27185.	29884.	29882.	0.	30760.]			
[69348.	57126.	64530.	72069.	72477.	36370.	40029.	34916.	40680.
42787.	49776.	47072.	39765.	44514.	61374.	55803.	47013.	47975.
60772.	69671.	41157.	43791.	37391.	35445.	35631.	30031.	28945.
30772.	30024.	30717.	29317.	28856.	28229.	34815.	29505.	48793.
48684.	38463.	43595.	46445.	40684.	35547.	38945.	37967.	39237.
30493.	29180.	36037.	44711.	31431.	34191.	37774.	45815.	53103.

```

50660. 32882. 28933. 37381. 30760. 0.]]

In [20]: dtw_train_average = np.ones([12,12])
         for row in range(0,12):
             for column in range(0,12):
                 #print(dtw_array[row*5:(row+1)*5,column*5:(column+1)*5].sum()/20)
                 if row == column:
                     dtw_train_average[row,column] = dtw_array[row*5:(row+1)*5,column*5:(column+1)*5].sum()/20
                 else:
                     dtw_train_average[row,column] = dtw_array[row*5:(row+1)*5,column*5:(column+1)*5].sum()/20
             #print(dtw_train_average)
         dtw_array_refined = dtw_array
         for i in range(0,12):
             for x in range(0,5):
                 dtw_array_refined[i+x,i+x] = dtw_train_average[i,i]
         print(dtw_array[0])

[[ 59137.5  61927.76  63627.52  66007.48  55876.16  64367.12  60231.28
   58732.2  60599.68  60296.72  66272.4  62771.96]
 [ 61927.76  38724.1  47760.68  55491.24  41461.36  40504.  36286.04
   43364.64  41978.76  38806.48  47602.44  39678.52]
 [ 63627.52  47760.68  50086.8  58620.96  44906.04  50506.28  47158.88
   49058.44  47770.2  47380.4  53370.52  50621.64]
 [ 66007.48  55491.24  58620.96  60775.3  48328.72  54324.28  51398.6
   52597.4  51341.32  50931.68  56211.6  54812.84]
 [ 55876.16  41461.36  44907.24  48328.72  31283.4  37101.32  32168.12
   38004.64  37113.08  35067.12  43752.96  38846.16]
 [ 64367.12  40504.  50506.28  54324.28  37101.32  15832.9  19934.2
   40364.24  36689.32  28194.  42689.48  26402.52]
 [ 60231.28  36306.48  47158.88  51398.6  32168.12  19934.2  19222.6
   37213.96  34013.48  26574.24  40553.8  28570.16]
 [ 58732.2  43364.64  49058.44  52597.4  38004.64  40363.68  37213.96
   42403.3  39330.52  38456.08  46614.08  43532.64]
 [ 60599.68  42015.84  47770.2  51317.68  37113.08  36689.32  34013.48
   39330.52  36106.5  36523.92  43428.96  38419.56]
 [ 60296.72  38806.48  47380.4  50931.68  35067.12  28154.04  26574.24
   38456.08  36523.92  31221.8  42093.08  33706.52]
 [ 66272.4  47602.44  53370.52  56211.6  43752.96  42689.48  40553.8
   46614.08  43428.96  42093.08  49039.5  44614.72]
 [ 62771.96  39678.52  50621.64  54812.84  38846.16  26402.52  28570.16
   43532.64  38419.56  33706.52  44614.72  30664.2 ]]

[ 59137.5  57382.  52965.  60451.  67179.  69730.  72022.  58292.
  70434.  61729.  60162.  73598.  60544.  73872.  68264.  63823.
  69766.  67746.  78625.  70195.  56073.  61667.  63817.  59159.
  53302.  71790.  72001.  66396.  69162.  69485.  66841.  64992.
  64891.  61162.  65467.  71017.  55983.  77792.  63066.  60053.
  67497.  69215.  70525.  70038.  70908.  63564.  62430.  60774.
  62301.  70260.  66066.  66128.  79773.  67292.  94774.  72362.
  65733.  69525.  63772.  69348. ]

In [10]: dtw_test = np.empty([488,60])
         for index_data,train_data in enumerate(test_x):
             for index_other,train_other in enumerate(train_x_sorted):
                 distance, path = fastdtw(train_data, train_other, dist=euclidean)
                 dtw_test[index_data,index_other] = distance

```

```

In [11]: dtw_test_averaged = np.ones([488,12])
         for index, test_case in enumerate(dtw_test):
             for classic in range(0,12):
                 dtw_test_averaged[index,classic] = dtw_test[index,classic*5 : (classic+1)*5].sum() / 5

In [67]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import accuracy_score
         class_label = []
         for i in range(1,13):
             class_label.extend([i,i,i,i,i])
         clf = RandomForestClassifier(n_estimators=20,max_features =12)
         clf = clf.fit(dtw_array_refined, class_label)
         pred = clf.predict(dtw_test)
         accuracy_score(test_y,pred)

Out[67]: 0.15778688524590165

In [22]: from sklearn.metrics import accuracy_score
         pred = []
         for test in dtw_test:
             dtw_series = []
             for train in dtw_array:
                 distance, path = fastdtw(train, test, dist=euclidean)
                 dtw_series.append(distance)
             pred.append(int(dtw_series.index(min(dtw_series))/5)+1)
         accuracy_score(test_y,pred)

Out[22]: 0.18237704918032788

In [ ]: print(dtw_test_averaged)

In [ ]: test_index

In [23]: out = open("submission.txt","w")

         for index,y in enumerate(test_y):
             result = 6
             if y == pred[index]:
                 result = 1
             else:
                 result = 0
             out.write(str(test_index[index])+",")
             out.write(str(result))
             out.write("\n")
         out.close()

In [ ]: print("asd")

In [ ]: np.save("dtw-test")

```