

## Source code: 3NN

```
1 %% Read in Dataset
2 data = csvread('homework03.csv');
3 %remove first line (Remove labels)
4 data = data(2:end,:);
5 coords = data(:,1:3)
6 labels = data(:,4)
7
8 xa = [4.1, -0.1, 2.2]
9 xb = [6.1, 0.4, 1.3]
10
11
12 %% Since the axis are not equally derivated we have to do some
    standardization
13
14 %% Doing the Standardization
15 xaStd(1) = (xa(1) - mean(coords(:,1)))./std(coords(:,1));
16 xaStd(2) = (xa(2) - mean(coords(:,2)))./std(coords(:,2));
17 xaStd(3) = (xa(3) - mean(coords(:,3)))./std(coords(:,3));
18
19 xbStd(1) = (xb(1) - mean(coords(:,1)))./std(coords(:,1));
20 xbStd(2) = (xb(2) - mean(coords(:,2)))./std(coords(:,2));
21 xbStd(3) = (xb(3) - mean(coords(:,3)))./std(coords(:,3));
22
23 coordsStd(:,1) = (coords(:,1) - mean(coords(:,1)))./std(coords
    (:,1));
24 coordsStd(:,2) = (coords(:,2) - mean(coords(:,2)))./std(coords
    (:,2));
25 coordsStd(:,3) = (coords(:,3) - mean(coords(:,3)))./std(coords
    (:,3));
26 %%LOOCV Tests
27 %Without Standardization
28 for valid = 1:size(labels,1)
29     dist = [];
30     x = data(valid,1:3);
31     for i = 1:size(labels,1)
32         %Calculating the dist to each point
33         dist = [norm(coords(i)-x),dist];
34     end
35     [~,i] = sort(dist);
36     %Since Training data are unique we get the 3 NN to Validation
        Point by taking the 3 Closest point with dist ~= 0 (i(1)=
        valid)
37     % the second part of the expression returns true iff there is a
        tie
38     out(valid) = (mode(labels(i(2:4))) == labels(valid)) || size(
        unique(labels(i(2:4))),2) == 3;
```

```

39 end
40
41 %With Standardization
42 for valid = 1:size(labels,1)
43     dist = [];
44     x = coords(valid,1:3);
45     for i = 1:size(labels,1)
46         dist = [norm(coordsStd(i,:)-x), dist];
47     end
48     [~,i] = sort(dist);
49     outStd(valid) = (mode(labels(i(2:4))) == labels(valid)) || size
        (unique(labels(i(2:4))),2) == 3;
50 end
51
52 %%%LOOCV Results
53
54 quality = mean(out)
55
56 qualityStd = mean(outStd)
57
58 %% Labeling the Points
59 %Classification without Standardization
60 for i = 1:size(labels,1)
61     dist(i) = norm(coords(i,:)-xa);
62 end
63 [~,i] = sort(dist);
64 va = mode(data(i(1:3),4))
65
66 for i = 1:size(labels,1)
67     dist(i) = norm(coords(i,:)-xb);
68 end
69
70 [~,index] = sort(dist);
71 vb = mode(data(index(1:3),4))
72
73
74
75 %Classification with Standardization
76 for i = 1:size(labels,1)
77     dist(i) = norm(coordsStd(i,:)-xaStd);
78 end
79
80 [~,i] = sort(dist);
81 vaStandardized = mode(labels(i(1:3)))
82
83 for i = 1:size(labels,1)
84     dist(i) = norm(coordsStd(i,:)-xbStd);
85 end
86

```

```

87 [~,i] = sort(dist);
88 vbStandardized = mode(labels(i(1:3)))
89
90
91 %% Regression (just using standardization)
92
93 for i = 1:size(labels,1)
94     dist(i) = norm(coordsStd(i,:) - xaStd);
95 end
96
97 [~,i] = sort(dist);
98 vareg = (1/sum(1./dist(i(1:3)))) * sum ( 1./(dist(i(1:3))) *
99     labels(i(1:3)))
100
101 for i = 1:size(labels,1)
102     dist(i) = norm(coordsStd(i,1:3) - xbStd);
103 end
104
105 [~,i] = sort(dist);
106 vbreg = (1/sum(1./dist(i(1:3)))) * sum ( 1./(dist(i(1:3))) *
107     labels(i(1:3)))

```

## Execution results

```
octave>hw2_4_5
```

```
coords =
```

5.50000	0.50000	4.50000
7.40000	1.10000	3.60000
5.90000	0.20000	3.40000
9.90000	0.10000	0.80000
6.90000	-0.10000	0.60000
6.80000	-0.30000	5.10000
4.10000	0.30000	5.10000
1.30000	-0.20000	1.80000
4.50000	0.40000	2.00000
0.50000	0.00000	2.30000
5.90000	-0.10000	4.40000
9.30000	-0.20000	3.20000
1.00000	0.10000	2.80000
0.40000	0.10000	4.30000
2.70000	-0.50000	4.20000

```
labels =
```

```

2
0

```

```

2
0
2
2
1
1
0
1
0
0
1
1
1

xa =

    4.10000   -0.10000    2.20000

xb =

    6.10000    0.40000    1.30000

quality = 0.13333
qualityStd = 0.26667
va = 0
vb = 2
vaStandardized = 1
vbStandardized = 0
vareg = 1.00000
vbreg = 0.50904

```