

Sep 22, 13 21:24	CS573:Proj1e:Knn_XUE YANG	Page 1/13
1. PROGRAM OUTPUT - For "weather2.csv", show the training dataset, the testing cases, the sorted d isplayed distance, and the determined majority class (k = 5) =====		
Group:1		
5	Training Set	
	outlook, \$temperature, \$humidity, windy, =play, #	
	notes	
	# sunny, 73.14, 85.14, FALSE, yes, #	
	expected	
	# 0.43, 5.24, 10.17, 0.71, 0.71, #	
	certainty	
	sunny, 80, 90, TRUE, no, #	
10	sunny, 69, 70, FALSE, yes, #	
	sunny, 72, 95, FALSE, no, #	
	overcast, 72, 90, TRUE, yes, #	
	rainy, 68, 80, FALSE, yes, #	
	rainy, 70, 96, FALSE, yes, #	
15	overcast, 81, 75, FALSE, yes, #	
>Test Case: sunny, 75, 70,TRUE, yes, 1 0.34 sunny, 80, 90,TRUE, no, 2 0.52 sunny, 69, 70,FALSE, yes, 20 3 0.60 overcast, 72, 90,TRUE, yes, 4 0.65 sunny, 72, 95,FALSE, no, 5 0.73 overcast, 81, 75,FALSE, yes, 6 0.74 rainy, 68, 80,FALSE, yes, 7 0.83 rainy, 70, 96,FALSE, yes, 25 want: yes got: yes		
>Test Case: rainy, 75, 80,FALSE, yes, 1 0.17 rainy, 68, 80,FALSE, yes, 2 0.28 rainy, 70, 96,FALSE, yes, 30 3 0.53 overcast, 81, 75,FALSE, yes, 4 0.54 sunny, 69, 70,FALSE, yes, 5 0.56 sunny, 72, 95,FALSE, no, 6 0.73 overcast, 72, 90,TRUE, yes, 7 0.73 sunny, 80, 90,TRUE, no, 35 want: yes got: yes		
>Test Case: rainy, 71, 91,TRUE, no, 1 0.50 overcast, 72, 90,TRUE, yes, 2 0.51 rainy, 70, 96,FALSE, yes, 40 3 0.54 rainy, 68, 80,FALSE, yes, 4 0.54 sunny, 80, 90,TRUE, no, 5 0.71 sunny, 72, 95,FALSE, no, 6 0.79 sunny, 69, 70,FALSE, yes, 7 0.79 overcast, 81, 75,FALSE, yes, 45 want: no got: yes		
>Test Case: overcast, 83, 86,FALSE, yes, 1 0.18 overcast, 81, 75,FALSE, yes, 2 0.57 overcast, 72, 90,TRUE, yes, 50 3 0.58 sunny, 72, 95,FALSE, no, 4 0.61 rainy, 70, 96,FALSE, yes, 5 0.62 rainy, 68, 80,FALSE, yes, 6 0.65 sunny, 69, 70,FALSE, yes, 7 0.71 sunny, 80, 90,TRUE, no, 55 want: yes got: yes		
>Test Case: sunny, 85, 85,FALSE, no, 1 0.35 sunny, 72, 95,FALSE, no, 2 0.45 sunny, 69, 70,FALSE, yes, 60 3 0.52 sunny, 80, 90,TRUE, no, 4 0.53 overcast, 81, 75,FALSE, yes, 5 0.64 rainy, 70, 96,FALSE, yes,		

Sep 22, 13 21:24	CS573:Proj1e:Knn_XUE YANG	Page 2/13
65	6 0.65 rainy, 68, 80,FALSE, yes, 7 0.78 overcast, 72, 90,TRUE, yes, want: no got: yes	
70	>Test Case: overcast, 64, 65,TRUE, yes, 1 0.45 overcast, 72, 90,TRUE, yes, 2 0.66 overcast, 81, 75,FALSE, yes, 3 0.72 sunny, 69, 70,FALSE, yes, 4 0.75 sunny, 80, 90,TRUE, no, 5 0.75 rainy, 68, 80,FALSE, yes, 6 0.88 rainy, 70, 96,FALSE, yes, 7 0.88 sunny, 72, 95,FALSE, no, want: yes got: yes	
80	>Test Case: rainy, 65, 70,TRUE, no, 1 0.53 rainy, 68, 80,FALSE, yes, 2 0.62 overcast, 72, 90,TRUE, yes, 3 0.66 rainy, 70, 96,FALSE, yes, 4 0.69 sunny, 80, 90,TRUE, no, 5 0.71 sunny, 69, 70,FALSE, yes, 6 0.81 overcast, 81, 75,FALSE, yes, 7 0.83 sunny, 72, 95,FALSE, no, want: no got: yes	
=====		
Group:2		
Training Set		
	outlook, \$temperature, \$humidity, windy, =play, #	
	notes	
90	# rainy, 74.00, 78.14, TRUE, yes, #	
	expected	
	# 0.43, 8.10, 9.86, 0.57, 0.57, #	
	certainty	
	sunny, 75, 70, TRUE, yes, #	
	rainy, 75, 80, FALSE, yes, #	
	rainy, 71, 91, TRUE, no, #	
95	overcast, 83, 86, FALSE, yes, #	
	sunny, 85, 85, FALSE, no, #	
	overcast, 64, 65, TRUE, yes, #	
	rainy, 65, 70, TRUE, no, #	
100	>Test Case: sunny, 80, 90,TRUE, no, 1 0.34 sunny, 75, 70,TRUE, yes, 2 0.52 sunny, 85, 85,FALSE, no, 3 0.54 rainy, 71, 91,TRUE, no, 4 0.69 rainy, 65, 70,TRUE, no, 105 5 0.71 overcast, 83, 86,FALSE, yes, 6 0.73 rainy, 75, 80,FALSE, yes, 7 0.75 overcast, 64, 65,TRUE, yes, want: no got: no	
110	>Test Case: sunny, 69, 70,FALSE, yes, 1 0.45 sunny, 85, 85,FALSE, no, 2 0.52 sunny, 75, 70,TRUE, yes, 3 0.54 rainy, 75, 80,FALSE, yes, 4 0.65 overcast, 83, 86,FALSE, yes, 115 5 0.71 rainy, 65, 70,TRUE, no, 6 0.72 overcast, 64, 65,TRUE, yes, 7 0.79 rainy, 71, 91,TRUE, no, want: yes got: yes	
120	>Test Case: sunny, 72, 95,FALSE, no, 1 0.35 sunny, 85, 85,FALSE, no, 2 0.56 rainy, 75, 80,FALSE, yes, 3 0.58 overcast, 83, 86,FALSE, yes, 4 0.65 sunny, 75, 70,TRUE, yes, 125 5 0.71 rainy, 71, 91,TRUE, no,	

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 3/13

```

6 0.83 rainy, 65, 70,TRUE, no,
7 0.88 overcast, 64, 65,TRUE, yes,
want: no got: yes

130 >Test Case: overcast, 72, 90,TRUE, yes,
1 0.45 overcast, 64, 65,TRUE, yes,
2 0.50 rainy, 71, 91,TRUE, no,
3 0.57 overcast, 83, 86,FALSE, yes,
4 0.60 sunny, 75, 70,TRUE, yes,
135 5 0.62 rainy, 65, 70,TRUE, no,
6 0.73 rainy, 75, 80,FALSE, yes,
7 0.78 sunny, 85, 85,FALSE, no,
want: yes got: yes

140 >Test Case: rainy, 68, 80,FALSE, yes,
1 0.17 rainy, 75, 80,FALSE, yes,
2 0.53 rainy, 65, 70,TRUE, no,
3 0.54 rainy, 71, 91,TRUE, no,
4 0.62 overcast, 83, 86,FALSE, yes,
145 5 0.65 sunny, 85, 85,FALSE, no,
6 0.74 sunny, 75, 70,TRUE, yes,
7 0.75 overcast, 64, 65,TRUE, yes,
want: yes got: no

150 >Test Case: rainy, 70, 96,FALSE, yes,
1 0.28 rainy, 75, 80,FALSE, yes,
2 0.51 rainy, 71, 91,TRUE, no,
3 0.61 overcast, 83, 86,FALSE, yes,
4 0.64 sunny, 85, 85,FALSE, no,
155 5 0.66 rainy, 65, 70,TRUE, no,
6 0.83 sunny, 75, 70,TRUE, yes,
7 0.88 overcast, 64, 65,TRUE, yes,
want: yes got: no

160 >Test Case: overcast, 81, 75,FALSE, yes,
1 0.18 overcast, 83, 86,FALSE, yes,
2 0.53 rainy, 75, 80,FALSE, yes,
3 0.53 sunny, 85, 85,FALSE, no,
4 0.66 overcast, 64, 65,TRUE, yes,
165 5 0.73 sunny, 75, 70,TRUE, yes,
6 0.79 rainy, 71, 91,TRUE, no,
7 0.81 rainy, 65, 70,TRUE, no,
want: yes got: yes
=====
170 Group:3
Training Set
      outlook, $temperature, $humidity, windy, =play, #
notes
# expected rainy, 72.71, 84.43, TRUE, yes, #
# certainty 0.43, 4.68, 11.13, 0.57, 0.57, #
175 sunny, 75, 70, TRUE, yes, #
rainy, 70, 96, FALSE, yes, #
sunny, 80, 90, TRUE, no, #
rainy, 65, 70, TRUE, no, #
overcast, 72, 90, TRUE, yes, #
180 rainy, 75, 80, FALSE, yes, #
sunny, 72, 95, FALSE, no, #

>Test Case: rainy, 71, 91,TRUE, no,
1 0.37 rainy, 65, 70,TRUE, no,
185 2 0.50 overcast, 72, 90,TRUE, yes,
3 0.51 rainy, 70, 96,FALSE, yes,
4 0.54 rainy, 75, 80,FALSE, yes,
5 0.54 sunny, 80, 90,TRUE, no,

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 4/13

```

6 0.61 sunny, 75, 70,TRUE, yes,
190 7 0.71 sunny, 72, 95,FALSE, no,
want: no got: yes

>Test Case: sunny, 69, 70,FALSE, yes,
1 0.41 sunny, 72, 95,FALSE, no,
195 2 0.52 sunny, 75, 70,TRUE, yes,
3 0.54 rainy, 75, 80,FALSE, yes,
4 0.65 sunny, 80, 90,TRUE, no,
5 0.65 rainy, 70, 96,FALSE, yes,
6 0.71 rainy, 65, 70,TRUE, no,
200 7 0.78 overcast, 72, 90,TRUE, yes,
want: yes got: yes

>Test Case: overcast, 64, 65,TRUE, yes,
1 0.45 overcast, 72, 90,TRUE, yes,
205 2 0.51 rainy, 65, 70,TRUE, no,
3 0.57 sunny, 75, 70,TRUE, yes,
4 0.75 sunny, 80, 90,TRUE, no,
5 0.79 rainy, 75, 80,FALSE, yes,
6 0.88 rainy, 70, 96,FALSE, yes,
210 7 0.88 sunny, 72, 95,FALSE, no,
want: yes got: yes

>Test Case: overcast, 81, 75,FALSE, yes,
1 0.53 rainy, 75, 80,FALSE, yes,
215 2 0.60 overcast, 72, 90,TRUE, yes,
3 0.63 sunny, 72, 95,FALSE, no,
4 0.66 rainy, 70, 96,FALSE, yes,
5 0.73 sunny, 75, 70,TRUE, yes,
6 0.75 sunny, 80, 90,TRUE, no,
220 7 0.81 rainy, 65, 70,TRUE, no,
want: yes got: yes

>Test Case: rainy, 68, 80,FALSE, yes,
1 0.17 rainy, 75, 80,FALSE, yes,
225 2 0.26 rainy, 70, 96,FALSE, yes,
3 0.53 rainy, 65, 70,TRUE, no,
4 0.56 sunny, 72, 95,FALSE, no,
5 0.73 overcast, 72, 90,TRUE, yes,
6 0.74 sunny, 75, 70,TRUE, yes,
230 7 0.78 sunny, 80, 90,TRUE, no,
want: yes got: yes

>Test Case: sunny, 85, 85,FALSE, no,
1 0.35 sunny, 72, 95,FALSE, no,
235 2 0.52 sunny, 80, 90,TRUE, no,
3 0.56 rainy, 75, 80,FALSE, yes,
4 0.60 sunny, 75, 70,TRUE, yes,
5 0.64 rainy, 70, 96,FALSE, yes,
6 0.78 overcast, 72, 90,TRUE, yes,
240 7 0.89 rainy, 65, 70,TRUE, no,
want: no got: yes

>Test Case: overcast, 83, 86,FALSE, yes,
1 0.54 rainy, 75, 80,FALSE, yes,
245 2 0.57 overcast, 72, 90,TRUE, yes,
3 0.58 sunny, 72, 95,FALSE, no,
4 0.61 rainy, 70, 96,FALSE, yes,
5 0.71 sunny, 80, 90,TRUE, no,
6 0.78 sunny, 75, 70,TRUE, yes,
250 7 0.87 rainy, 65, 70,TRUE, no,
want: yes got: yes
=====
255 Group:4
Training Set
      outlook, $temperature, $humidity, windy, =play, #
notes
# overcast, 74.43, 78.86, FALSE, yes, #
# expected 0.43, 8.36, 9.34, 0.71, 0.71, #
# certainty rainy, 71, 91, TRUE, no, #

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 5/13

```

sunny, 69, 70, FALSE, yes, #
260 overcast, 64, 65, TRUE, yes, #
overcast, 81, 75, FALSE, yes, #
rainy, 68, 80, FALSE, yes, #
sunny, 85, 85, FALSE, no, #
overcast, 83, 86, FALSE, yes, #

265 >Test Case: sunny, 75, 70,TRUE, yes,
1 0.52 sunny, 69, 70,FALSE, yes,
2 0.57 overcast, 64, 65,TRUE, yes,
3 0.60 sunny, 85, 85,FALSE, no,
270 4 0.61 rainy, 71, 91,TRUE, no,
5 0.73 overcast, 81, 75,FALSE, yes,
6 0.74 rainy, 68, 80,FALSE, yes,
7 0.78 overcast, 83, 86,FALSE, yes,
want: yes got: yes

275 >Test Case: rainy, 70, 96,FALSE, yes,
1 0.26 rainy, 68, 80,FALSE, yes,
2 0.51 rainy, 71, 91,TRUE, no,
3 0.61 overcast, 83, 86,FALSE, yes,
280 4 0.64 sunny, 85, 85,FALSE, no,
5 0.65 sunny, 69, 70,FALSE, yes,
6 0.66 overcast, 81, 75,FALSE, yes,
7 0.88 overcast, 64, 65,TRUE, yes,
want: yes got: yes

285 >Test Case: sunny, 80, 90,TRUE, no,
1 0.52 sunny, 85, 85,FALSE, no,
2 0.54 rainy, 71, 91,TRUE, no,
3 0.65 sunny, 69, 70,FALSE, yes,
290 4 0.71 overcast, 83, 86,FALSE, yes,
5 0.75 overcast, 64, 65,TRUE, yes,
6 0.75 overcast, 81, 75,FALSE, yes,
7 0.78 rainy, 68, 80,FALSE, yes,
want: no got: yes

295 >Test Case: rainy, 65, 70,TRUE, no,
1 0.37 rainy, 71, 91,TRUE, no,
2 0.51 overcast, 64, 65,TRUE, yes,
3 0.53 rainy, 68, 80,FALSE, yes,
300 4 0.71 sunny, 69, 70,FALSE, yes,
5 0.81 overcast, 81, 75,FALSE, yes,
6 0.87 overcast, 83, 86,FALSE, yes,
7 0.89 sunny, 85, 85,FALSE, no,
want: no got: yes

305 >Test Case: overcast, 72, 90,TRUE, yes,
1 0.45 overcast, 64, 65,TRUE, yes,
2 0.50 rainy, 71, 91,TRUE, no,
3 0.57 overcast, 83, 86,FALSE, yes,
310 4 0.60 overcast, 81, 75,FALSE, yes,
5 0.73 rainy, 68, 80,FALSE, yes,
6 0.78 sunny, 85, 85,FALSE, no,
7 0.78 sunny, 69, 70,FALSE, yes,
want: yes got: yes

315 >Test Case: rainy, 75, 80,FALSE, yes,
1 0.17 rainy, 68, 80,FALSE, yes,
2 0.53 overcast, 81, 75,FALSE, yes,
3 0.54 rainy, 71, 91,TRUE, no,
320 4 0.54 overcast, 83, 86,FALSE, yes,
5 0.54 sunny, 69, 70,FALSE, yes,
6 0.56 sunny, 85, 85,FALSE, no,
7 0.79 overcast, 64, 65,TRUE, yes,
want: yes got: yes

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 6/13

```

325 >Test Case: sunny, 72, 95,FALSE, no,
1 0.35 sunny, 85, 85,FALSE, no,
2 0.41 sunny, 69, 70,FALSE, yes,
3 0.56 rainy, 68, 80,FALSE, yes,
330 4 0.58 overcast, 83, 86,FALSE, yes,
5 0.63 overcast, 81, 75,FALSE, yes,
6 0.71 rainy, 71, 91,TRUE, no,
7 0.88 overcast, 64, 65,TRUE, yes,
want: no got: yes
335 knn_acc = ['57.14', '57.14', '71.43', '57.14']

- Show 2*2 xval results for k=5nn and nb for soybean, diabetes.
1). Diabetes
> 5nn : ['74.74', '69.27', '72.14', '72.66']
> nb : ['76.30', '72.92', '76.82', '74.48']
2). Soybean
> 5nn : ['86.22', '87.10', '89.44', '87.68']
> nb : ['86.51', '90.62', '91.50', '83.87']

345 2. SOURCE CODE
*****
File <dist.py>

350 import lib

def dist(this, that, lst, table):
    sum = 0.0
    for k in lst:
        v1 = this[k]
        v2 = that[k]
        if v1 == '?' and v2 == '?': sum += 1
        elif k in table.nump:
            i = table.nump.index(k)
            #print table.lo[i], table.hi[i]
            aLittle = 10**-7
            if v1 == '?': v1 = 1.0 if v2 < 0.5 else 0.0
            else:
                v1 = (float(v1) - float(table.lo[i]))/(float(table.hi[i]) - floa
t(table.lo[i]) + aLittle)
            if v2 == '?': v2 = 1.0 if v1 < 0.5 else 0.0
            else:
                v2 = (float(v2) - float(table.lo[i]))/(float(table.hi[i]) - floa
t(table.lo[i]) + aLittle)
            sum += (float(v2) - float(v1))**2
        else:
            if v1 == '?': sum += 1.0
            elif v2 == '?': sum += 1.0
            elif v1 != v2: sum +=1.0
            else: sum += 0.0
    return sum**0.5/len(lst)**0.5

375 def closest(i, table):
    minval = lib.INF
    this = [table.data[i][s] for s in range(len(table.data))]
    for j in range(len(table.data[0])):
        if i == j: continue
        that = [table.data[j][s] for s in range(len(table.data))]
        d = dist(this, that, table.indep, table)
        if d < minval: minval = d; out = j
    return out

385 def furthest(i, table):
    maxval = lib.NINF
    this = [table.data[i][s] for s in range(len(table.data))]
    for j in range(len(table.data[0])):
        if i == j: continue
        that = [table.data[j][s] for s in range(len(table.data))]
        d = dist(this, that, table.indep, table)
        if d > maxval: maxval = d; out = j
    return out

395 *****

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 7/13

File <knn.py>

```

import reader
import dist
400 import xval
import lib
import tablestr

def knn(testT, trainT, tables, kn):
405     tablestr.tableprint(trainT['0'])
    k = testT['0'].klass[0]
    acc = 0.0
    for d in range(len(testT['0'].data[0])):
        want = testT['0'].data[k][d]
410         row = [testT['0'].data[s][d] for s in range(len(testT['0'].data))]

        got = knn1(row, trainT, tables, kn)
        acc += want == got
        print 'want:', want, ' got:', got
    return 100 * float(acc/len(testT['0'].data[0]))

415 def knn1(this, trainT, tables, kn):
    kmax = neighbors(this, trainT, tables)
    #print '\n>Test Case: ' + lib.rowprint(this, 5)
    for k in range(len(kmax)):
420         that = [trainT['0'].data[s][int(kmax[k][0])] for s in range(len(trainT['
0'].data))]
        print k+1, tables['0'].CONVFMT%float(kmax[k][1]), lib.rowprint(that, 5)
        seen = nearestk(this, kn, trainT, kmax)
        return mostSeen(seen)

425 def neighbors(this, trainT, tables):
    lst = {}
    for d in range(len(trainT['0'].data[0])):
        that = [trainT['0'].data[s][d] for s in range(len(trainT['0'].data))]
        lst[d] = dist.dist(this, that, range(len(trainT['0'].data)-1), tables['0
'])
430     return sorted(lst.items(), key=lambda d:d[1])

def nearestk(this, kn, trainT, lst):
    k = trainT['0'].klass[0] # get index for class
    seen = {}
435     for i in range(kn):
        that = int(lst[i][0])
        got = trainT['0'].data[k][that]
        if got in seen.keys(): seen[got] += 1
        else: seen[got] = 0
440     return seen

def mostSeen(seen):
    maxval = 0
    out = seen.keys()[0]
445     for s in seen.keys():
        if seen[s] > maxval: maxval = seen[s]; out = s
    return out
*****
File <uxval.py>
450
import lib
import reader
import tablestr
import random

455 def uxvals(tables, x, b):
    rows = lib.indexes(tables['0'].data[0])
    s = int(len(rows)/b)
    uxvaltables = {}
460     for i in range(x): # x times
        random.shuffle(rows)
        for bl in range(b): # b bins
            obj = uxval(bl*s, (bl+1)*s, rows, tables)
            uxvaltables[i*x+bl+1] = obj
465     return uxvaltables

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 8/13

```

def uxval(start, stop, rows, tables):
    testT = tablestr.Table()
470     trainT = tablestr.Table()
    reader.makeTable(tables['0'].name, testT)
    reader.makeTable(tables['0'].name, trainT)
    for r in range(len(rows)):
        d = rows[r]
475         a = []
        for j in range(len(tables['0'].order)):
            a.append(tables['0'].data[j][d])
            if r >= start and r < stop: #belonging to testing data set
                reader.addRow(a, testT)
480         else:
            reader.addRow(a, trainT)
    test = {}
    train = {}
    test['0'] = testT
485     train['0'] = trainT
    tables = {}
    tables['train'] = train
    tables['test'] = test
    return tables
490 *****
File <lib.py>

import math
inf = 10^17
495 NINF = -1 * inf
PINCH = 1 / inf
PI = 3.1415926535
EE = 2.7182818284

500 def indexes(data):
    rows = [] #get the indexes for the data
    for i in range(len(data)):
        rows.append(i)
    return rows

505 def rowprint(a):
    max = len(a)
    line = ''
    for j in range(max):
510         line += (a[j] + ',').rjust(15)
    return line

def maybeInt(x):
    return int(x) if x % 1 == 0.0 else float(x)

515 def norm(x, m, s):
    s += PINCH
    return 1/math.sqrt(2*PI*s**2.0)*EE**(-1*(x-m)**2.0/(2*s**2.0))
*****
520 File <nb.py>

import lib

def nb(testT, trainT, hypotheses, k, m):
525     ck = testT['0'].klass[0] #locate the index for class col
    total = acc = 0.0
    total += len(trainT['0'].data[ck])
    for t in range(len(testT['0'].data[ck])):
        want = testT['0'].data[ck][t]
530         row = []
        for i in range(len(testT['0'].data)):
            row += [testT['0'].data[i][t]]
        got = likelihood(row, trainT, total, hypotheses, k, m)
        acc += want == got
535     return 100 * acc/len(testT['0'].data[ck])

def likelihood(row, trainT, total, hypotheses, k, m):
    like = lib.NINF

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 9/13

```

540     total += k * len(hypotheses)
    best = ''
    for h in hypotheses:
        nh = len(trainT[h].data[trainT[h].klass[0]])
        prior = float(nh + k)/total
        tmp = prior
545     for c in trainT[h].nump:
        i = trainT[h].nump.index(c)
        x = row[c]
        if x == "?": continue
        y = lib.norm(float(x), float(trainT[h].mu[i]), float(trainT[h].sd[i]
    ))
550     tmp *= y
    for c in trainT[h].term:
        x = row[c]
        if x == "?": continue
        y = 0.0
555     for i in range(len(trainT[h].data[c])):
        if trainT[h].data[c][i] == x: y += 1
        tmp *= (y + m*prior) / (nh + m)
    if tmp >= like:
        like = tmp
        best = h
560     return best
    *****
File <reader.py>

565 import re
import tablestr
def readcsv(filename, table):
    "read in data from csv and create a table"
    FS = ',' #define field separator
570     f = open(filename)
    seen = 0
    while True:
        str = line(f)
        if str == -1:
            if seen == 0: print("WARNING: empty or missing file")
            return -1
575         a = str.split(FS) #compute the number of attributes in table
        if len(a) > 1:
            if seen: addRow(a, table)
            else: makeTable(a, table)
            seen += 1

580     def line(f):
        "get one line data (without comments and whitespace)"
        str = f.readline()
        if not str: return -1 #readline finds nothing, output error
        else:
            str = "".join(str.split()) #kill whitespace
            str = re.sub(r'#[*',' ',str) #kill comments
590         if len(str) >= 1 and str[-1] == ',': return str + line(f)
        else: return str

    def makeTable(a, table):
        "read table titles and set all corresponding parameters"
595         c = 0
        for ite in range(len(a)):
            if a[ite][0] == '?': continue #the col with '?' is ignored
            table.order.append(ite)
            x = a[ite]
600             table.name.append(x)
            isNum = 1
            if x.find('=') != -1:
                table.dep.append(c)
                table.klass.append(c)
605             isNum = 0
            elif x.find('+') != -1:
                table.dep.append(c)
                table.more.append(c)
            elif x.find('-') != -1:
                table.dep.append(c)
610

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 10/13

```

        table.less.append(c)
        elif x.find('$') != -1:
            table.indep.append(c)
            table.num.append(c)
615        else:
            table.indep.append(c)
            table.term.append(c)
            isNum = 0
            table.n.append('0')
620        if isNum:
            table.nump.append(c)
            table.hi.append(-1.0*10**32)
            table.lo.append(10.0**32)
            table.mu.append(0)
625            table.m2.append(0)
            table.sd.append(0)
        else:
            table.wordp.append(c)
            table.most.append(0)
            table.count.append({})
            table.mode.append('')
630        c += 1
        for i in range(c): table.data.append([])

635 def addRow(a, table):
    "add a row of data to the table"
    for c in range(len(table.name)):
        f = table.order[c]
        x = a[f]
640        table.data[c].append(x)
        if x.find('?') == -1:
            table.n[c] = int(table.n[c]) + 1
            if c in table.wordp:
                k = table.wordp.index(c)
645                if table.count[k].has_key(x): table.count[k][x] += 1
                else: table.count[k][x] = 1
                new = table.count[k][x]
                if new > table.most[k]:
                    table.mode[k] = x
                    table.most[k] = new
650            else:
                k = table.nump.index(c)
                if float(x) > float(table.hi[k]): table.hi[k] = x
                if float(x) < float(table.lo[k]): table.lo[k] = x
                delta = float(x) - table.mu[k]
655                table.mu[k] += delta/table.n[c]
                table.m2[k] += delta*(float(x) - table.mu[k])
                if table.n[c] > 1:
                    table.sd[k] = (table.m2[k]/(table.n[c] - 1))**0.5
660                c += 1

    def classes(table):
        "generate a set of tables based on different classes"
        if len(table.klass) == 0:
            print "No labeled classes in the given data set"
            return -1
        # assume there is only one class feature in the data set
        data = table.data[table.klass[0]]
        classnames = []
670        for s in data:
            if s not in classnames:
                classnames.append(s)
            tables = klass1(table, classnames, data)
            tables['0'] = table
675            tables['names'] = classnames
            return tables

    def klass1(table, classnames, data):
        tables = {}
680        for s in classnames:
            tables[s] = tablestr.Table()
            makeTable(table.name, tables[s])
            for i in range(len(data)):

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 11/13

```

        if s == data[i]:
            a = []
            for j in range(len(table.order)):
                a.append(table.data[j][i])
            addRow(a, tables[s])
        return tables
690 *****
File <tablestr.py>

import lib
class Table:
695     def __init__(self):
        self.data = []      #data[[col1,...],[col2,...]]
        self.name = []      #name of i-th column
        self.order = []     #order of the col
        self.nump = []      #is i-th column numeric?
        self.wordp = []     #is i-th column non-numeric?
        self.indep = []     #list of indep columns
        self.dep = []       #list of dep columns
        self.less = []      #numeric goal to be minimized
        self.more = []      #numeric goal to be maximized
        self.klass = []     #non-numeric goal
        self.term = []      #non-numeric non-goal
        self.num = []       #numeric non-goal
        # for all cols
        self.n = []         #count of things in this col
        # for wordp columns:
        self.count = []     #count of each word
        self.mode = []      #most common word
        self.most = []      #count of most common word
        # for nump columns:
        self.hi = []        #upper bound
        self.lo = []        #lower bound
        self.mu = []        #mean
        self.m2 = []        #sum of all nums
        self.sd = []        #standard deviation# -*- coding: utf-8 -*-
        # table printing format
        self.CONVFMT = '%4.2f'

    def centroid(table):
        "update the mode and most values for wordp type cols or update the mean and
        sd values for nump cols"
        rows = [[]]
        for c in range(len(table.name)):
            s = table.mode[table.wordp.index(c)] if c in table.wordp else table.CONV
            FMT%table.mu[table.nump.index(c)]
            rows[0].append(str(s))
            if table.n[c] == '0':
                s = 0.0
            else:
                s = float(table.most[table.wordp.index(c)]/table.n[c] if c in table
                .wordp else table.sd[table.nump.index(c)]
            rows[1].append(str(table.CONVFMT%s))
        return rows

735     def tableprint(table, stats=''):
        "print table on the console"
        print ' '
        if stats != '': table.CONVFMT = stats
        print(' ' + lib.rowprint(table.name)+ ' ' # notes'.ljust(10))
        print('#' + lib.rowprint(centroid(table)[0]) + ' ' # expected'.ljust(10))
        print('#' + lib.rowprint(centroid(table)[1]) + ' ' # certainty'.ljust(10))

        for j in range(len(table.data[0])):
            line = []
        740         for i in range(len(table.data)):
            line.append(table.data[i][j])
        745         print(' ' + lib.rowprint(line)+ ' ' #'.ljust(10))

    def tableprint_txt(table, f, stats=''):
        "print table on the indicated txt file with table name"
        f.write('\n')
        #f.write('\n' +tablename + '\n'*2)

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 12/13

```

        if stats != '': table.CONVFMT = stats
        f.write(' ' + lib.rowprint(table.name)+ ' ' # notes'.ljust(10) + '\n')
        f.write('#' + lib.rowprint(centroid(table)[0]) + ' ' # expected'.ljust(10) +
        '\n')
        f.write('#' + lib.rowprint(centroid(table)[1]) + ' ' # certainty'.ljust(10) +
        '\n')
        for j in range(len(table.data[0])):
            line = []
            for i in range(len(table.data)):
                line.append(table.data[i][j])
        760         f.write(' ' + lib.rowprint(line)+ ' ' #'.ljust(10) + '\n')
        *****
        File <xval.py>

        import lib
        import tablestr
        import reader
        import random

        770     def xvals(tables, x, b):
            k = tables['0'].order.index(tables['0'].klass[0])
            rows = lib.indexes(tables['0'].data[k])
            s = int(len(rows)/b)
            xvaltables = {}
        775         for i in range(x): # x times
            random.shuffle(rows)
            for b1 in range(b): # b bins
                obj = xval(b1*s, (b1+1)*s, rows, tables)
                xvaltables[i*x+b1+1] = obj
        780         return xvaltables

    def xval(start, stop, rows, tables):
        testT = tablestr.Table()
        trainT = tablestr.Table()
        reader.makeTable(tables['0'].name, testT)
        reader.makeTable(tables['0'].name, trainT)
        for r in range(len(rows)):
            d = rows[r]
            a = []
        790             for j in range(len(tables['0'].order)):
                a.append(tables['0'].data[j][d])
            if r >= start and r < stop: #belonging to testing data set
                reader.addRow(a, testT)
        795             else:
                reader.addRow(a, trainT)
            testT = reader.klasses(testT)
            trainT = reader.klasses(trainT)
            tables = {}
            tables['train'] = trainT
            tables['test'] = testT
            return tables
        *****
        File <main.py>

        805     import reader
        import tablestr
        import zeror
        import xval
        810     import nb
        import knn

        if __name__ == "__main__":
            filename = 'data/diabetes.csv'
            table = tablestr.Table()
            reader.readcsv(filename, table) #create raw data structure
            f = '%4.2f' #read the .csv data set
            tables = reader.klasses(table) #set the formatting for the output
            b = x = 2
            kn = 5
            k = 1
            m = 2
            xvaltables = xval.xvals(tables, x, b)

```

Sep 22, 13 21:24

CS573:Proj1e:Knn_XUE YANG

Page 13/13

```
825     knn_acc = []
        nb_acc = []
        for s in range(b*x):
            s += 1
            acc = knn.knn(xvaltables[s]['test'], xvaltables[s]['train'], tables, kn)
            acc2 = nb.nb(xvaltables[s]['test'], xvaltables[s]['train'], tables['name
s'], k, m)
830         knn_acc += [f%acc]
            nb_acc += [f%acc2]
        print 'knn_acc =', knn_acc
        print 'nb_acc =', nb_acc

835
```