Proposed solution for Ex 8

**Q1: Ensembles**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Math** | **English** | **Civics** | **Science** | **PE** | **History** | **Status** |
| Tom | 6 | 6 | 6 | 6 | 6 | 6 | Excellent |
| Peter | 1 | 1 | 1 | 1 | 1 | 1 | Poor |
| Jane | 3 | 6 | 4 | 4 | 4 | 4 | Good |
| Jack | 6 | 2 | 2 | 5 | 3 | 3 | Good |
| Mary | 4 | 4 | 5 | 4 | 3 | 5 | Good |
| Phyllis | 4 | 2 | 2 | 6 | 2 | 3 | Good |
| Ron | 2 | 4 | 3 | 2 | 1 | 2 | Poor |
| Diane | 5 | 4 | 6 | 6 | 4 | 6 | Excellent |
| Fiona | 5 | 5 | 5 | 5 | 3 | 5 | Excellent |
| Rodger | 2 | 2 | 2 | 3 | 2 | 1 | Poor |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Joy | 3 | 2 | 2 | 3 | 4 | 1 | ?? |
| Bob | 6 | 1 | 1 | 1 | 1 | 1 | ?? |
| Sam | 2 | 3 | 2 | 3 | 6 | 4 | ?? |

The upper table shows how students are graded in different subjects and what overall status that they are given on their final score card. This is your training set. The table below shows a non-labelled set of records that the computer is going to categorize.

1. Use KNN to train a system to help the categorization. Use K=3. Test the result for Joy, Bob and Sam.
2. Demonstrate bagging with KNN and use a committee approach to determine the result for Joy, Bob and Sam
3. Repeat the procedure in 1 above, but use subspace modelling and committee approach.
4. Compare the result in 2 and 3 with the outcome of 1.

You can do this part by hand, with Excel or Python.

#1 By excel

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Name** | **Math** | **English** | **Civics** | **Science** | **PE** | **History** | **Status** |  |  |
|  | Tom | 6 | 6 | 6 | 6 | 6 | 6 | Excellent |  |  |
|  | Peter | 1 | 1 | 1 | 1 | 1 | 1 | Poor |  |  |
|  | Jane | 3 | 6 | 4 | 4 | 4 | 4 | Good |  |  |
|  | Jack | 6 | 2 | 2 | 5 | 3 | 3 | Good |  |  |
|  | Mary | 4 | 4 | 5 | 4 | 3 | 5 | Good |  |  |
|  | Phyllis | 4 | 2 | 2 | 6 | 2 | 3 | Good |  |  |
|  | Ron | 2 | 4 | 3 | 2 | 1 | 2 | Poor |  |  |
|  | Diane | 5 | 4 | 6 | 6 | 4 | 6 | Excellent |  |  |
|  | Fiona | 5 | 5 | 5 | 5 | 3 | 5 | Excellent |  |  |
|  | Rodger | 2 | 2 | 2 | 3 | 2 | 1 | Poor |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Joy | 3 | 2 | 2 | 3 | 4 | 1 |  |  |  |
|  | Bob | 6 | 1 | 1 | 1 | 1 | 1 |  |  |  |
|  | Sam | 2 | 3 | 2 | 3 | 6 | 4 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | K=3 |  |  |  |  |  |  |  |  |  |
|  | Joy vs |  |  |  |  |  |  | SUM | Rank |  |
|  | Tom | 9 | 16 | 16 | 9 | 4 | 25 | 79 |  |  |
|  | Peter | 4 | 1 | 1 | 4 | 9 | 0 | 19 |  |  |
|  | Jane | 0 | 16 | 4 | 1 | 0 | 9 | 30 |  |  |
|  | Jack | 9 | 0 | 0 | 4 | 1 | 4 | 18 | 3 |  |
|  | Mary | 1 | 4 | 9 | 1 | 1 | 16 | 32 |  |  |
|  | Phyllis | 1 | 0 | 0 | 9 | 4 | 4 | 18 | 3 | Good |
|  | Ron | 1 | 4 | 1 | 1 | 9 | 1 | 17 | 2 | Poor |
|  | Diane | 4 | 4 | 16 | 9 | 0 | 25 | 58 |  |  |
|  | Fiona | 4 | 9 | 9 | 4 | 1 | 16 | 43 |  |  |
|  | Rodger | 1 | 0 | 0 | 0 | 4 | 0 | 5 | 1 | Poor |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Joy has status Poor by simple majority | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Bob vs |  |  |  |  |  |  | SUM | Rank |  |
|  | Tom | 0 | 25 | 25 | 25 | 25 | 25 | 125 |  |  |
|  | Peter | 25 | 0 | 0 | 0 | 0 | 0 | 25 | 2 | Poor |
|  | Jane | 9 | 25 | 9 | 9 | 9 | 9 | 70 |  |  |
|  | Jack | 0 | 1 | 1 | 16 | 4 | 4 | 26 | 3 | Good |
|  | Mary | 4 | 9 | 16 | 9 | 4 | 16 | 58 |  |  |
|  | Phyllis | 4 | 1 | 1 | 25 | 1 | 4 | 36 |  |  |
|  | Ron | 16 | 9 | 4 | 1 | 0 | 1 | 31 |  |  |
|  | Diane | 1 | 9 | 25 | 25 | 9 | 25 | 94 |  |  |
|  | Fiona | 1 | 16 | 16 | 16 | 4 | 16 | 69 |  |  |
|  | Rodger | 16 | 1 | 1 | 4 | 1 | 0 | 23 | 1 | Poor |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Bob has status poor by simple majority | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Sam vs |  |  |  |  |  |  | SUM | Rank |  |
|  | Tom | 16 | 9 | 16 | 9 | 0 | 4 | 54 |  |  |
|  | Peter | 1 | 4 | 1 | 4 | 25 | 9 | 44 |  |  |
|  | Jane | 1 | 9 | 4 | 1 | 4 | 0 | 19 | 1 | Good |
|  | Jack | 16 | 1 | 0 | 4 | 9 | 1 | 31 |  |  |
|  | Mary | 4 | 1 | 9 | 1 | 9 | 1 | 25 | 2 | Good |
|  | Phyllis | 4 | 1 | 0 | 9 | 16 | 1 | 31 |  |  |
|  | Ron | 0 | 1 | 1 | 1 | 25 | 4 | 32 |  |  |
|  | Diane | 9 | 1 | 16 | 9 | 4 | 4 | 43 |  |  |
|  | Fiona | 9 | 4 | 9 | 4 | 9 | 1 | 36 |  |  |
|  | Rodger | 0 | 1 | 0 | 0 | 16 | 9 | 26 | 3 | Poor |
|  |  |  |  |  |  |  |  |  |  |  |
|  | Sam is good by simple majority | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

Demonstrate bagging with KNN and use a committee approach to determine the result for Joy, Bob and Sam

A subset of records are picked randomly, analyzed and replaced. With a committee of 3 we repeat this 3 times. With a committee of 300 we do it 300 times. Here we apply 3 for illustration:

Committee 1= subset 1: 2,3,4,5, which implies the data for Peter, Jane, Jack, Mary

Committee 2 = subset 2: 1,3,8,10, which implies the data for Tom, Jane, Diane, Rodger

Committee 3 = subset 3: 1,4,5,6, which implies the data for Tom, Jack, Mary, Phyllis

K=3 for each candidate and committee

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Joy |  |  | Bob |  |  | Sam |  |  |
|  |  | Score | Rank | Status | Score | Rank | Status | Score | Rank | Status |
| Com1 | Peter | 19 | 2 | Poor | 25 | 1 | Poor | 44 |  |  |
|  | Jane | 30 | 3 | Good | 70 |  |  | 19 | 1 | Good |
|  | Jack | 18 | 1 | Good | 26 | 2 | Good | 31 | 3 | Good |
|  | Mary | 32 |  |  | 58 | 3 | Good | 25 | 2 | Good |
| Com2 | Tom | 79 |  |  | 125 |  |  | 54 |  |  |
|  | Jane | 30 | 2 | Good | 70 | 2 | Good | 19 | 1 | Good |
|  | Diane | 58 | 3 | Excel | 94 | 3 | Excel | 43 | 3 | Excel |
|  | Rodger | 5 | 1 | Poor | 23 | 1 | Poor | 26 | 2 | Poor |
| Com3 | Tom | 79 |  |  | 125 |  |  | 54 |  |  |
|  | Jack | 18 | 2 | Good | 26 | 1 | Good | 31 | 2 | Good |
|  | Mary | 32 | 3 | Good | 58 | 3 | Good | 25 | 1 | Good |
|  | Phyllis | 18 | 1 | Good | 36 | 2 | Good | 31 | 3 | Good |
|  |  |  |  |  |  |  |  |  |  |  |
| P(Poor): |  | 2 | 9 | 0,222 | 2 | 9 | 0,222 | 1 | 9 | 0,111 |
| P(Good): |  | 6 | 9 | 0,667 | 6 | 9 | 0,667 | 7 | 9 | 0,778 |
| P(Excellent): |  | 1 | 9 | 0,111 | 1 | 9 | 0,111 | 1 | 9 | 0,111 |
| Majority rule says |  |  | Good |  |  | Good |  |  | Good |  |
|  |  |  |  |  |  |  |  |  |  |  |

Repeat the procedure in 1 above, but use subspace modelling and committee approach.



**Q2 Assume that the numbers «1» and «2» can be encoded in binary form as shown below in a 9x9 matrix:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1:  0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| The number 2 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

You have two kernels:

A and B

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 |  |  | 0 | 1 | 0 |
| 1 | 1 | 1 |  |  | 0 | 1 | 0 |
| 0 | 0 | 0 |  |  | 0 | 1 | 0 |

Write small Python program that demonstrates convolution with stride 1 for both kernels. Present the resulting matrices.

The convolution principle demonstrated



Example code

Created on Fri May 11 13:33:48 2018

@author: Bernt

"""

import numpy as np

import math

import matplotlib.pyplot as plt

one = [[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1,0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0]]

two = [[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 1, 1, 1, 1, 1, 0, 0, 0],

[0, 1, 1, 0, 0, 1, 1, 0, 0],

[0, 1, 1, 0, 0, 1, 1, 0, 0],

[0, 0, 0, 0, 1, 1, 1, 0, 0],

[0, 0, 0, 1, 1, 0, 0, 0, 0],

[0, 0, 1, 1, 0, 0, 0, 0, 0],

[0, 1, 1, 1, 1, 1, 1, 0, 0],

[0, 1, 1, 1, 1, 1, 1, 0, 0]]

kernel1 = [[0,1,0], [0,1,0],[0,1,0]]

kernel2 = [[1,1,1], [0,0,0],[0,0,0]]

def convolve(submatrix, kernel):

j= 0

allS = 0

while j < len(kernel):

vect1 = submatrix[j]

vect2 = kernel[j]

i = 0

partS = 0

#print('convolve', vect1, vect2)

while i < len(vect2):

partS = partS + vect1[i]\*vect2[i]

#print(vect1[i], vect2[i])

i = i + 1

allS = allS + partS

j = j + 1

return allS

def createSubMatrix (matrix, y1,y2,xVal):

return(matrix[y1:y2,xVal])

def stride (stride, vector, kernel):

ky = len(vector)

kx = len(kernel)

matrix = np.array(vector)

conMatrix = []

y = 0

while y < ky-kx:

y2 = y + kx

x = 0

eM = []

while x < ky-kx+1:

k = 0

strider = []

while k < kx:

strider.append(k+x)

#print(strider)

k = k +1

subMatrix = createSubMatrix(matrix,y,y2,strider)

#print('Strider', strider, 'y', y, 'y2', y2, '----->', subMatrix)

x = x +1

n = convolve(subMatrix,kernel)

#print('reduced', subMatrix, 'new result' ,n)

eM.append(n)

y = y + stride

print ('New convoluted matrix', eM)

conMatrix.append(eM)

return conMatrix

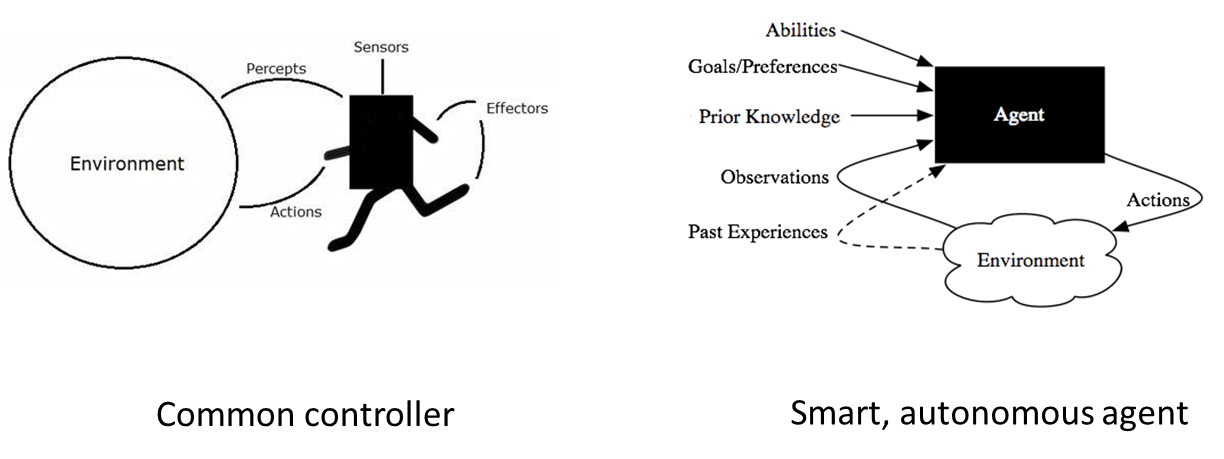
**Q3 Agents**

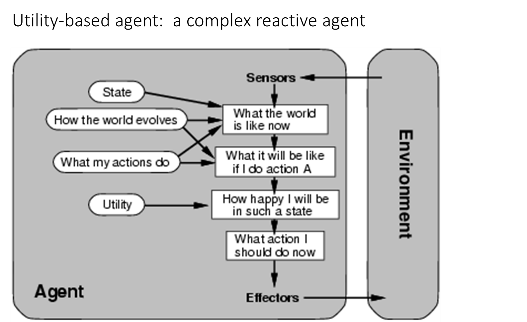
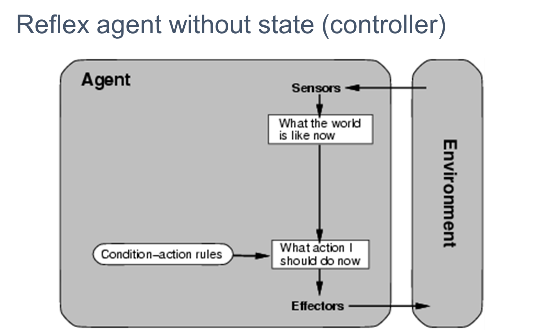
1. Illustrate by a drawing what the difference between a simple controller and a sophisticated, high level reactive agent is.
2. Use the drawing in a. above and illustrate the difference between a mediated MAS, peer-to-peer MAS and a colony.

**Q3 Agents**

1. Illustrate by a drawing what the difference between a simple controller and a sophisticated, high level reactive agent is.
2. Use the drawing in a. above and illustrate the difference between a mediated MAS, peer-to-peer MAS and a colony.

Proposed answer (from lecture):





B:

