**Задание**

Построить дерево принятия решений с использованием критерия Джини

| **X1** | **X2** | **X3** | **Y** |
| --- | --- | --- | --- |
| 12,4 | -3,8 | 11,2 | 0 |
| 0 | 99 | 78,4 | 0 |
| 65 | 4,1 | 14 | 1 |
| -5 | -11,2 | 8 | 0 |
| 3,4 | 80 | 1 | 1 |
| 2,2 | 18,1 | 12 | 0 |
| 0 | 15 | 47,1 | 0 |
| 1000 | -1000 | 1 | 0 |
| 45 | 46 | -7,1 | 1 |

Определить результат классификации для следующих данных:

| **X1** | **X2** | **X3** |
| --- | --- | --- |
| 14 | -80 | 1,12 |
| -4 | 100 | -7,4 |
| 80,4 | -44,7 | 1 |
| 64,2 | 102,4 | 14 |

**Решение**

Следуя жадному алгоритму, мы должны выбирать локально оптимальные решения. То есть необходимо вычислить критерий Джини каждого разбиения. С этой целью была написана программа на языке Python, реализующая формулу критерия Джини при разбиении на две части.

N — число примеров в узле — предке, L, R — число примеров соответственно в левом и правом потомке, li и ri — число экземпляров i-го класса в левом/правом потомке, n – количество классов.

В нашей задаче N = 9, n = 2.

В результате работы программы был получен массив критериев, соответствующий

разбиению xj ≥ const.

[[3.60, 3.75, 3.86, 100, **2.40**, 3.0, 3.75, 3.75, 3.0],

[3.43, 3.75, 3.0, 3.75, 3.86, 3.60, 3.90, 100, 3.0],

[3.60, 3.75, 4.0, 3.0, 3.0, 3.90, 3.43, 3.0, 100]]

Для:

[[12.4, 0, 65, -5, **3.4**, 2.2, 0, 1000, 45],

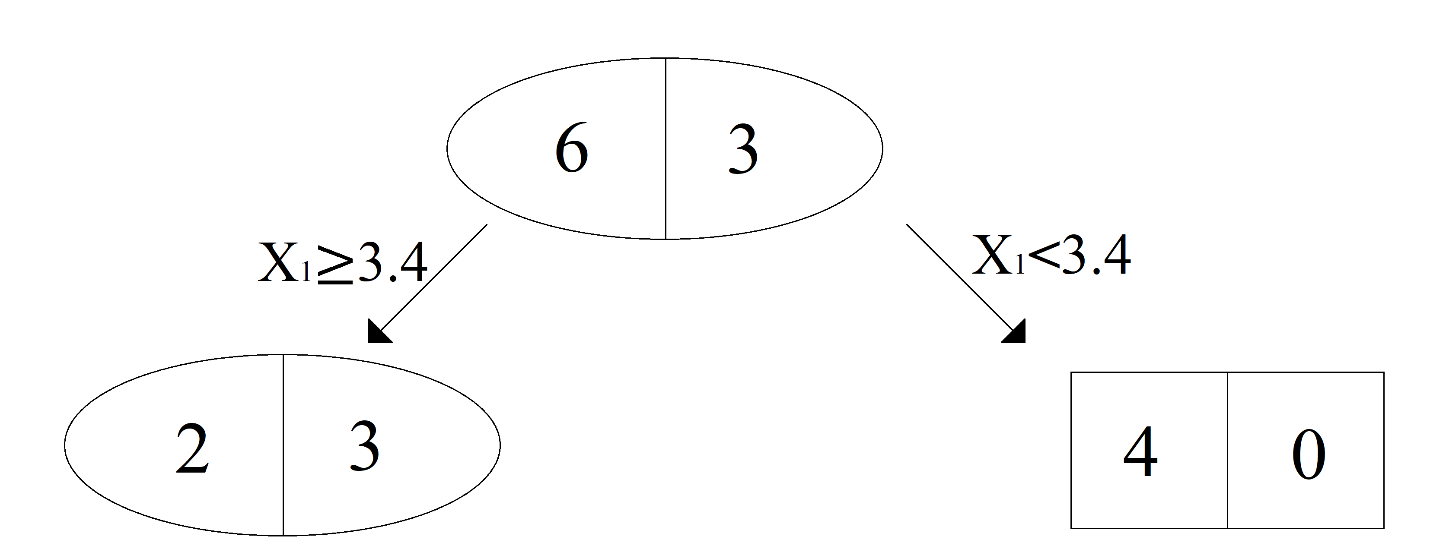
[-3.8, 99, 4.1, -11.2, 80, 18.1, 15, -1000, 46],

[11.2, 78.4, 14, 8, 1, 12, 47.1, 1, -7.1],

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

Минимальное значения критерия Джини = 2.40, соответствующее разбиению x1 ≥ 3.4

При ветвлении с одной стороны был получен набор только из элементов первого класса, с другой – двух первого и трёх второго.

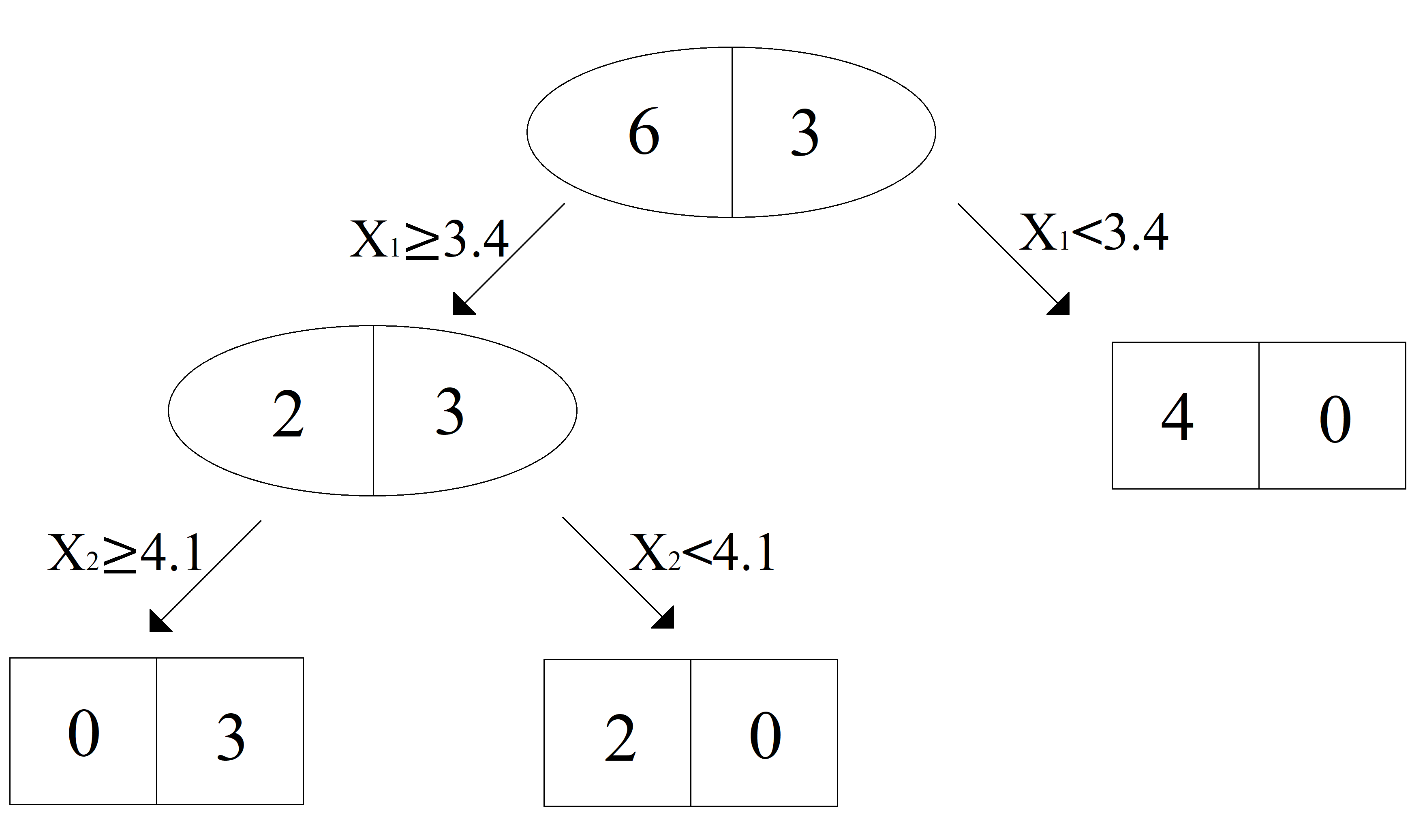


|  |  |  |  |
| --- | --- | --- | --- |
| **X1** | **X2** | **X3** | **Y** |
| 12,4 | -3,8 | 11,2 | 0 |
| 65 | 4,1 | 14 | 1 |
| 3,4 | 80 | 1 | 1 |
| 1000 | -1000 | 1 | 0 |
| 45 | 46 | -7,1 | 1 |

В данной таблице легко найти локально оптимальное решение. Это x2 ≥ 4.1, критерий

Джини = 0. В этом случае мы получим ещё две ветви, соответствующие набору из двух

элементов первого класс и набору из трёх элементов второго класса. Так как в наборах нет элементов разных классов, то задача построения дерева решений выполнена.



Классификация выполняется в соответствии с выделенными правилами:

Y = 1, если x1 ≥ 3.4 и x2 ≥ 4.1, иначе Y = 0

| **X1** | **X2** | **X3** | **Y** |
| --- | --- | --- | --- |
| 14 | -80 | 1,12 | 0 |
| -4 | 100 | -7,4 | 0 |
| 80,4 | -44,7 | 1 | 0 |
| 64,2 | 102,4 | 14 | 1 |

Программа:

table = [[12.4, 0, 65, -5, 3.4, 2.2, 0, 1000, 45],

[-3.8, 99, 4.1, -11.2, 80, 18.1, 15, -1000, 46],

[11.2, 78.4, 14, 8, 1, 12, 47.1, 1, -7.1],

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

gini = [[0, 0, 0, 0, 0, 0, 0, 0, 0],

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

[0, 0, 0, 0, 0, 0, 0, 0, 0]] j = 0

for j in range(3):

igini= -1

for x1 in table[j]:

igini=igini+1

left=0

right=0

i=-1

cleft=0

cright=0

for x in table[j]: #Подсчёт количества элементов "слева", "справа" и элементов второго класса Y=1

i=i+1

if (x>=x1):

right=right+1

cright=cright+table[3][i]

else:

left=left+1

cleft=cleft+table[3][i]

if (right != 0) and (left != 0): #Вычисление критерия Джини

gini[j][igini]= i+1 - ((cright\*\*2+(right-cright)\*\*2)/right + (cleft\*\*2+(left-cleft)\*\*2)/left)

else:

gini[j][igini]= 100 #Для избежания деления на 0, данное разбиение не имеет смысла

print (gini)

print (table)