

```

1  while nn >= 2:
2      countCutOffPrev = 0
3      qq = 0
4      while qq < wsls.countClasses-1:
5          ww = 0
6          while ww < wsls.countInstancesEachClassTraining[qq]:
7              ee = qq + 1
8              while ee < wsls.countClasses:
9                  rr = 0
10                 while rr < wsls.countInstancesEachClassTraining[ee]:
11                     tt = 0
12                     while tt < wsls.sizeVector:
13                         #первоначальное приближение вектора весов
14                         wsls.currentWeights[tt] =
15                             wsls.inputsClassTraining[qq][ww][tt] -
16                             wsls.inputsClassTraining[ee][rr][tt]
17                         tt += 1
18
19                     #Инициализировать столбец «значение скалярного произведения»
20                     wsls.initColScalarMul(wsls.currentWeights)
21                     #Определить целевую и противоположную категории
22                     mm = wsls.getMinMaxScalarMul()
23                     opCat = int(mm[0])
24                     taCat = int(mm[1])
25                     #Определить максимальное значение скалярного произведения в НЕ
26                     noTargMax = wsls.calcNoTarMax(taCat)
27                     #print(noTargMax)
28                     #Определить количество отсечённых экземпляров в целевой категории
29                     countCutOffTarget = wsls.calcCutOffSignTarget(taCat, noTargMax)
30                     #Определить минимальное значение скалярного произведения в НЕ
31                     noOppoMin = wsls.calcNoOppMin(opCat)
32                     #print(noOppoMin)
33                     #Определить количество отсечённых экземпляров в противоположной
34                     countCufOffOpposit = wsls.calcCutOffSignOpposit(opCat, noOppoMin)
35                     countCutOffCurrent = countCutOffTarget + countCufOffOpposit
36                     if countCutOffPrev < countCutOffCurrent:
37                         countCutOffPrev = countCutOffCurrent
38                         wsls.previousWeights = wsls.currentWeights.copy()
39                         countCutOffRight = countCutOffTarget
40                         categoryRight = taCat
41                         maxNoRight = noTargMax
42                         countCufOffLeft = countCufOffOpposit
43                         categoryLeft = opCat
44                         minNoLeft = noOppoMin
45                         wsls.gradientDescentScanning(categoryRight, categoryLeft,
46                                                         countCutOffPrev)
47
48                         rr += 1
49                         #ff += 1
50
51                     ee += 1
52                     #print(ww)
53                     ww += 1
54                     qq += 1
55
56                     #Инициализировать столбец «значение скалярного произведения»
57                     wsls.initColScalarMul(wsls.previousWeights)
58                     #становить в 1 столбец «признак отсеченности» в целевой категории
59                     thresholdRight = wsls.setColCutOffSignTarget(categoryRight, maxNoRight)
60                     thresholdRight = (thresholdRight // 2) + (maxNoRight // 2)
61                     #Сортировать указанную категорию по возрастанию «признака отсеченности»
62                     wsls.sortCategoryCutOff(categoryRight)
63                     #Установить в 2 столбец «признак отсеченности» в противоположной категории
64                     thresholdLeft = wsls.setColCutOffSignOpposit(categoryLeft, minNoLeft)
65                     thresholdLeft = (thresholdLeft // 2) + (minNoLeft // 2)
66                     #Сортировать указанную категорию по возрастанию «признака отсеченности»
67                     wsls.sortCategoryCutOff(categoryLeft)
68                     wsls.countInstancesEachClassTraining[categoryLeft] -= countCufOffLeft
69                     wsls.countInstancesEachClassTraining[categoryRight] -= countCutOffRight
70
71                     nn = 0
72                     rr = 0
73                     while rr < wsls.countClasses:
74                         if wsls.countInstancesEachClassTraining[rr] > 0:
75                             nn += 1
76                             rr += 1

```