Trabalho Prático 1

Henrique Furst Scheid - 2017014898

Janeiro 2022

Contents

1 Implementação **2**

2 Análise experimental **6**

3 Gráficos **8**

3.1 australian.dat 8

3.2 banana.dat 10

3.3 hayes-roth.dat 12

3.4 heart.dat 15

3.5 monk-2.dat 17

3.6 phoneme.dat 19

3.7 ring.dat 21

3.8 tae.dat 23

3.9 titanic.dat 26

3.10 twonorm.dat 28

Implementação

O algoritmo

O algoritmo implementado desempenha a função de classificar pontos de acordo com sua proximidade espacial a pontos anteriormente avaliados. O mecanismo para tal avaliação consiste em um registro *a priori* de um conjunto de dados pertencente ao mesmo 'espaço' dos dados a serem avaliados. Desse modo, durante a classificação resgatam-se do registro um número arbitrário (oferecido pelo usuário do algoritmo) os dados espacialmente mais próximos do dado a se classificar. A classificação resultante corresponde à classificação presente na maioria de tais dados resgatados. Trata-se, pois, de uma implementação do algoritmo **k-nearest-neighbours**.

Denominam-se os dados oferecidos *a priori* de **conjunto de aprendizado** e os dados a serem classificados pelo algoritmo de **conjunto de teste**. O algoritmo está implementado em *python*.

Estruturas de dados

kdtree

O aspecto chave do algoritmo está no armazenamento e resgate de dados espaciais, com número arbitrário de dimensões. Foi utilizada uma árvore k-dimensional (*k-d tree*) com vértices estruturais e valores registrados exclusivamente em folhas. Dessa forma, o resgate de dados próximos a um ponto oferecido consistem em uma pesquisa (de complexidade logarítmica) a uma folha de valor mais próximo que atua como 'centro', seguido de 'retrocessos' a vértices em seu ramo para a adição de suas folhas vizinhas mais próximas (espacialmente, os pontos dentro de um raio partindo de tal centro).

A implementação da *kdtree* consistiu na criação de duas classes principais: $KDtree$ e $BaseNode$. Esta se ramifica em duas subclasses $Node$ e $Leaf$, que representam os elementos da árvore.

KDtree possui métodos para:

- Construção de uma kdtree
- Pesquisa de pontos (retorna o ponto salvo mais próximo do ponto pesquisado)
- Resgate de todas as folhas que ramificam de um nó (retorna um conjunto de pontos)
- Exibição da kdtree, utilizado para testes

BaseNode possui métodos para:

- Construção de um vértice com valor interno (float) e referência para um vértice pai.
- Resgate do vértice pai
- Resgate de seu valor

Node possui métodos para:

- Construção de um vértice BaseNode com valor comparador e referência ao vértice paterno, adicionado de:
  - dois BaseNodes filhos (esquerdo para valores menores que o comparador, direito para valores maiores que o comparador)
  - dimensão do espaço à qual o comparador pertence
- Configuração e resgate do vértice filho esquerdo
- Configuração e resgate do vértice filho direito

Leaf possui métodos para:

- Construção de um vértice BaseNode com valor de classificação e referência ao vértice paterno, adicionado das coordenadas espaciais do ponto referente.

k-nearest-neighbours

Criou-se uma classe xNN para acomodar o conjunto de teste, assim como a *kdtree* populada com o conjunto de aprendizado. Além dos dados, a classe possui os métodos correspondentes ao algoritmo *k-nearest-neighbours* em si, que toma como parâmetro o número de vizinhos a serem utilizados.

A execução do algoritmo exige computação sucessiva de pontos vizinhos do ponto de teste P. Tais vizinhos são armazenados em uma **fila de prioridade** (implementada no módulo *heapq* da biblioteca padrão de *python*). A fila de prioridade permite a ordenação de novos vizinhos mediante menor distância com relação a P a cada inserção, de forma que pontos mais distantes ocupam as folhas. Tal fila é sempre "podada" para manter apenas os K vizinhos mais próximos de P.

Algoritmos

Construção de kdtree

A construção de uma árvore k-dimensional é feita através da construção recursiva de nós. Uma lista inicial com todos os pontos é separada "maiores" e "menores" através da comparação com o valor mediano de uma das dimensões. Um novo nó é criado cujos filhos esquerdos são o ramo "menores" e os direitos o ramo "maiores". Tal processo é feito recursivamente para novas dimensões até que haja no máximo 2 pontos, instante no qual folhas são criadas para armazená-los e o algoritmo retorna.

```
constroiArvore(lista_pontos, dimensão, nó_pai):
    Se 'lista_pontos' está vazio:
        retorne
    Se 'lista_pontos' possui 2 pontos:
        crie um novo nó filho de 'nó_pai'
        o valor do novo nó é a mediana dos pontos na dimensão atual
        crie duas folhas para armazenar os pontos:
            direita para o ponto maior que a mediana na dimensão atual
            esquerda para o outro
        insira as folhas como filhas do nó
        retorne o novo nó

    Se 'lista_pontos' possui 1 ponto:
        retorne uma folha com o ponto

    Senão:
        crie um novo nó filho de 'nó_pai'
        o valor do novo nó é a mediana dos pontos da dimensão atual
        divida 'lista_pontos' em pontos maiores e menores que a mediana
            na dimensão atual

        filho direito do novo nó = constoiArvore(pontos maiores,
                                                 próxima dimensão,
                                                 nó atual)
        filho esquerdo do novo nó = constroiArvore(pontos menores,
                                                 próxima dimensão,
                                                 nó atual)
        retorne o novo nó
```

Consulta de ponto mais próximo

Para resgatar dados a partir de um ponto oferecido, basta comparar recursivamente o valor do input com o valor do comparador no nó em sua dimensão correspondente, a começar da raiz da árvore. Se o valor do input for maior, pesquise recursivamente no filho da direita. Caso contrário, pesquise recursivamente no filho da esquerda. Chegando-se em uma folha, retorne-a: este é o valor desejado.

Resgatar as folhas a partir de um nó

Dado um nó da kdtree, crie uma lista vazia. Adicione a essa lista as folhas dos nós do seu filho à esquerda, depois as folhas dos seus filhos da direita. Para encontrar tais filhos, execute este mesmo algoritmo, recursivamente.

```
folhas(nó):
    Se 'nó' é vazio:
        retorne lista vazia
    Se 'nó' é uma folha:
        retorne uma lista contendo 'nó'
    Senão:
        crie uma lista vazia
        adicione à lista o valor de folhas(filho da esquerda de 'nó')
        adicione à lisa o valor de folhas(filho da direita de 'nó')
        retorne a lista
```

K-nearest neighbours

O programa recebe uma base de dados e uma lista com os números de vizinhos a serem utilizados para a execução do algoritmo. Dessa forma, com apenas uma chamada o programa pode exibir resultados com diferentes graus de eficácia de acordo com o número de vizinhos oferecido. A base de dados é dividida em aprendizado (70% dos dados) e teste (30% dos dados). Um objeto xNN recebe tais conjuntos, criando uma *kdtree* a partir do primeiro e fazendo sucessivas classificações (uma para cada número de vizinhos oferecido) no segundo.

O programa realiza o seguinte procedimento para todos os elementos do conjunto de teste:

1. Dado um ponto P do conjunto de teste para classificação e um número K de vizinhos como parâmetro, o algoritmo pesquisa em sua kdtree (habitada pelo conjunto de aprendizado) o ponto X mais próximo de P.
2. Em seguida, o algoritmo retrocede para o pai X' de tal ponto. Se as folhas que ramificam de X totalizam um número de vizinhos menor que K, retroceda para o pai de X', X'' e pegue as folhas que retrocedem de X''. Repita o processo até encontrar K vizinhos.
3. Se o valor comparador do próximo pai a se retroceder estiver a uma distância maior do que a distância máxima entre os vizinhos atuais e o ponto P, interrompa laço. Senão, retroceda novamente, remova os vizinhos mais distantes para permanecer com K vizinhos sendo considerados.

Análise experimental

Foram escolhidos 10 conjuntos de dados, todos contendo apenas valores numéricos. As sucessivas avaliações de cada conjunto foram armazenadas em dois tipos de arquivo:

- Um *output* contendo cada ponto avaliado e sua classificação
- Um contendo o desempenho do algoritmo (acurácia, precisão e revocação) diante de todas as avaliações

O arquivo com os dados de desempenho foi utilizado para a geração de uma melhor vizualização dos resultados. A seguir estão as observações acerca do desempenho do algoritmo para cada conjunto de dados. As classificações aos quais as vizualizações se referem tratam-se das classes (*tags*) atribuídas pelo algoritmo, e não das originais.

australian.dat

Percebe-se que o aumento do número de vizinhos possui um efeito inversamente proporcional em dois momentos: entre o precisão e revocação de uma classe, e entre o comportamentos das classes. As curvas "Precisão da classificação 0" e "Revocação da clasificação 1" São similates, tal como "Precisão da classificação 1" e "Revocação da clasificação 0". Além disso, tais pares parecem espelhados. O pico de acurácia em 36 vizinhos coincide com o "ponto médio" de todas as curvas.

banana.dat

O comportamento se assemelha ao conjunto de dados anterior. Não há simetria evidente entre pares de gráficos todavia, visto que "Precisão da classificação -1" e "Revocação da classificação 1" se comportam como uma curva com concavidade para baixo, como o poderia fazer uma função quadrática. Isso se opõe ao comportamento de reta do outro par, resultando em um interessante comportamento logarítmico (observado) na curva de acurácia.

hayes-roth.dat

A classificação de valores "3" apresenta degeneração em todos os valores exceto 4. Isso se dá pois o algoritmo não classificou nenhum ponto como "3" a partir de 8 vizinhos, tornando o cálculo indefinido. A precisão da classificação "1" apresenta um comportamento (observado) oscilatório de amplitude crescente, enquanto a revocação da classificação "2" oscila com amplitude decrescente.

heart.dat

Todas as curvas apresentam concavidade para cima, em um comportamento ligeiramente similar. Como resultado, a acurácia se mostra mais alta nos pontos periféricos do gráfico observado.

monk-2.dat

A precisão da classificação "1", tal como a revocação da classificação "0", apresentam um comportamento bastante mais previsível que seu par complementar. Comum a todos, os melhores resultados ocorrem com poucos vizinhos, o que se reflete na acurácia.

phoneme.dat

Todas as métricas apresentam um decrescimento (observado) previsível similar a uma função inversa. Fugindo a este padrão estão as extremidades dos gráficos, que podem sugerir um comportamento oscilatório. A Acurácia reflete o comportamento previsível tal como a influência das exceções, resultando em valor ótimo com 8 vizinhos ao invés de 4.

ring.dat

Dois comportamentos distintos são observados. a Partir de 8 valores, a precisão das classificações com "0" é de 100%, tal como a revocação das classificações com "1". Paralelamente, a precisão das classificações com "1" e a revocação das classificações com "0" se comportam muito similarmente a uma função inversa. Como resultado, a acurácia máxima é tida com o menor número de vizinhos testado, 4.

tae.dat

Simetrias foram observadas tal como nos demais conjuntos de dados, apesar de não haverem comportamentos regulares. Observa-se uma tendência a maiores precisões em torno de 50 vizinhos, um valor mais alto comparado aos demais testes. Apenas a classificação "3" apresentou revocação igualmente alta nesse valor, enquanto os demais apresentam comportamento decrescente. A acurácia máxima foi atingida em torno de 40 vizinhos.

titanic.dat

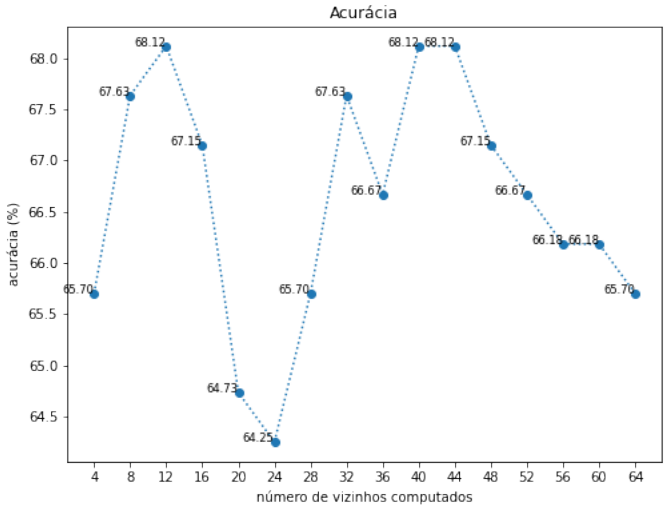
Todos os gráficos apresentam comportamento parecido com funções degrau: valores constantes seguidos de súbitas variações. Além disso, todos os valores máximos encontram-se com números pequenos de vizinhos. Acurácia máxima encontrada em torno de 12 vizinhos.

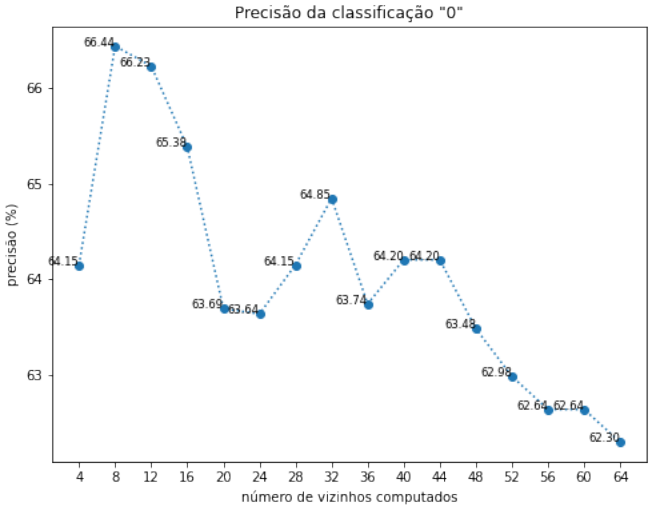
twonorm.dat

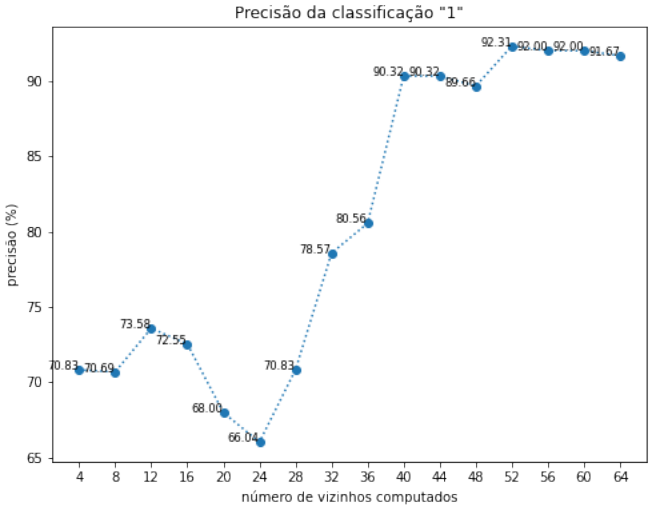
Todas as métricas paresentam súbito crescimento seguido de pequena oscilação em torno de uma constante próxima a 97%. Valores máximos encontrados com números médios de vizinhos (entre 20 e 40). Acurácia máxima obtida com 24 vizinhos.

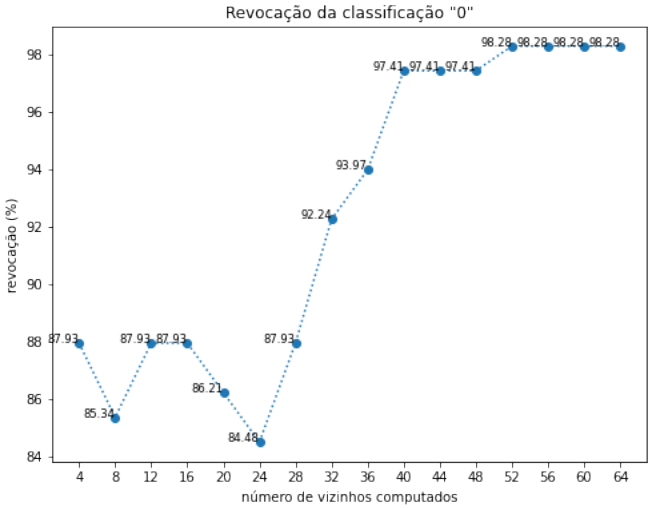
Gráficos

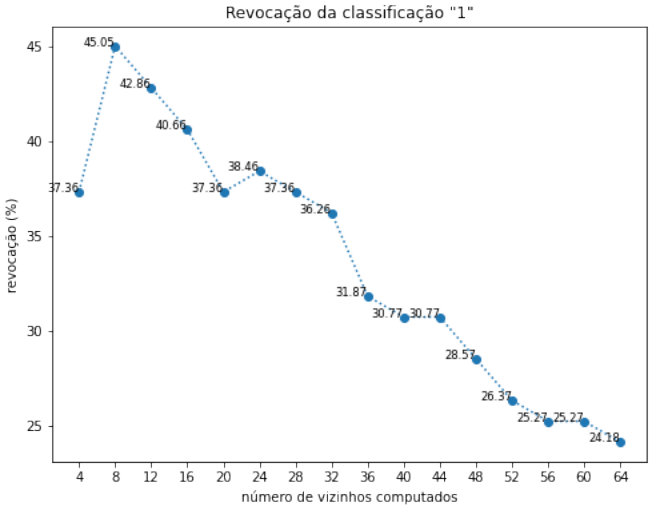
3.1 australian.dat



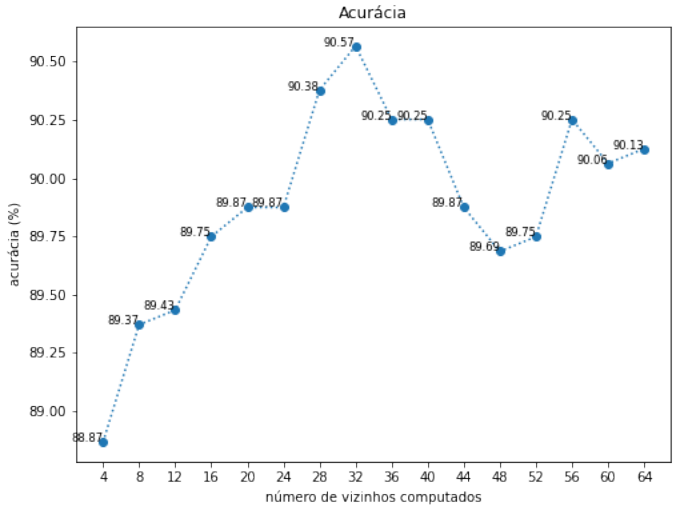


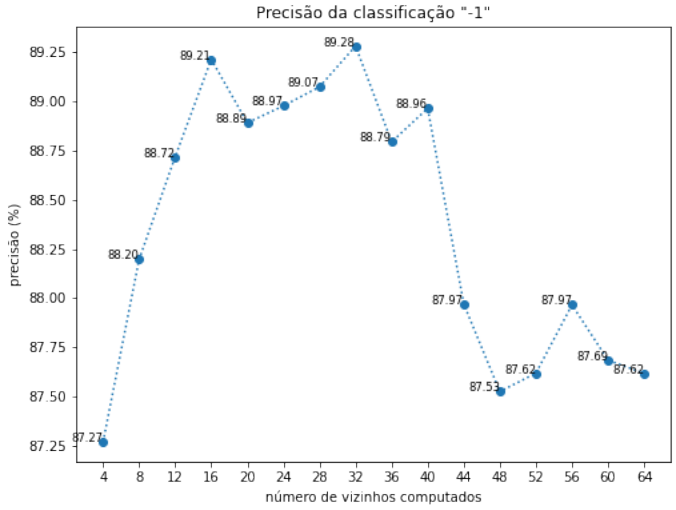

Precisão da classificação "1"
precisão (%)
número de vizinhos computados
70.83
70.69
73.58
72.55
68.00
66.04
70.83
78.57
80.56
90.32
90.32
89.66
92.31
92.00
92.00
91.67

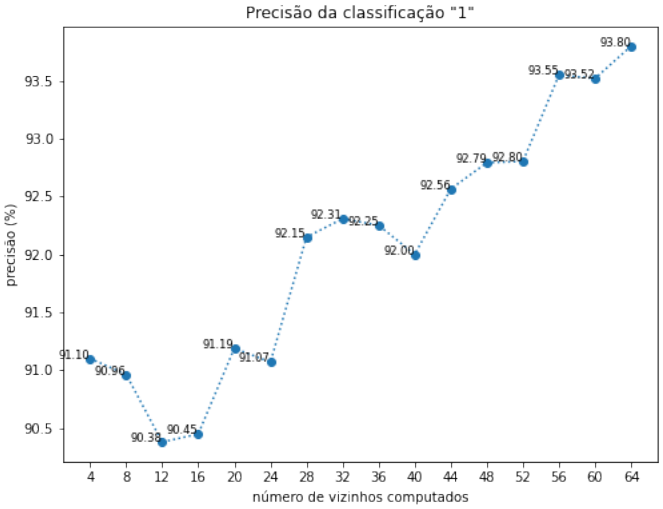

Revocação da classificação "0"
revocação (%)
número de vizinhos computados
87.93
85.34
87.93
87.93
86.21
84.48
87.93
92.24
93.97
97.41
97.41
97.41
98.28
98.28
98.28
98.28

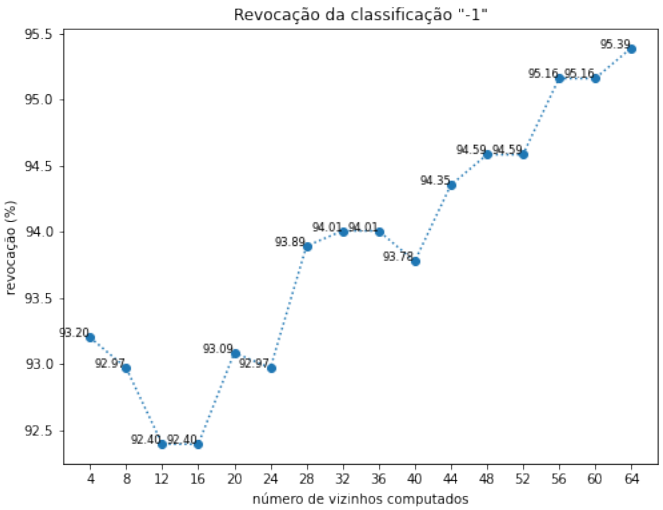

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
37.36
45.05
42.86
40.66
37.36
38.46
37.36
36.26
31.87
30.77
30.77
28.57
26.37
25.27
25.27
24.18

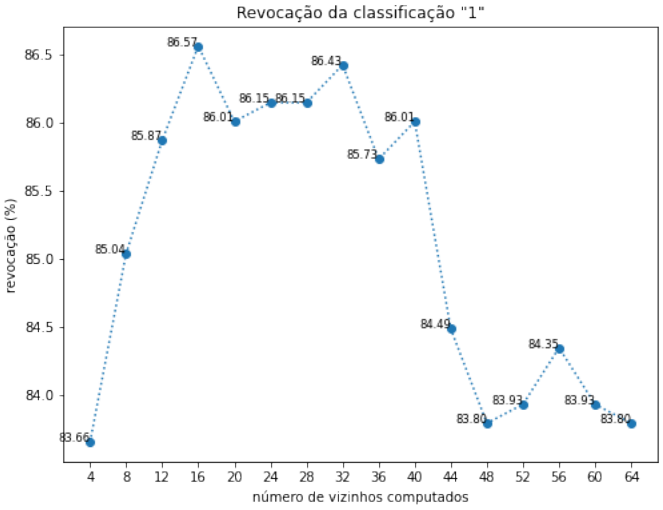
3.2 banana.dat



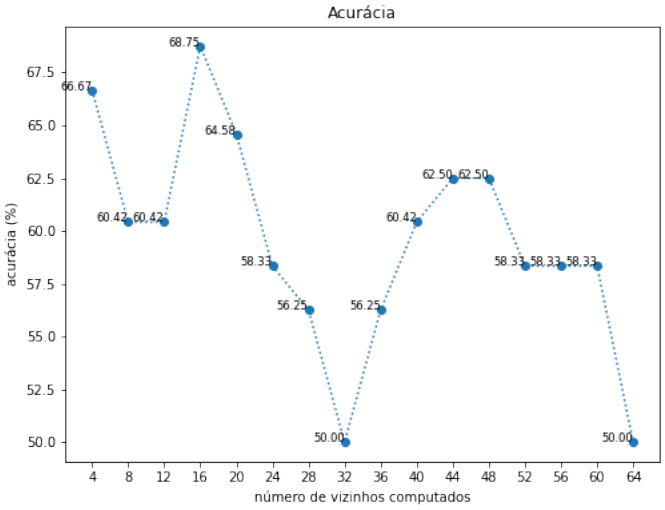


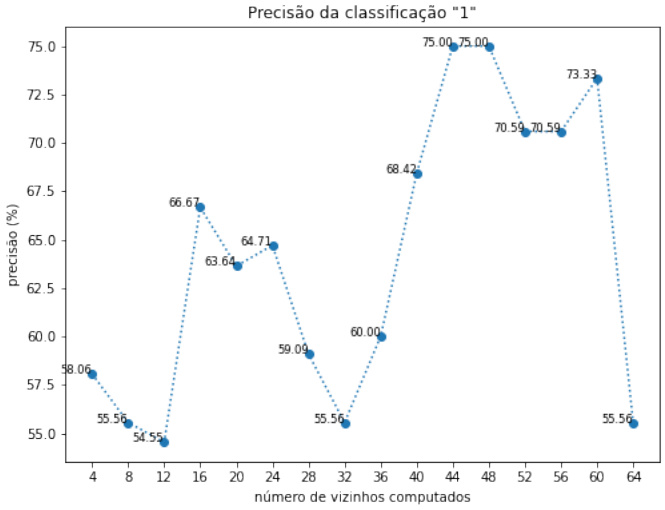

Precisão da classificação "1"
precisão (%)
número de vizinhos computados
91.10
90.96
90.38
90.45
91.19
91.07
92.15
92.31
92.25
92.00
92.56
92.79
92.80
93.55
93.52
93.80

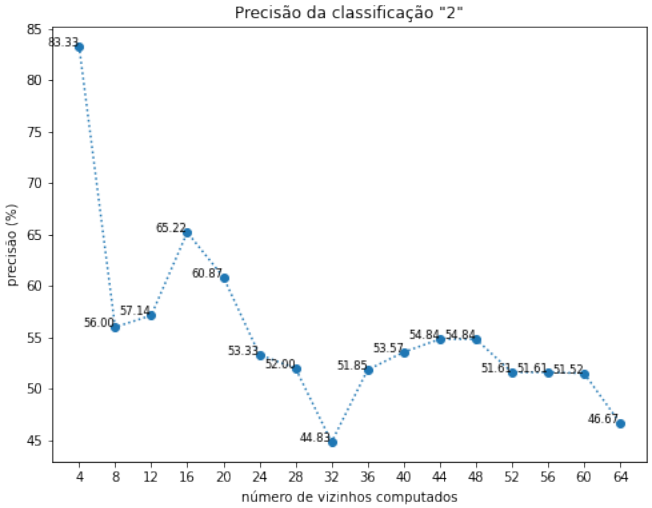

Revocação da classificação "-1"
revocação (%)
número de vizinhos computados
93.20
92.97
92.40
92.40
93.09
92.97
93.89
94.01
94.01
93.78
94.35
94.59
94.59
95.16
95.16
95.39

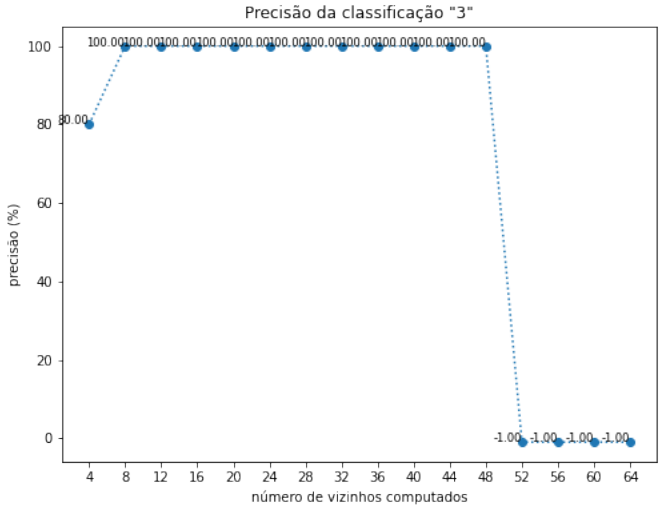

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
83.66
85.04
85.87
86.57
86.01
86.15
86.15
86.43
85.73
86.01
84.49
83.80
83.93
84.35
83.93
83.80

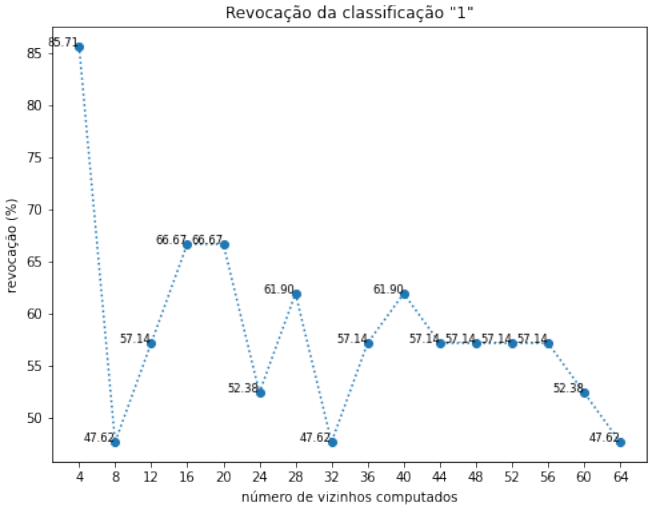
3.3 hayes-roth.dat

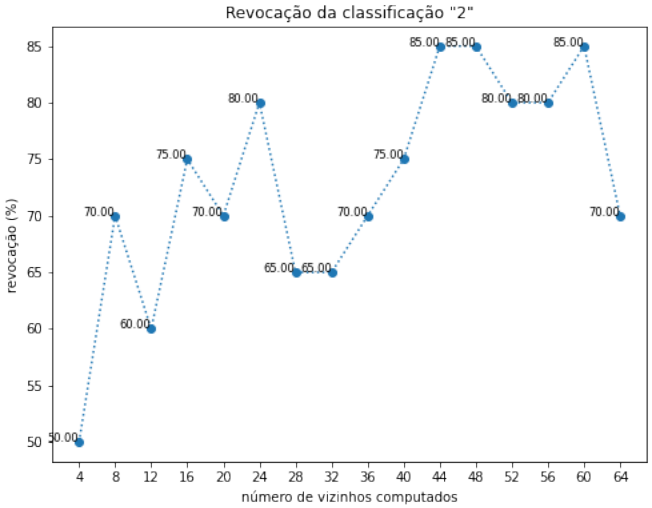


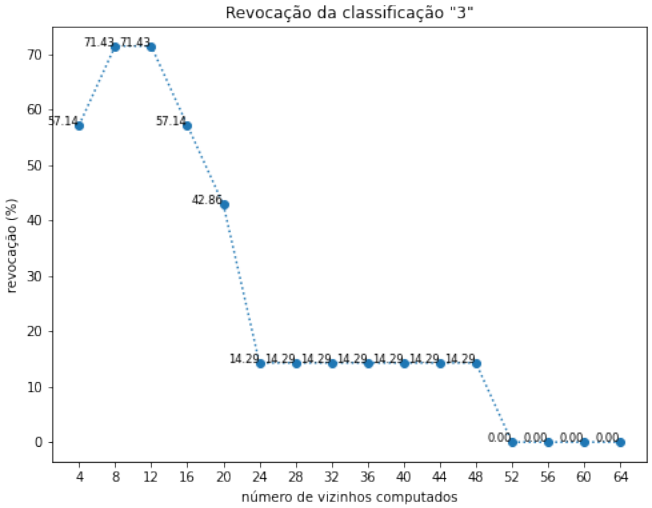



Precisão da classificação "2"
precisão (%)
número de vizinhos computados
83.33
56.00
57.14
65.22
60.87
53.33
52.00
44.83
51.85
53.57
54.84
54.84
51.61
51.61
51.52
46.67

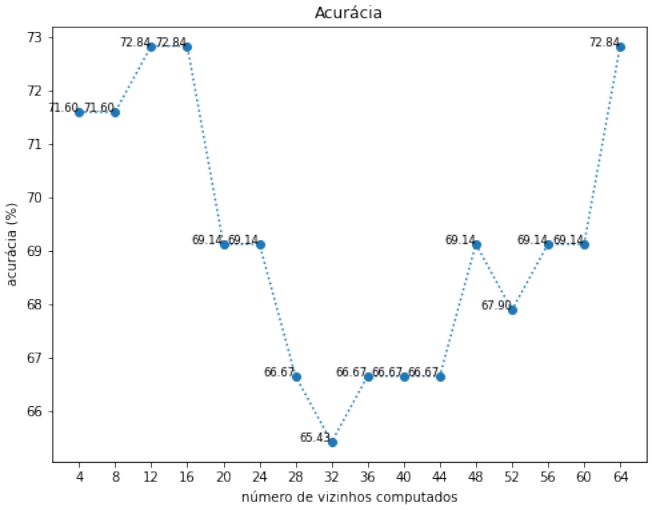

Precisão da classificação "3"
precisão (%)
número de vizinhos computados
80.00
-1.00
-1.00
-1.00
-1.00

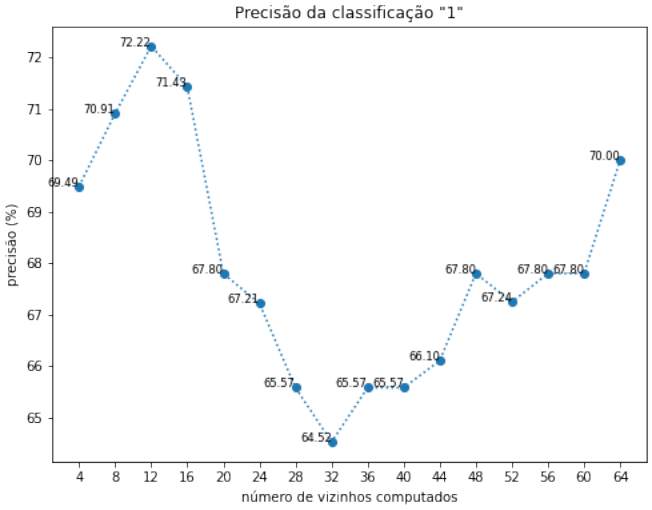

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
85.71
47.62
57.14
66.67
66.67
52.38
61.90
47.62
57.14
61.90
57.14
57.14
57.14
57.14
52.38
47.62

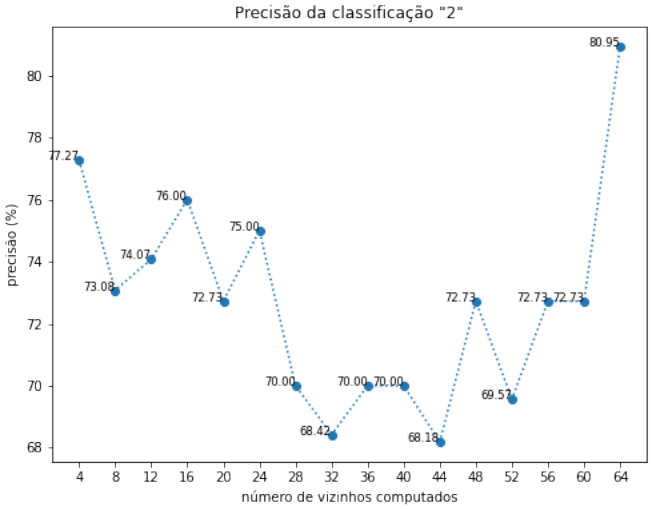

Revocação da classificação "2"
revocação (%)
número de vizinhos computados
50.00
70.00
60.00
75.00
70.00
80.00
65.00
65.00
70.00
75.00
85.00
85.00
80.00
80.00
85.00
70.00

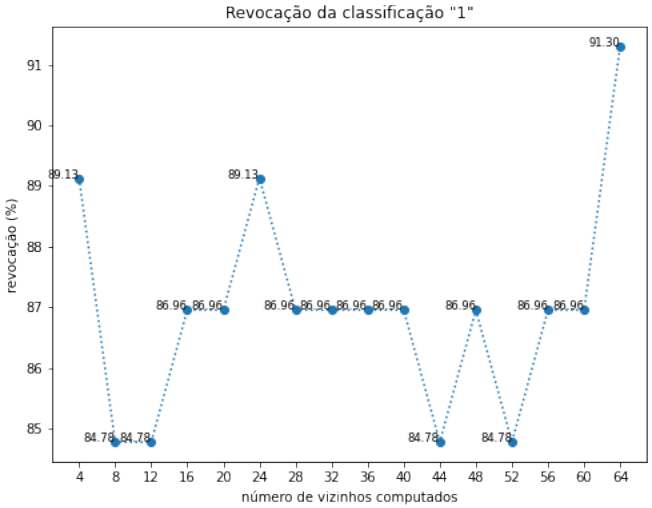

Revocação da classificação "3"
revocação (%)
número de vizinhos computados
57.14
71.43
71.43
57.14
42.86
14.29
14.29
14.29
14.29
14.29
14.29
14.29
0.00
0.00
0.00
0.00

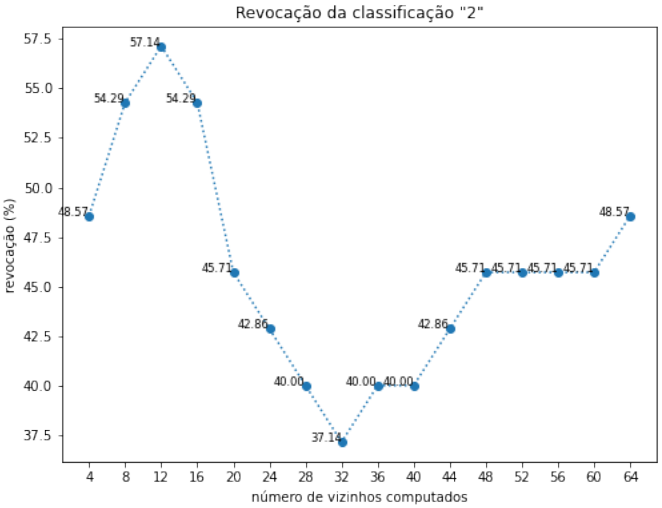
3.4 heart.dat


Acurácia
acurácia (%)
número de vizinhos computados
72.84
72.84
72.84
71.60
71.60
69.14
69.14
69.14
69.14
69.14
69.14
67.90
66.67
66.67
66.67
66.67
65.43

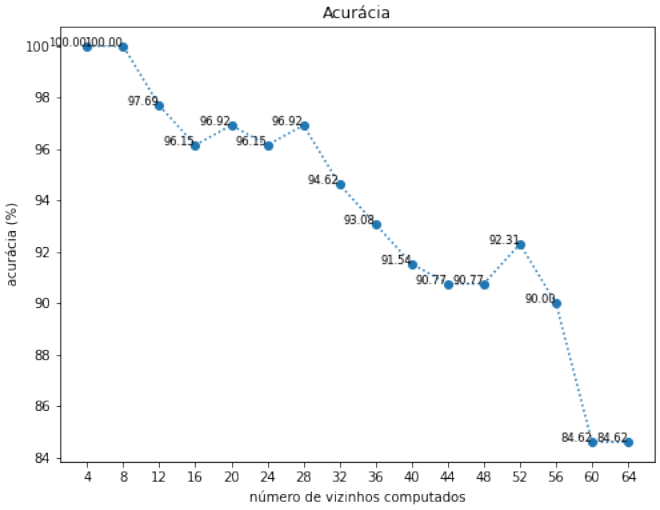

Precisão da classificação "1"
precisão (%)
número de vizinhos computados
72.22
71.43
70.91
69.49
70.00
67.80
67.80
67.80
67.80
67.21
67.24
66.10
65.57
65.57
65.57
64.52

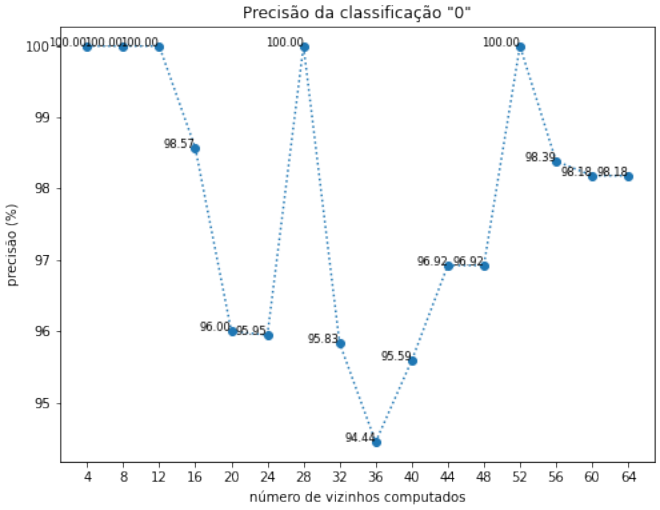

Precisão da classificação "2"
precisão (%)
número de vizinhos computados
77.27
73.08
74.07
76.00
72.73
75.00
70.00
68.42
70.00
70.00
68.18
72.73
69.57
72.73
72.73
80.95

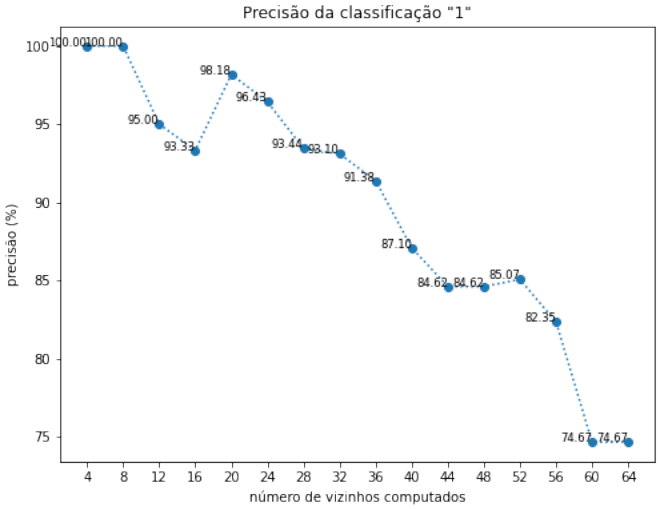

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
89.13
84.78
84.78
86.96
86.96
89.13
86.96
86.96
86.96
86.96
84.78
86.96
84.78
86.96
86.96
91.30

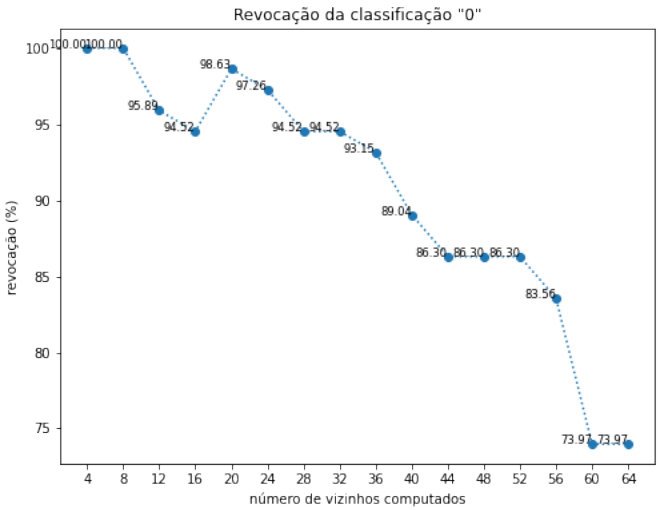

Revocação da classificação "2"
revocação (%)
número de vizinhos computados
48.57
54.29
57.14
54.29
45.71
42.86
40.00
37.14
40.00
40.00
42.86
45.71
45.71
45.71
45.71
48.57

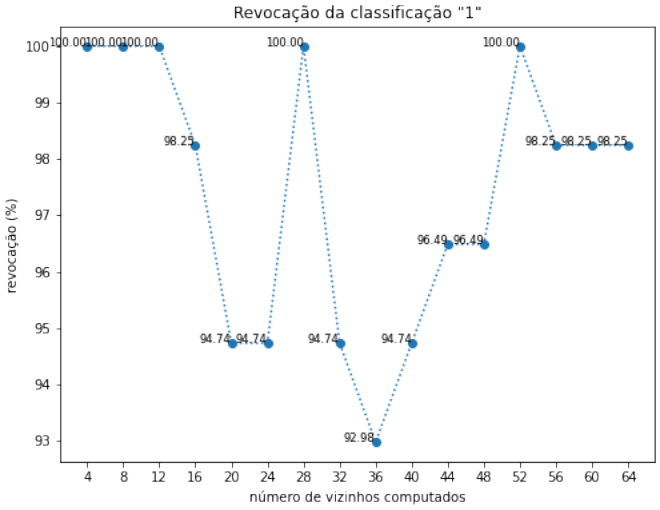
3.5 monk-2.dat


Acurácia
acurácia (%)
número de vizinhos computados
100.00 100.00 97.69 96.15 96.92 96.15 96.92 94.62 93.08 91.54 90.77 90.77 92.31 90.00 84.62 84.62

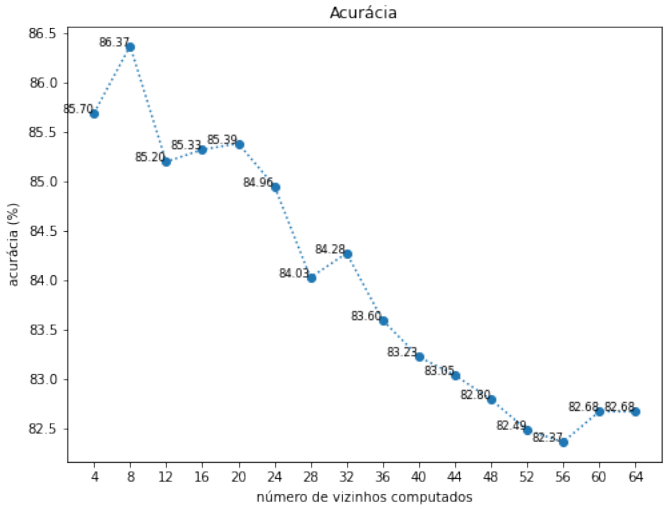

Precisão da classificação "0"
precisão (%)
número de vizinhos computados
100.00 100.00 100.00 98.57 96.00 95.95 100.00 95.83 94.44 95.59 96.92 96.92 100.00 98.39 98.18 98.18

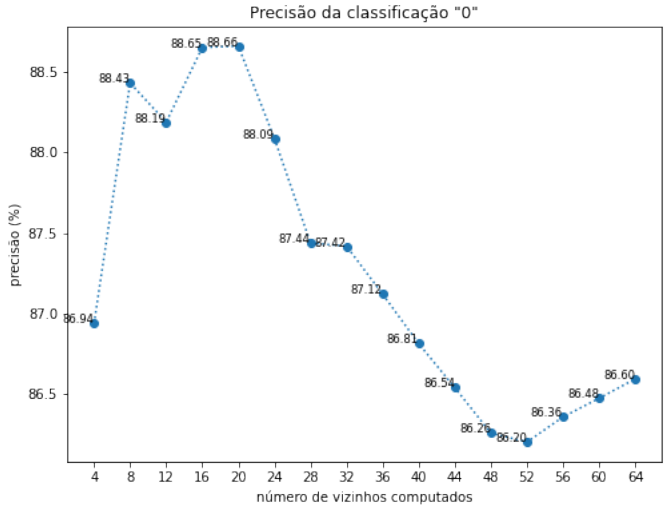

Precisão da classificação "1"
precisão (%)
número de vizinhos computados
100.00 100.00 98.18 96.43 95.00 93.33 93.44 93.10 91.38 87.10 84.62 84.62 85.07 82.35 74.67 74.67

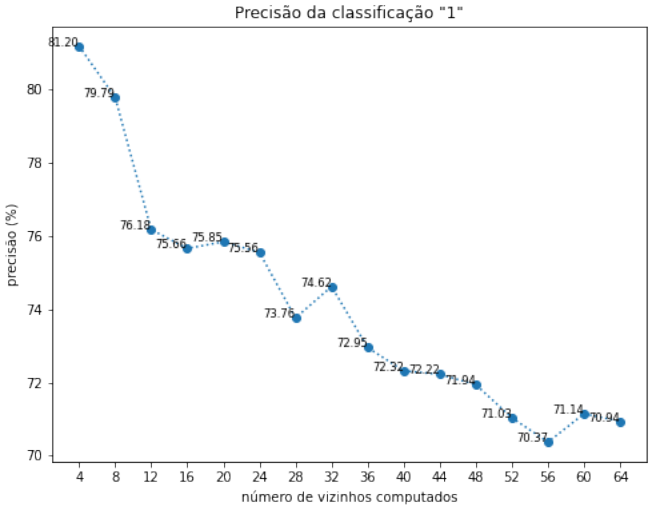

Revocação da classificação "0"
revocação (%)
número de vizinhos computados
100.00 100.00 98.63 97.26 95.89 94.52 94.52 94.52 93.15 89.04 86.30 86.30 86.30 83.56 73.97 73.97

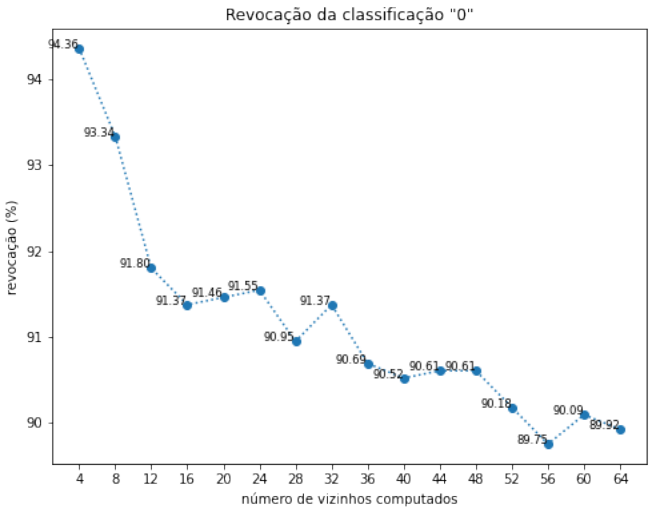

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
100.00 100.00 100.00 98.25 94.74 94.74 100.00 94.74 92.98 94.74 96.49 96.49 100.00 98.25 98.25 98.25

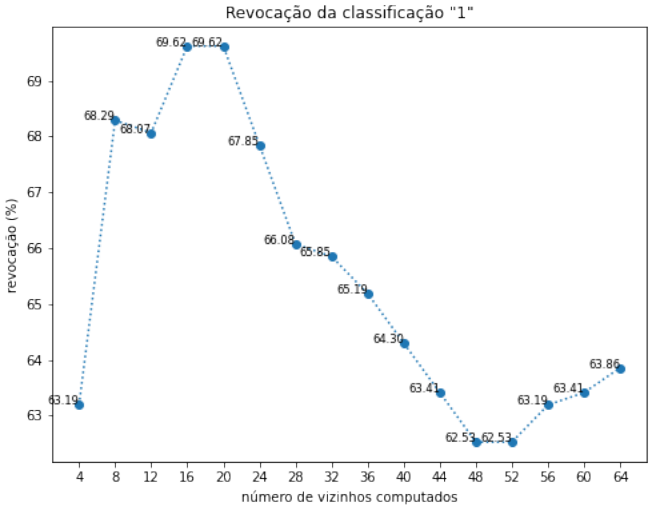
3.6 phoneme.dat



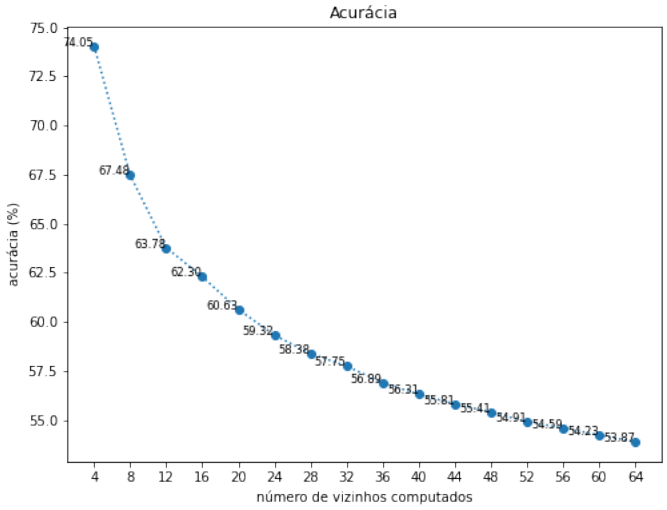


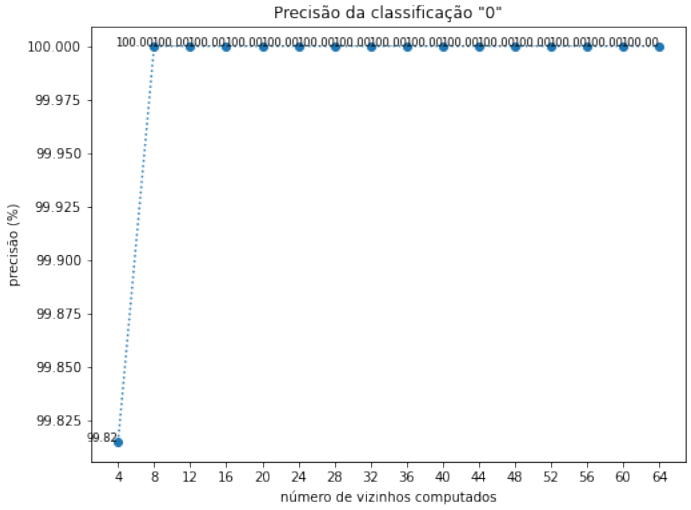

Precisão da classificação "1"
precisão (%)
número de vizinhos computados
81.20
79.79
76.18
75.66
75.85
75.56
74.62
73.76
72.95
72.32
72.22
71.94
71.03
70.37
71.14
70.94

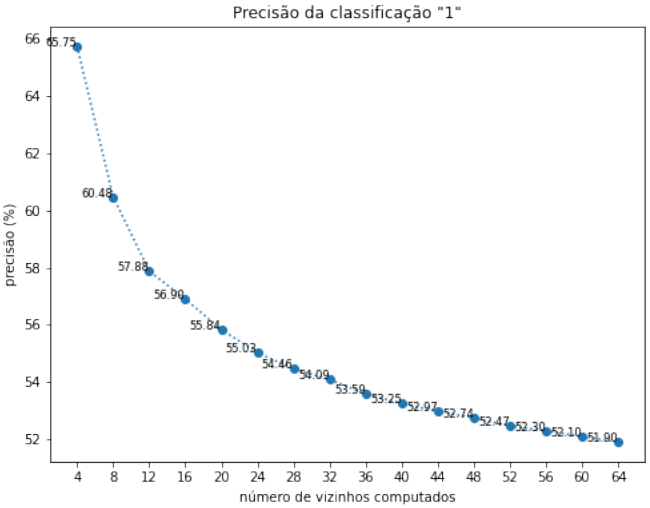

Revocação da classificação "0"
revocação (%)
número de vizinhos computados
94.36
93.34
91.80
91.37
91.46
91.55
90.95
91.37
90.69
90.52
90.61
90.61
90.18
89.73
90.09
89.92

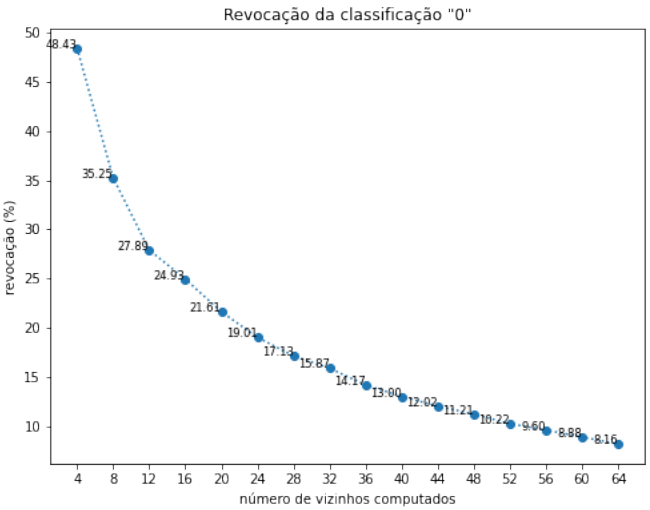

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
63.19
68.29
68.07
69.62
69.62
67.85
66.08
65.85
65.19
64.30
63.41
62.53
62.53
63.19
63.41
63.86

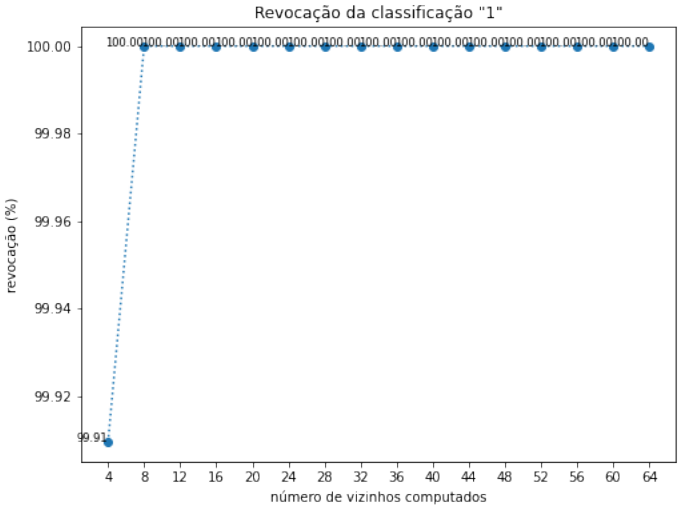
3.7 ring.dat


Acurácia
acurácia (%)
número de vizinhos computados
74.05
67.48
63.78
62.30
60.63
59.32
58.38
57.75
56.89
56.31
55.81
55.41
54.91
54.59
54.23
53.87

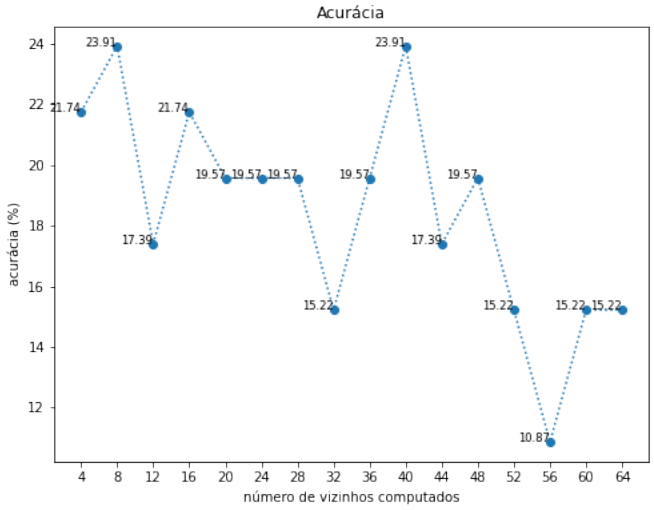

Precisão da classificação "0"
precisão (%)
número de vizinhos computados
99.82
100.00

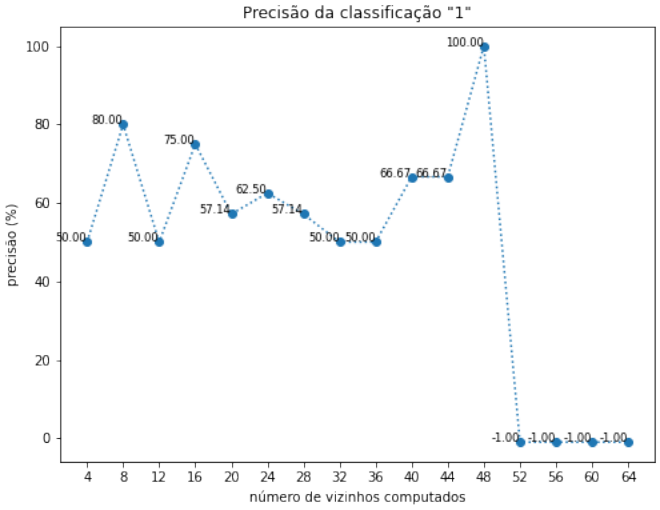

Precisão da classificação "1"
precisão (%)
número de vizinhos computados
65.75
60.48
57.88
56.90
55.84
55.03
54.46
54.09
53.50
53.25
52.97
52.74
52.47
52.30
52.10
51.90

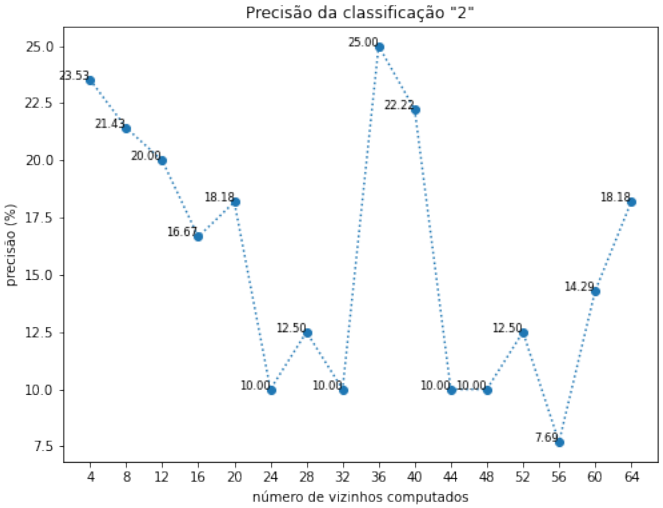

Revocação da classificação "0"
revocação (%)
número de vizinhos computados
48.43
35.25
27.89
24.93
21.61
19.01
17.13
15.87
14.17
13.00
12.02
11.21
10.22
9.60
8.88
8.16

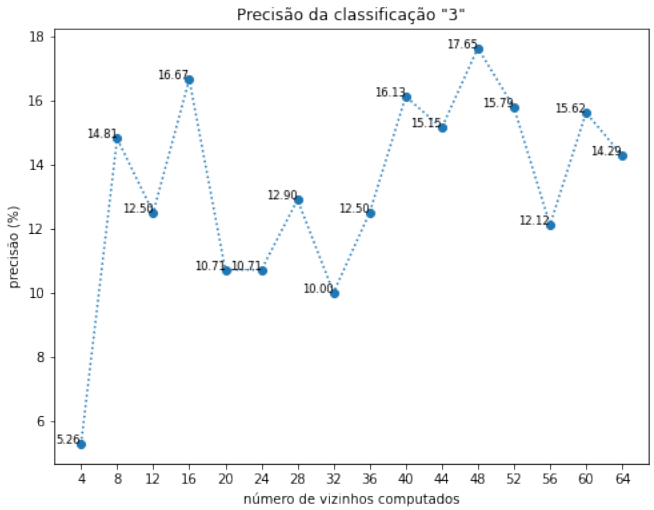

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
99.91
100.00

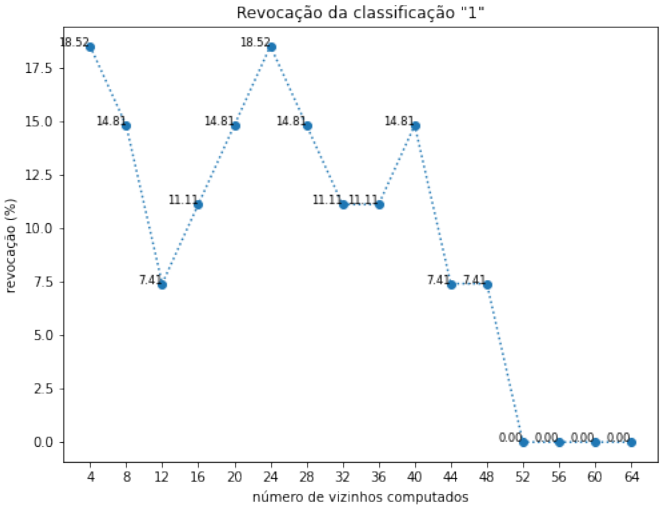
3.8 tae.dat

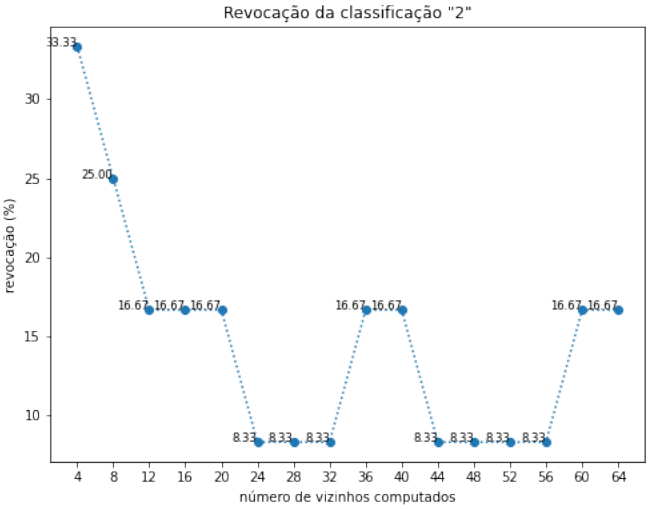

Acurácia
acurácia (%)
número de vizinhos computados
23.91
21.74
17.39
19.57
15.22
10.87

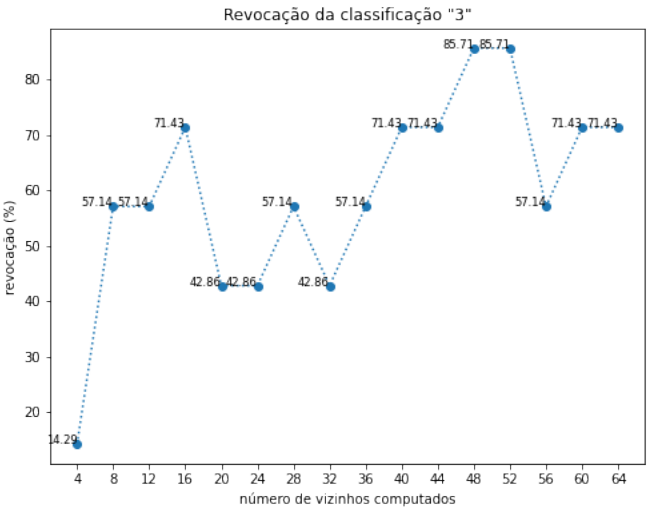

Precisão da classificação "1"
precisão (%)
número de vizinhos computados
100.00
80.00
75.00
66.67
62.50
57.14
50.00
-1.00


Precisão da classificação "2"
precisão (%)
número de vizinhos computados
25.00
23.53
22.22
21.43
20.00
18.18
18.18
16.67
14.29
12.50
12.50
10.00
10.00
10.00
10.00
7.69

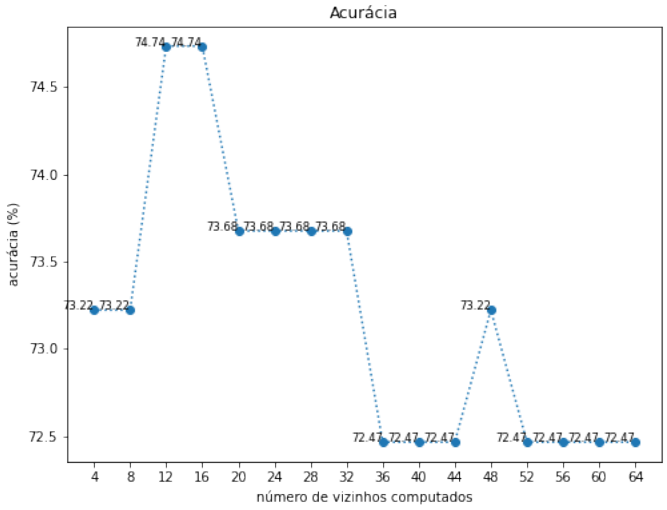

Precisão da classificação "3"
precisão (%)
número de vizinhos computados
17.65
16.67
16.13
15.79
15.62
15.15
14.81
14.29
12.90
12.50
12.50
12.12
10.71
10.71
10.00
5.26

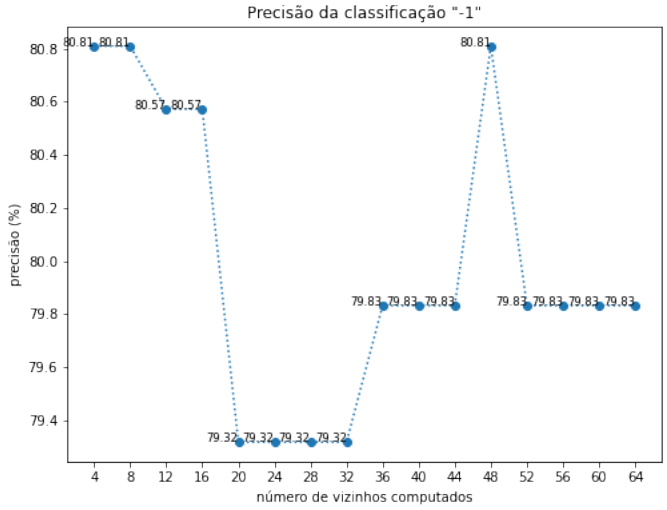

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
18.52
18.52
14.81
14.81
14.81
14.81
14.81
11.11
11.11
11.11
7.41
7.41
7.41
0.00
0.00
0.00
0.00

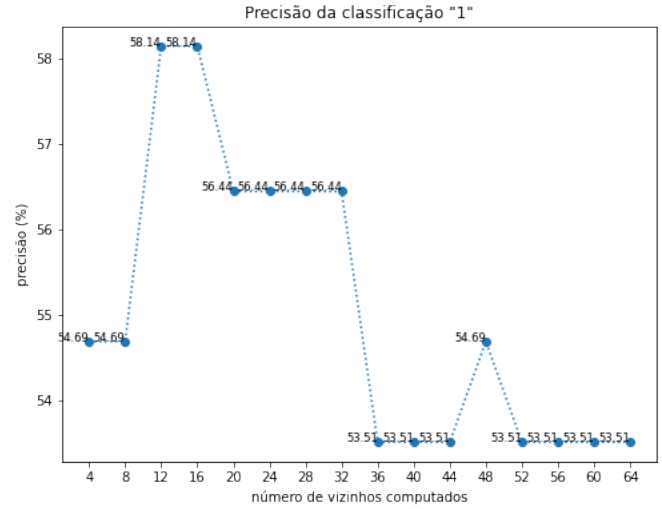

Revocação da classificação "2"
revocação (%)
número de vizinhos computados
33.33
25.00
16.67
16.67
16.67
16.67
8.33
8.33
8.33
8.33
16.67
16.67
16.67
8.33
8.33
8.33
8.33
8.33
16.67
16.67

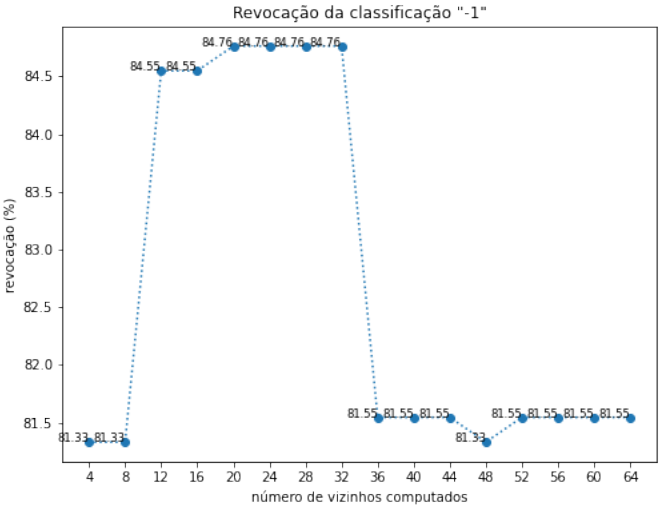

Revocação da classificação "3"
revocação (%)
número de vizinhos computados
14.29
57.14
57.14
71.43
42.86
42.86
42.86
57.14
57.14
71.43
71.43
85.71
85.71
57.14
71.43
71.43

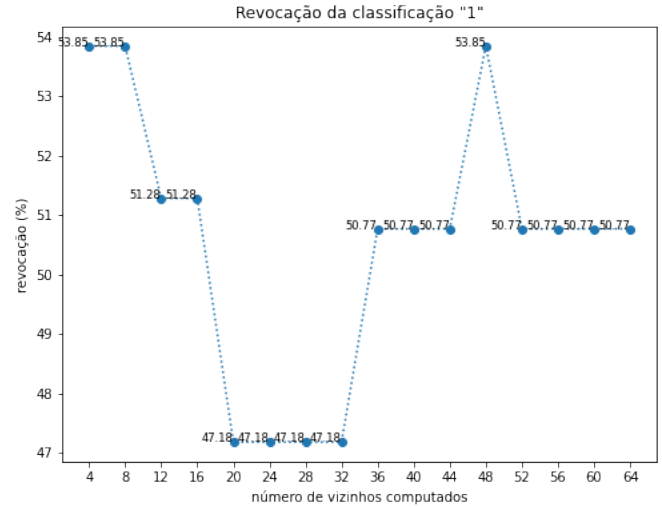
3.9 titanic.dat



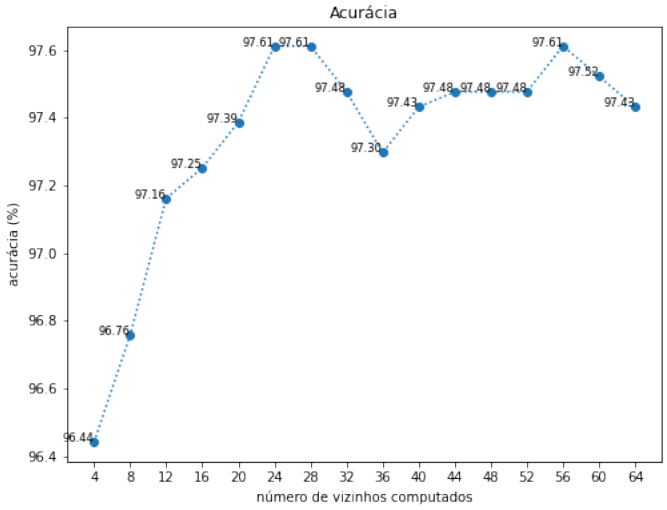


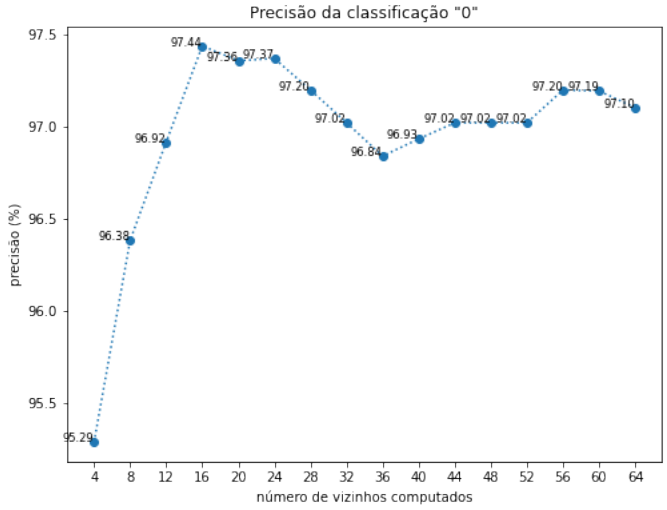

Precisão da classificação "1"
precisão (%)
número de vizinhos computados
58.14 58.14 58.14
56.44 56.44 56.44 56.44 56.44
54.69 54.69
54.69
53.51 53.51 53.51
53.51 53.51 53.51 53.51

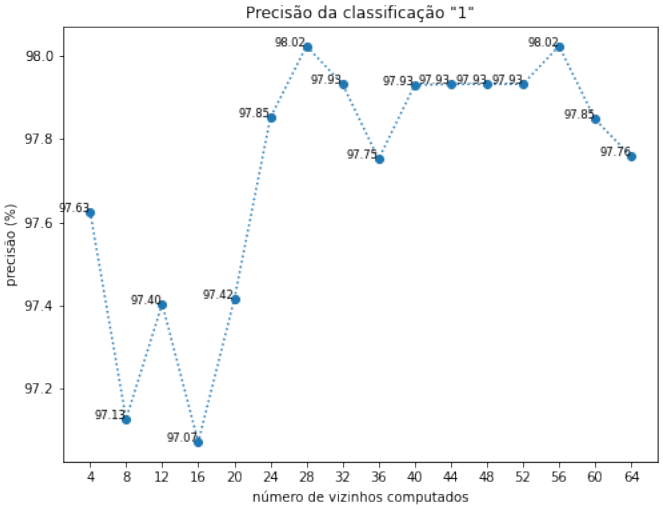

Revocação da classificação "-1"
revocação (%)
número de vizinhos computados
84.76 84.76 84.76 84.76 84.76
84.55 84.55
81.33 81.33
81.55 81.55 81.55
81.33
81.55 81.55 81.55 81.55

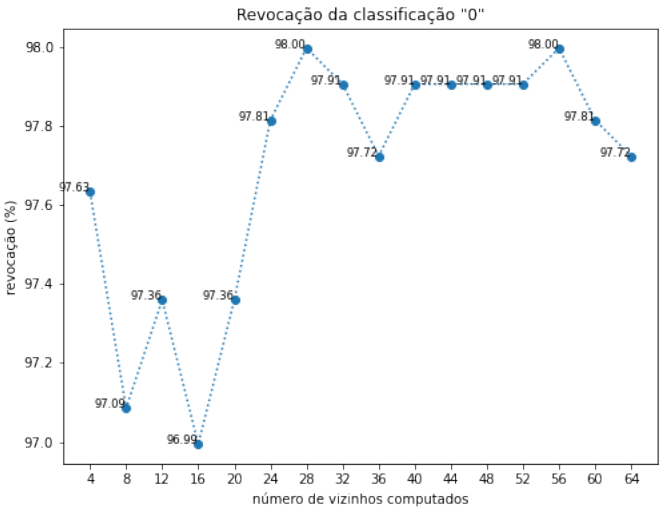

Revocação da classificação "1"
revocação (%)
número de vizinhos computados
53.85 53.85
53.85
51.28 51.28
50.77 50.77 50.77
50.77 50.77 50.77 50.77
47.18 47.18 47.18 47.18

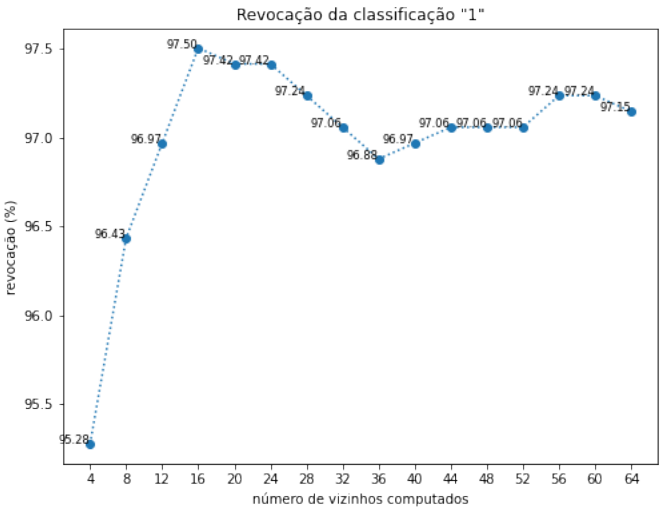
3.10 twonorm.dat






Precisão da classificação "1"
precisão (%)
número de vizinhos computados
98.02
97.93
97.85
97.75
97.63
97.40
97.42
97.13
97.01
97.76


Revocação da classificação "0"
revocação (%)
número de vizinhos computados
98.00
97.91
97.81
97.72
97.63
97.36
97.09
96.99


Revocação da classificação "1"
revocação (%)
número de vizinhos computados
97.50
97.42
97.24
97.06
96.97
96.88
96.43
95.28
97.15