K-NN (K-Nearest Neighbors)

Marcela Ribeiro de Oliveira, GRR20157372

I. EXECUÇÃO

Para compilar basta executar o comando *make*.
Para executar o programa:

```
./knn <treino> <teste> <k>
```

II. IMPLEMENTAÇÃO

A implementação foi feita na linguagem C. Após ter sido lido o arquivo com a base de treino e o arquivo com a base de teste, um vetor de tamanho k é alocado. Este vetor irá armazenar os k pontos mais próximos do ponto que está sendo classificado. Dessa forma, a cada distância calculada, uma função de inserção em vetor ordenado é chamada. Se a quantidade de elementos no vetor é menor que k, ou seja, o vetor não está cheio, a distância calculada é apenas inserida ordenada no vetor. Senão, verifica-se se esta distância é menor que alguma outra já presente no vetor. Se sim, insere-se ordenado essa distância no vetor, e as distâncias que vem depois dela são movidas para a próxima posição do vetor, sendo, a distância que estava na última posição do vetor, descartada. Senão, essa distância é descartada, pois ela é maior que as k distâncias mais próximas.

Após calcular todas as distâncias, tendo o vetor k com as distâncias mais próximas, esse vetor é percorrido, para encontrar a classe para a qual o ponto em análise será classificado, ou verificar que é uma rejeição.

Se o ponto foi classificado corretamente, incrementa-se 1 ao contador de acertos, se o mesmo foi classificado de forma errada, incrementa-se 1 ao contador de erros. Já se houve um rejeição, incrementa-se 1 ao contador de rejeições.

Após esse processo, as taxas de acerto, erro e rejeição são calculadas e exibidas. Logo após é mostrada a matriz de confusão.

III. TESTES

Os testes foram feitos com a base CCtrain utilizada para treino, e as bases CCtest1 e CCtest2 utilizadas para teste. Os valores de k testados foram 1, 2, 3, 5, 6, 7, 10. Os resultados obtidos para cada base de teste são mostrados a seguir.

Observação: A posição [0][0] da matriz de confusão mostra a quantidade de classes encontradas.

A. Base - CCtest1

Para **k=1** a taxa de acerto foi 0.98565%, a taxa de erro foi 0.01435% e a rejeição foi 0.00000%. O tempo de execução foi 24m31.351s.

Abaixo é mostrada a matriz de confusão.

$$\begin{pmatrix}
10 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 5815 & 0 & 1 & 5 & 1 & 2 & 7 & 0 & 32 & 1 \\
1 & 1 & 6475 & 2 & 3 & 1 & 3 & 6 & 0 & 30 & 0 \\
2 & 3 & 44 & 5919 & 27 & 6 & 2 & 0 & 9 & 12 & 0 \\
3 & 5 & 6 & 23 & 5923 & 0 & 55 & 0 & 12 & 10 & 10 \\
4 & 18 & 2 & 1 & 0 & 5821 & 1 & 2 & 4 & 15 & 19 \\
5 & 3 & 1 & 2 & 36 & 0 & 5579 & 18 & 0 & 16 & 2 \\
6 & 19 & 0 & 1 & 0 & 14 & 8 & 5865 & 0 & 12 & 0 \\
7 & 1 & 23 & 10 & 21 & 5 & 2 & 0 & 6200 & 4 & 87 \\
8 & 11 & 6 & 7 & 7 & 2 & 7 & 2 & 0 & 5728 & 5 \\
9 & 17 & 10 & 1 & 14 & 23 & 25 & 0 & 29 & 30 & 5902
\end{pmatrix}$$

Para **k=2** a taxa de acerto foi 0.97750%, a taxa de erro foi 0.00832% e a rejeição foi 0.01418%. O tempo de execução foi 24m25.736s.

Abaixo é mostrada a matriz de confusão.

$$\begin{pmatrix}
10 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 5779 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 17 & 1 \\
1 & 1 & 6454 & 2 & 0 & 1 & 1 & 3 & 0 & 26 & 0 \\
2 & 1 & 42 & 5890 & 13 & 1 & 1 & 0 & 5 & 4 & 0 \\
3 & 2 & 3 & 11 & 5853 & 0 & 36 & 0 & 5 & 4 & 7 \\
4 & 10 & 1 & 0 & 0 & 5798 & 1 & 2 & 3 & 9 & 14 \\
5 & 2 & 1 & 1 & 11 & 0 & 5525 & 7 & 0 & 6 & 0 \\
6 & 13 & 0 & 0 & 0 & 10 & 7 & 5847 & 0 & 6 & 0 \\
7 & 0 & 19 & 6 & 14 & 2 & 0 & 0 & 6160 & 4 & 61 \\
8 & 3 & 1 & 3 & 4 & 1 & 3 & 0 & 0 & 5630 & 0 \\
9 & 8 & 5 & 0 & 9 & 13 & 15 & 0 & 13 & 19 & 5801
\end{pmatrix}$$

Para **k=3** a taxa de acerto foi 0.98203%, a taxa de erro foi 0.01033% e a rejeição foi 0.00764%. O tempo de execução foi 24m20.544s.

Abaixo é mostrada a matriz de confusão.

$$\begin{pmatrix}
10 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 5796 & 0 & 0 & 3 & 1 & 1 & 4 & 0 & 24 & 1 \\
1 & 1 & 6462 & 2 & 2 & 1 & 1 & 4 & 0 & 29 & 0 \\
2 & 1 & 43 & 5907 & 15 & 3 & 1 & 0 & 6 & 8 & 0 \\
3 & 2 & 5 & 12 & 5900 & 0 & 45 & 0 & 7 & 6 & 9 \\
4 & 11 & 1 & 1 & 0 & 5813 & 1 & 2 & 3 & 12 & 16 \\
5 & 2 & 1 & 1 & 18 & 0 & 5557 & 8 & 0 & 8 & 1 \\
6 & 14 & 0 & 1 & 0 & 11 & 7 & 5854 & 0 & 7 & 0 \\
7 & 0 & 22 & 7 & 18 & 4 & 1 & 0 & 6181 & 4 & 71 \\
8 & 6 & 2 & 5 & 4 & 1 & 5 & 0 & 0 & 5684 & 1 \\
9 & 12 & 7 & 0 & 10 & 19 & 19 & 0 & 17 & 23 & 5855
\end{pmatrix}$$

Para **k=5** a taxa de acerto foi 0.98113%, a taxa de erro foi 0.00990% e a rejeição foi 0.00897%. O tempo de execução foi 24m23.489s.

Abaixo é mostrada a matriz de confusão.

$$\begin{pmatrix}
10 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 5797 & 0 & 0 & 3 & 0 & 1 & 5 & 0 & 24 & 1 \\
1 & 1 & 6459 & 1 & 1 & 1 & 1 & 3 & 0 & 29 & 0 \\
2 & 1 & 43 & 5904 & 13 & 3 & 1 & 0 & 4 & 6 & 0 \\
3 & 3 & 3 & 11 & 5893 & 0 & 46 & 0 & 7 & 7 & 9 \\
4 & 10 & 1 & 1 & 0 & 5806 & 1 & 2 & 3 & 12 & 16 \\
5 & 2 & 1 & 1 & 18 & 0 & 5548 & 7 & 0 & 4 & 0 \\
6 & 12 & 0 & 1 & 0 & 12 & 6 & 5860 & 0 & 9 & 0 \\
7 & 0 & 23 & 6 & 17 & 3 & 2 & 0 & 6169 & 4 & 69 \\
8 & 4 & 0 & 4 & 4 & 1 & 5 & 0 & 0 & 5681 & 1 \\
9 & 11 & 7 & 0 & 10 & 17 & 16 & 0 & 20 & 24 & 5838
\end{pmatrix}$$

Para **k=10** a taxa de acerto foi 0.97900%, a taxa de erro foi 0.00935% e a rejeição foi 0.01165%. O tempo de execução foi 26m25.884s.

Abaixo é mostrada a matriz de confusão.

10	0	1	2	3	4	5	6	7	8	9
0	5795	0	0	3	0	0	5	0	21	1
1	0	6454	1	1	1	1	2	0	30	0
2	1	43	5897	13	2	1	0	5	5	0
3	3	4	9	5884	0	42	0	6	5	9
4	9	0	1	0	5802	1	2	3	12	15
5	2	1	1	13	0	5539	7	0	5	0
6	11	0	1	0	12	6	5858	0	6	0
7	0	21	7	17	3	2	0	6144	4	68
8	4	0	4	4	1	4	1	0	5659	1
9	8	8	0	10	16	15	0	19	23	5795

B. Base - CCtest2

Para **k=1** a taxa de acerto foi 0.96295%, a taxa de erro foi 0.03705% e a rejeição foi 0.00000%. O tempo de execução foi 24m44.397s.

Abaixo é mostrada a matriz de confusão.

10	0	1	2	3	4	5	6	7	8	9
0	5513	1	9	4	6	6	24	0	33	7
1	3	6371	10	2	3	3	10	12	19	6
2	3	121	5750	23	7	2	5	61	26	7
3	6	35	48	5658	0	133	0	56	41	82
4	3	51	4	2	5459	0	5	55	22	59
5	1	6	7	51	0	5244	60	3	34	16
6	21	8	3	0	71	33	5708	0	31	1
7	1	36	42	42	29	8	0	5847	17	84
8	8	9	12	18	9	45	45	7	5401	29
9	1	17	3	19	138	65	1	56	71	5522

Para **k=2** a taxa de acerto foi 0.94726%, a taxa de erro foi 0.02234% e a rejeição foi 0.03040%. O tempo de execução foi 24m30.467s.

Abaixo é mostrada a matriz de confusão.

10	0	1	2	3	4	5	6	7	8	9
0	5480	0	7	3	2	3	11	0	24	5
1	1	6302	7	1	1	3	4	4	13	6
2	0	91	5679	14	2	2	2	34	17	5
3	6	20	21	5581	0	86	0	31	17	63
4	2	43	2	0	5371	0	2	33	14	37
5	0	2	0	19	0	5115	29	1	19	7
6	18	4	1	0	58	20	5637	0	16	1
7	1	25	31	33	9	5	0	5738	12	58
8	4	3	8	11	1	19	22	1	5261	16
9	0	11	2	10	84	35	0	28	47	5389

Para **k=3** a taxa de acerto foi 0.95572%, a taxa de erro foi 0.02747% e a rejeição foi 0.01681%. O tempo de execução foi 23m37.694s.

Abaixo é mostrada a matriz de confusão.

10	0	1	2	3	4	5	6	7	8	9
0	5496	0	8	4	2	4	16	0	27	5
1	2	6319	9	1	2	3	7	6	17	6
2	1	102	5726	18	3	2	2	43	20	5
3	6	24	27	5628	0	104	0	38	24	73
4	3	46	2	0	5425	0	3	45	18	43
5	0	2	1	32	0	5196	36	1	22	9
6	19	6	2	0	68	22	5675	0	22	1
7	1	34	40	35	14	6	0	5793	13	69
8	5	4	11	16	2	23	29	1	5341	20
9	0	17	2	15	100	45	0	41	54	5450

Para **k=5** a taxa de acerto foi 0.95333%, a taxa de erro foi 0.02568% e a rejeição foi 0.02099%. O tempo de execução foi 24m49.635s.

Abaixo é mostrada a matriz de confusão.

10	0	1	2	3	4	5	6	7	8	9
0	5496	0	7	2	3	3	14	0	26	4
1	3	6302	8	1	2	3	7	5	14	6
2	2	101	5710	15	4	2	2	42	22	3
3	5	22	26	5619	0	93	0	36	22	71
4	2	48	2	1	5404	0	3	44	16	40
5	0	1	1	27	0	5180	27	0	18	10
6	19	6	2	0	67	20	5673	0	21	1
7	1	32	36	32	9	6	0	5762	15	65
8	4	4	9	11	2	25	28	1	5330	19
9	0	13	2	14	95	42	0	38	51	5433

Para **k=10** a taxa de acerto foi 0.94900%, a taxa de erro foi 0.02464% e a rejeição foi 0.02636%. O tempo de execução foi 24m26.464s.

Abaixo é mostrada a matriz de confusão.

10	0	1	2	3	4	5	6	7	8	9
0	5488	0	7	2	3	3	17	0	24	5
1	2	6280	8	0	1	3	6	7	12	5
2	2	97	5693	14	3	2	1	38	22	4
3	6	23	24	5602	0	96	0	33	21	74
4	2	48	2	1	5379	0	3	41	18	35
5	0	1	1	21	0	5153	28	0	15	9
6	17	6	2	0	64	17	5660	0	18	1
7	1	32	34	31	10	6	0	5715	13	58
8	5	3	7	14	1	24	24	2	5295	17
9	0	14	1	15	87	45	0	36	50	5390