INTRODUÇÃO

O algoritmo K Nearest Neighbor (KNN), utilizado em aprendizado de maquina, é um algoritmo simples e versátil podendo ser amplamente aplicado em diversas áreas como: saúde, finanças, detecção de escrita, reconhecimento de imagens entre outras. É um algoritmo muito utilizado para problemas de classificação e regressão.

Conforme mencionado em aula o KNN conta o número de vizinhos mais próximo para fazer sua classificação, ou seja, primeiro ele calcula a distância entre os vizinhos, depois verifica a quantidade de vizinhos mais próximos e por fim rotula o elemento a ser classificado.

OBJETIVOS

Conforme relatado no moodle o objetivo desse laboratório é analisar o algoritmo do KNN, gerando diferentes vetores de características, variando o valores dos vetores para obter resultados individuais e fazer comparação entre as matrizes de confusão geradas, apresentar a que teve melhor representação analisando as confusões que foram resolvidas.

METODOLOGIA

Os materiais utilizados foram o Google colab, scikti learn e o algoritmo fornecido pelo professor em aula e disponibilizado no moodle. A partir desses dados apenas alteramos alguns parâmetros para chegarmos ao resultado desejado.

Conforme orientado no documento - impactos da representação - a base de dados foi dividida em 50% para treinamento e 50% para validação.

A partir desse ponto foi gerado 3 vetores de características diferentes para experimentos, a acurácia foi calculada para um números de vizinhos variando de 3 até 49, alternou-se também algumas métricas de distância e assim obter as matrizes de confusão.

RESULTADOS E DISCUSSÃO

No primeiro caso analisado foi utilizado um vetor característica com os valores 20 e 10, um k variando de 3 a 49 que será utilizado nos demais experimentos. A seguir apresentamos as matrizes de confusão do melhor e do pior caso, a acurácia para cada caso e um gráfico.

Os valores para as distâncias Euclidiana, Manhattan e Minkowski com p =3, possuem resultados semelhantes (iguais), devido a esse motivo apresento apenas os resultados para a distância Euclidiana evitando ficar repetitivo. O tempo de execução com distância de Minkowski foi de aproximadamente 15 minutos, por esse motivo foi testado apenas para p = 3. Se calculada com a distância de Chebyshev a acurácia se torna muito pequena com o melhor percentual em 17% quando k=9, chegando a 8% quando k = 49, o que torna uma distância inviável para para tal calculo, nesse caso foi utilizado com os vetores de característica com tamanho 50.

Pode-se perceber que quanto maior o K, menor a acurácia, os erros e confusões ficam mais evidentes com as imagens dos números 9 e 7 além do 1 e 2 que aparecem erros com frequência também, percebe-se que para k = 3 temos os melhores resultados, quando utilizamos as distâncias Euclidiana, Manhattan e Minkowski.

Resultados abaixo foram calculados com os vetores X e Y nos valores de tamanho 20 e 10 e distância Euclidiana.

Matriz de confusão:

K = 3	K = 49
92 0 0 0 0 1 0 0 0 0	92 0 0 0 0 1 0 0 0 0
0 111 1 0 0 0 0 0 0 0	0 101 5 1 2 0 1 0 2 0
2 7 96 2 1 0 0 1 1 0	9 22 67 3 0 0 3 4 2 0
0 0 0 88 0 3 0 1 0 0	1 1 1 85 0 2 1 1 0 0
0 6 0 0 92 0 0 1 0 1	0 13 0 0 86 0 0 0 0 1
2 1 0 5 0 79 1 0 0 0	2 1 0 9 0 75 1 0 0 0
3 2 0 0 1 1 93 0 0 0	2 9 0 0 1 0 88 0 0 0
0 5 0 0 0 0 0 94 0 2	3 19 0 1 0 1 0 75 0 2
3 5 0 6 2 1 0 3 76 1	1 10 0 10 2 3 1 10 58 2
0 1 0 0 3 0 0 9 0 94	0 4 0 0 8 0 0 13 0 82

	precision	recall	f1-score	support	precision	recall	f1-score	support
0	0.90	0.99	0.94	93	0.84	0.99	0.91	93
1	0.80	0.99	0.89	112	0.56	0.90	0.69	112
2	0.99	0.87	0.93	110	0.92	0.61	0.73	110
3	0.87	0.96	0.91	92	0.78	0.92	0.85	92
4	0.93	0.92	0.92	100	0.87	0.86	0.86	100
5	0.93	0.90	0.91	88	0.91	0.85	0.88	88
6	0.99	0.93	0.96	100	0.93	0.88	0.90	100
7	0.86	0.93	0.90	101	0.73	0.74	0.74	101
8	0.99	0.78	0.87	97	0.94	0.60	0.73	97
9	0.96	0.88	0.92	107	0.94	0.77	0.85	107

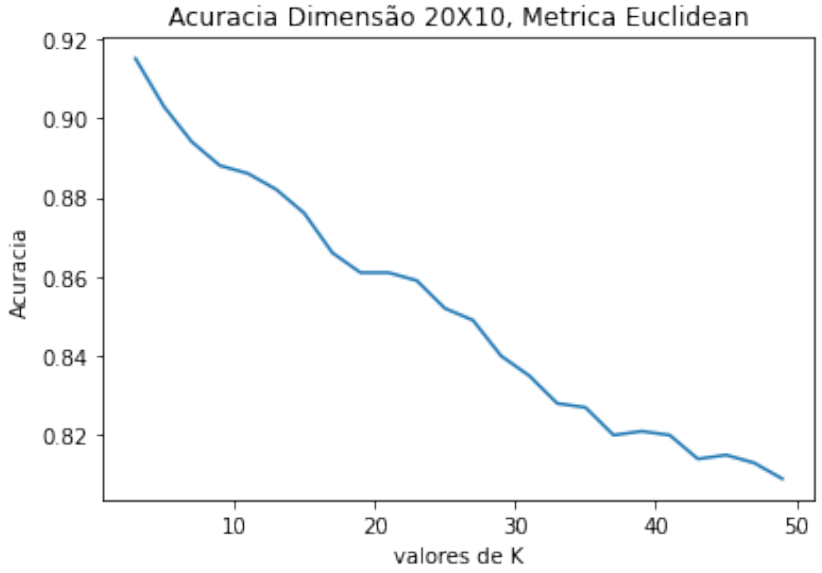
accuracy			0.92	1000	accuracy		0.81	1000
macro avg	0.92	0.91	0.92	1000	macro avg	0.84	0.81	0.81	1000
weighted avg	0.92	0.92	0.92	1000	weighted avg	0.84	0.81	0.81	1000

Vetor com os valores de acurácia para cada valor de K:

[0.915, 0.903, 0.894, 0.888, 0.886, 0.882, 0.876, 0.866, 0.861, 0.861, 0.859, 0.852, 0.849, 0.84, 0.835, 0.828, 0.827, 0.82, 0.821, 0.82, 0.814, 0.815, 0.813, 0.809]

Vetor para os valores de K:

[3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]



Para vetores X e Y de tamanhos 20 e 40, a acurácia tem uma pequena melhora porém quando alteramos as distâncias o comportamento se mantém semelhante conforme descrito anteriormente.

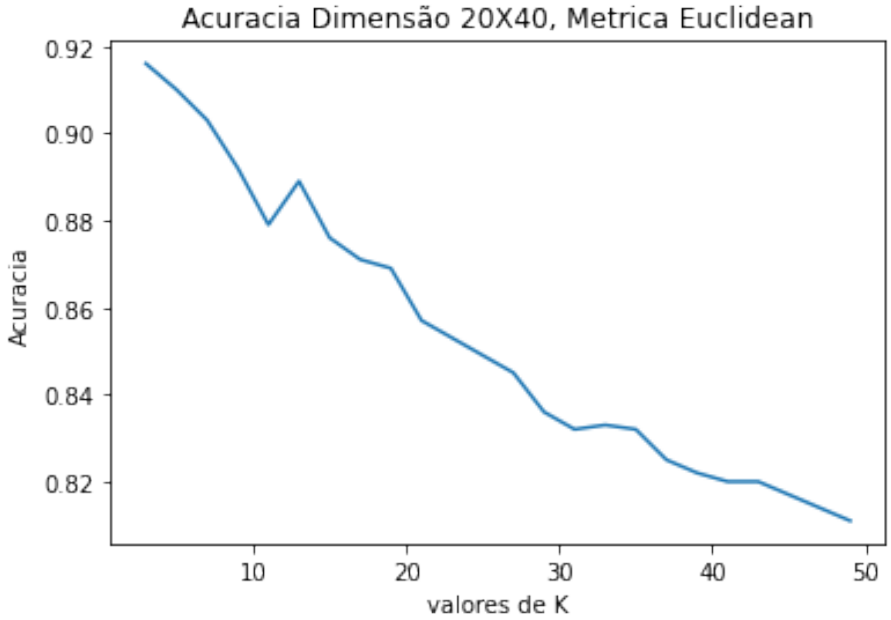
K = 3	K = 49
91 0 0 0 0 1 1 0 0 0 0 110 2 0 0 0 0 0 0 0 2 5 98 0 1 0 0 2 2 0 0 0 0 89 0 2 0 0 1 0 0 8 0 0 91 0 0 0 0 1 1 0 0 5 0 81 1 0 0 0 1 1 0 0 1 1 96 0 0 0 0 6 0 0 0 0 0 92 0 3 3 6 0 5 1 2 0 2 75 3 0 0 0 0 4 0 0 10 0 93	92 0 0 0 0 1 0 0 0 0 0 101 6 1 1 0 1 0 2 0 9 17 71 3 0 0 3 5 2 0 0 1 1 85 0 2 1 2 0 0 0 14 0 0 84 0 0 0 0 2 2 1 0 8 0 76 1 0 0 0 5 8 0 0 1 0 86 0 0 0 2 20 0 0 0 1 0 76 0 2 1 8 0 10 1 7 1 7 61 1 0 7 0 0 6 0 0 15 0 79
precision recall f1-score support 0 0.93 0.98 0.95 93 1 0.81 0.98 0.89 112 2 0.98 0.89 0.93 110 3 0.90 0.97 0.93 92 4 0.93 0.91 0.92 100 5 0.93 0.92 0.93 88 6 0.98 0.96 0.97 100 7 0.87 0.91 0.89 101 8 0.96 0.77 0.86 97 9 0.93 0.87 0.90 107	precision recall f1-score support 0.93 0.98 0.95 93 0.81 0.98 0.89 112 0.98 0.89 0.93 110 0.90 0.97 0.93 92 0.93 0.91 0.92 100 0.93 0.92 0.93 88 0.98 0.96 0.97 100 0.87 0.91 0.89 101 0.96 0.77 0.86 97 0.93 0.87 0.90 107
accuracy 0.92 1000 macro avg 0.92 0.92 0.92 1000 weighted avg 0.92 0.92 0.92 1000	accurac y 0.81 1000 macro avg 0.84 0.81 0.82 1000 weighted avg 0.84 0.81 0.81 1000

Vetor com os valores de acurácia para cada valor de K:

[0.916, 0.91, 0.903, 0.892, 0.879, 0.889, 0.876, 0.871, 0.869, 0.857, 0.853, 0.849, 0.845, 0.836, 0.832, 0.833, 0.832, 0.825, 0.822, 0.82, 0.82, 0.817, 0.814, 0.811]

Vetor para os valores de K:

[3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]



K = 3	K = 49
91 0 0 0 0 1 1 0 0 0	92 0 0 0 0 1 0 0 0 0
0 110 2 0 0 0 0 0 0 0	0 100 7 1 1 0 1 0 2 0
1 5 99 0 1 0 0 2 2 0	6 18 73 2 0 0 3 5 3 0
0 1 0 89 0 1 0 0 1 0	1 1 1 84 0 2 1 2 0 0
0 8 0 0 91 0 0 0 0 1	0 14 0 0 83 0 0 0 0 3
1 0 0 5 0 81 1 0 0 0	2 1 0 8 0 76 1 0 0 0
3 2 0 0 1 1 93 0 0 0	3 9 0 0 1 1 86 0 0 0
0 3 0 0 0 0 0 96 0 2	3 21 0 0 0 1 0 74 0 2
3 4 0 5 1 2 0 2 77 3	1 10 0 11 2 6 1 8 56 2
0 3 0 0 3 0 0 9 0 92	0 6 0 0 6 0 0 20 0 75

	precision	recall	f1-score	support
0	0.92	0.98	0.95	93
1	0.81	0.98	0.89	112
2	0.98	0.90	0.94	110

precision	recall	f1-score	support
0.85	0.99	0.92	93
0.56	0.89	0.68	112
0.90	0.66	0.76	110

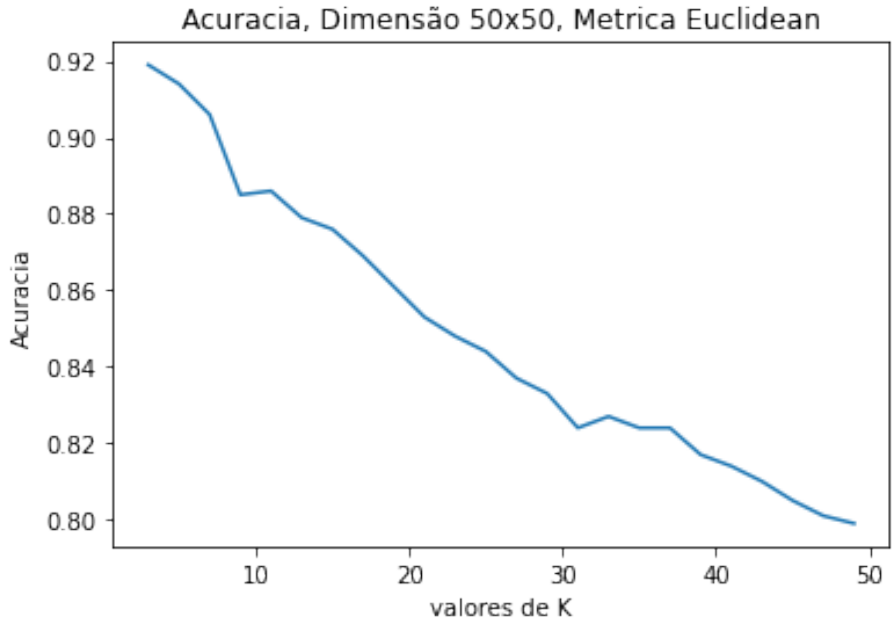
3	0.90	0.97	0.93	92		0.79	0.91	0.85	92
4	0.94	0.91	0.92	100		0.89	0.83	0.86	100
5	0.94	0.92	0.93	88		0.87	0.86	0.87	88
6	0.98	0.93	0.95	100		0.92	0.86	0.89	100
7	0.88	0.95	0.91	101		0.68	0.73	0.70	101
8	0.96	0.79	0.87	97		0.92	0.58	0.71	97
9	0.94	0.86	0.90	107		0.91	0.70	0.79	107
accuracy			0.92	1000		accuracy		0.80	1000
macro avg	0.92	0.92	0.92	1000		macro avg	0.83	0.80	0.80	1000
weighted avg	0.92	0.92	0.92	1000		weighted avg	0.83	0.80	0.80	1000

Valores de K:

[3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]

Valores da acurácia:

[0.919, 0.914, 0.906, 0.885, 0.886, 0.879, 0.876, 0.869, 0.861, 0.853, 0.848, 0.844, 0.837, 0.833, 0.824, 0.827, 0.824, 0.824, 0.817, 0.814, 0.81, 0.805, 0.801, 0.799]



CONCLUSÃO

É possível perceber que a acurácia não se altera para o problema proposto quando alteramos apenas a métrica de distância porém as métricas utilizadas são muito semelhantes, euclidiana, manhattan e mikowsk. A métrica de distância de Chebychev foi a única com resultados diferentes de forma negativa, pois aumentou a quantidade de confusões.

As confusões mais frequentes, conforme mostrado pelas matrizes de confusão, foram com as imagens do 1, 2, 7 e 9, os demais números tiveram uma boa acurácia tendo em vista a baixa quantidade de confusão, entregando um bom resultado.

Foram feitas varias alterações, porém os resultados mais consistentes pertenceram as três métricas euclidiana, manhattan e mikowsk, com vantagem para euclidiana pela velocidade de execução. Quanto mais aumenta a dimensão dos vetores mais lento se torna a obtenção de resultados, além disso chega em momento que o resultado deixa de ser tão expressivo, o que torna algo inviável.