Discente: Deivison Venicio Souza

Laboratório Aprendizagem de Máquina

Lab1: Impactos da Representação

O script **digits.py** extrai a representação mais simples possível de uma base de dados dígitos manuscritos. Para cada posição da imagem, verifica-se o valor de intensidade do **pixel e se esse valor for > 128**, **a característica é igual a 1, caso contrário 0**. As imagens tem tamanho variável e como os classificadores precisam de um vetor de tamanho fixo, as imagens são **normalizadas** utilizando as **variáveis X e Y** dentro da função **rawpixel**. Após a execução do programa, um arquivo chamado **features.txt** é criado no diretório corrente. Esse arquivo contem 2000 linhas no formato:

0 0:0 1:0 2:1 3:1

O primeiro caractere indica o rótulo da classe. A sequencia i:v indica o índice da característica e o valor da mesma. Nesse caso, a características 0, 1, 2, e 3 tem valores 0, 0, 1 e 1, respectivamente.

Sua tarefa consiste em **gerar diferentes vetores de características variando os valores de X e Y.** Utilizando um kNN (k=3 e distância Euclidiana), e**ncontre o conjunto de características que produziu os piores e melhores resultados de classificação.** A base de dados deve ser dividida em 50% para treinamento e 50% para validação. Compare as matrizes de confusão nesses dois casos e reporte quais foram as confusões resolvidas pela melhor representação. Para a sua melhor solução, verifique se é possível melhorar o resultados mudando os valores de k e métrica de distância.

A) Fixando k = 3 e distância = Euclidiana

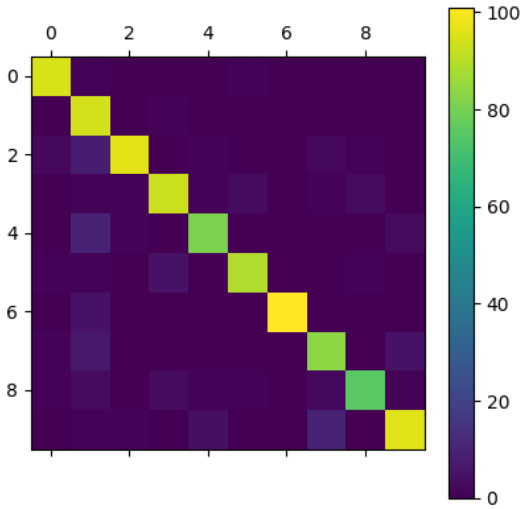
1^0 TESTE: Extraindo vetores de características fixando dimensões para imagem de X = 20 e Y = 10, e usando k=3 e distância Euclidiana.

Dimensão 1:
x = 20
y = 10

Acurácia: 0.905

Matriz de Confusão:

	Real									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	[[95	1	0	0	0	1	0	0	0	0]
1	[0	94	0	1	0	0	0	0	0	0]
2	[2	8	97	0	1	0	0	2	1	0]
3	[0	1	1	93	1	3	0	1	3	0]
4	[0	10	1	0	81	0	0	0	0	3]
5	[1	1	0	5	0	89	0	0	1	0]
6	[0	5	0	0	0	0	101	0	0	0]
7	[1	7	0	0	0	0	0	84	0	5]



8 [1 3 0 3 1 1 0 2 75 1]
9 [0 1 1 0 4 0 0 10 0 96]]

2^0 TESTE: Extraindo vetores de características fixando dimensões para imagem de X = 30 e Y = 20, e usando k=3 e distância Euclidiana.

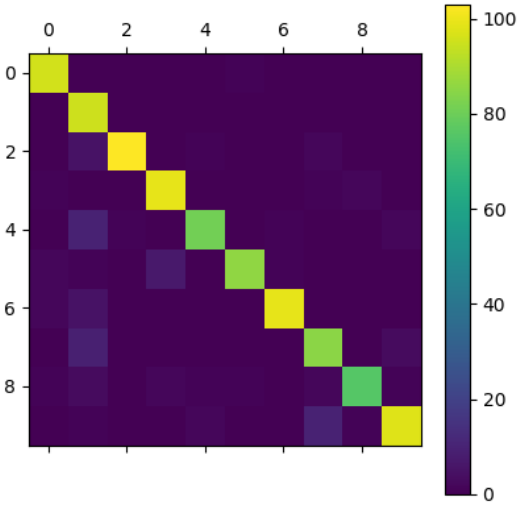
Dimensão 2:
x = 30
y = 20

Acurácia: 0.918

Matriz de Confusão:

Real

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
[[	96	0	0	0	0	1	0	0	0	0]
[	0	95	0	0	0	0	0	0	0	0]
[	0	5	103	0	1	0	0	2	0	0]
[	1	0	0	99	0	0	0	1	2	0]
[	0	10	1	0	81	0	1	0	0	2]
[	2	1	0	7	0	86	1	0	0	0]
[	2	5	0	0	0	0	99	0	0	0]
[	0	9	0	0	0	0	0	85	0	3]
[	1	3	0	2	1	1	0	2	76	1]
[	0	1	0	0	2	0	0	10	1	98]]



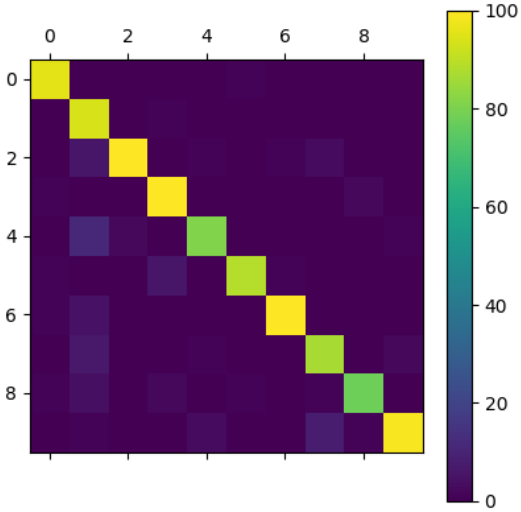
3^0 TESTE: Extraindo vetores de características fixando dimensões para imagem de X = 28 e Y = 28, e usando k=3 e distância Euclidiana.

Dimensão 3:
x = 28
y = 28

Acurácia: 0.924

Matriz de Confusão:

[[	96	0	0	0	0	1	0	0	0	0]
[	0	94	0	1	0	0	0	0	0	0]
[	0	6	100	0	1	0	1	3	0	0]
[	1	0	0	100	0	0	0	0	2	0]
[	0	11	2	0	81	0	0	0	0	1]
[	1	0	0	6	0	89	1	0	0	0]
[	1	5	0	0	0	0	100	0	0	0]
[	0	7	0	0	1	0	0	87	0	2]



[1 4 0 2 0 1 0 1 78 0]
[0 1 0 0 3 0 0 8 1 99]]

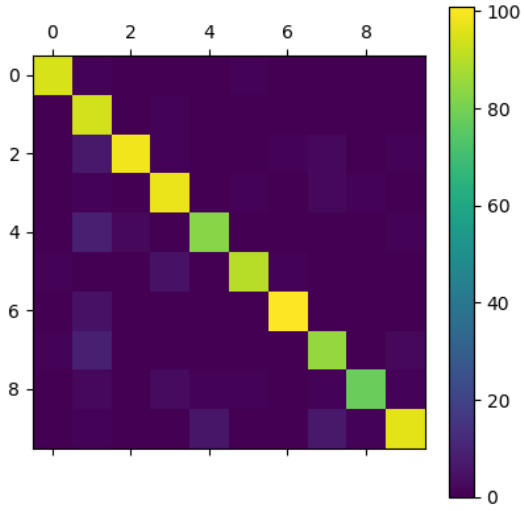
4^0 TESTE: Extraindo vetores de características fixando dimensões para imagem de X = 35 e Y = 35, e usando k=3 e distância Euclidiana.

Dimensão 4:
x = 35
y = 35

Acurácia: 0.92

Matriz de Confusão:

[[95 1 0 0 0 1 0 0 0 0]
 [0 94 0 1 0 0 0 0 0 0]
 [0 7 99 1 0 0 1 2 0 1]
 [0 1 0 98 0 1 0 2 1 0]
 [0 9 2 0 83 0 0 0 0 1]
 [1 0 0 5 0 90 1 0 0 0]
 [0 5 0 0 0 0 101 0 0 0]
 [1 9 0 0 0 0 0 85 0 2]
 [0 2 0 3 1 1 0 1 78 1]
 [0 1 0 0 6 0 0 7 1 97]]



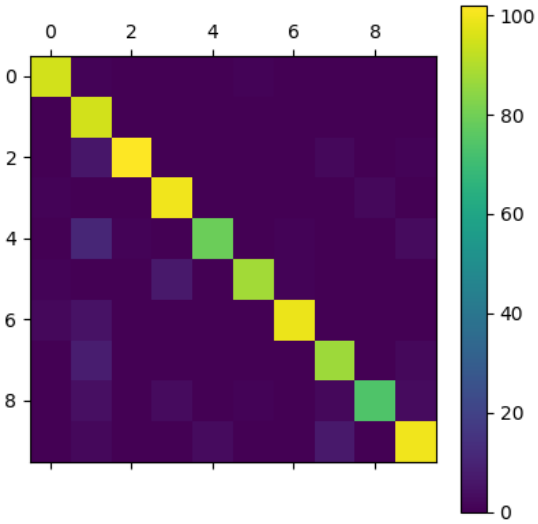
5^0 TESTE: Extraindo vetores de características fixando dimensões para imagem de X = 28 e Y = 20, e usando k=3 e distância Euclidiana.

Dimensão 5:
x = 28
y = 20

Acurácia: 0.919

Matriz de Confusão:

[[95 1 0 0 0 1 0 0 0 0]
 [0 95 0 0 0 0 0 0 0 0]
 [0 6 102 0 0 0 0 2 0 1]
 [1 0 0 100 0 0 0 0 2 0]
 [0 11 1 0 79 0 1 0 0 3]
 [1 0 0 7 0 88 1 0 0 0]
 [2 5 0 0 0 0 99 0 0 0]
 [0 8 0 0 0 0 0 87 0 2]
 [0 4 0 3 0 1 0 2 74 3]
 [0 2 0 0 3 0 0 7 0 100]]



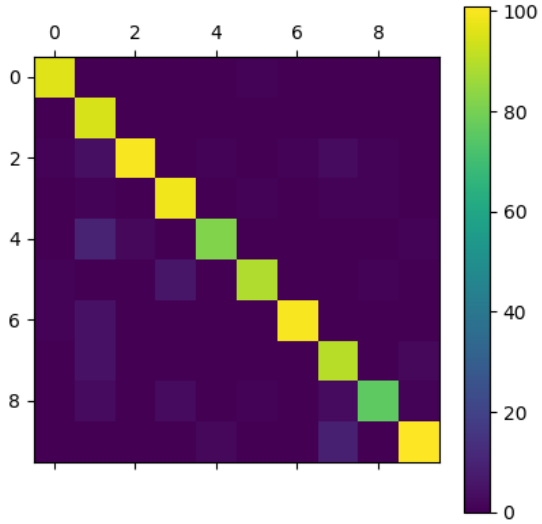
6⁰ TESTE: Extraindo vetores de características fixando dimensões para imagem de X = 28 e Y = 25, e usando k=3 e distância Euclidiana.

Dimensão 6:
x = 28
y = 25

Acurácia: 0.928

Matriz de Confusão:

```
[[ 96   0   0   0   0   1   0   0   0   0]
 [  0  95   0   0   0   0   0   0   0   0]
 [  1   4 100   0   1   0   1   3   1   0]
 [  0   1   0  99   0   1   0   1   1   0]
 [  0  10   2   0  82   0   0   0   0   1]
 [  1   0   0   6   0  89   0   0   1   0]
 [  1   5   0   0   0   0 100   0   0   0]
 [  0   5   0   0   0   0   0  90   0   2]
 [  0   3   0   3   0   1   0   3  76   1]
 [  0   0   0   0   2   0   0   9   0 101]]
```



Considerações: O primeiro experimento do trabalho foi variar as dimensões da imagem dentro do arquivo **digits.py,** fiixando k = 3 e distância = Euclidiana e, em seguida verificar a acurácia do classificador tendo em em vistas as diferentes representações das imagens (vetores de características) obtidas com tal variação. Foram testadas 6 variações (X=20, Y=10), (X=30, Y=20), (X=28, Y=28), (X=35, Y=35), (X=28, Y=20) e (X=28, Y=25). Para cada variação é apresentada a acurácia do classificador kNN e a matriz de confusão. Quando usadas imagens de dimensões (X=20, Y=10) teve-se uma acurácia de 0.905, e percebeu-se que a maior dificuldade do classificador KNN foi de diferenciar o **dígito 1,** havendo confusão com todos os outros dígitos, sobretudo com os dígitos 2, 4 e 7. Na segunda variação, os vetores X=30 e Y=20 foram aumentados em 10 unidades cada para perceber o impacto sobre as predições. Nesta variação observou-se que o dígito 1 deixou de ser confundido com os dígitos 0 e 3, porém os acertos totais na classificação do dígito 1 e acurácia total (0.918) pouco modificaram. Em termos gerais, os tamanhos de imagens testados não foram capazes de melhorar substancialmente o desempenho do classificador KNN, usando-se k=3 e distancia Euclidiana. A maior dificuldade do classificador está em distinguir o dígito 1. Por fim, dentro todas as variações testadas a dimensão (X=28, Y=25) foi aquele com maior acurácia total = 0,928. A seguir são testadas algumas variações dos parâmetros de tuning do KNN, isto é, variações de k e diferentes tipos de métrica de distância.

B) Variando k = (5, 7, 9, 11) e fixando a distância = Euclidiana e tamanho (X=28, Y=25)

1⁰ TESTE: Classificando no conjunto de teste usando k=5

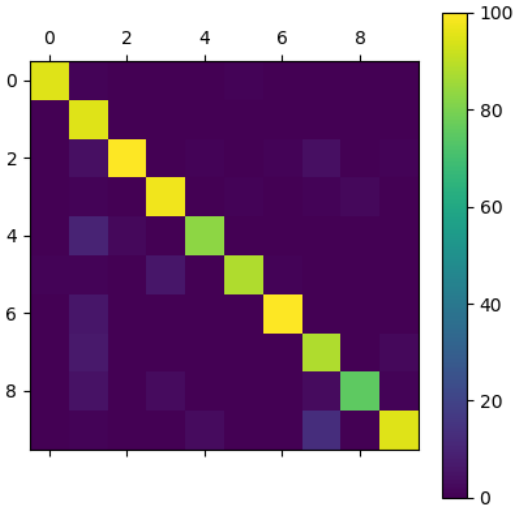
---

k = 5

Acurácia: 0.917

Matriz de Confusão:

```
[[ 95   1   0   0   0   1   0   0   0   0]
 [  0  95   0   0   0   0   0   0   0   0]
 [  0   4 100   0   1   0   1   4   0   1]
 [  0   1   0  98   0   1   0   1   2   0]
 [  0  10   2   0  83   0   0   0   0   0]
 [  1   1   0   6   0  88   1   0   0   0]
 [  0   6   0   0   0   0 100   0   0   0]
 [  0   7   0   0   0   0   0  88   0   2]
 [  0   5   0   3   0   0   0   3  75   1]
 [  0   1   0   0   3   0   0  13   0  95]]
```



2⁰ TESTE: Classificando no conjunto de teste usando k=7

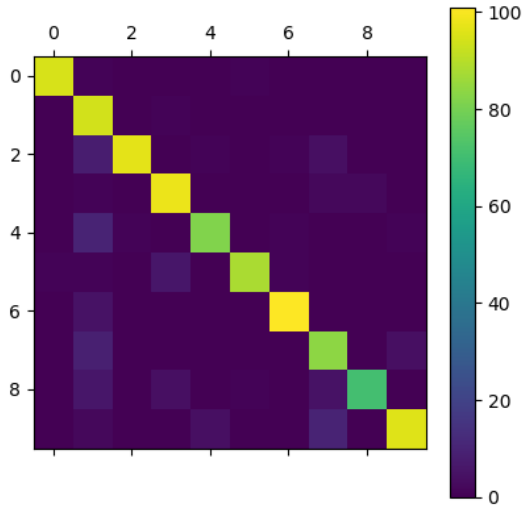
---

k = 7

Acurácia: 0.906

Matriz de Confusão:

```
[[ 95   1   0   0   0   1   0   0   0   0]
 [  0  94   0   1   0   0   0   0   0   0]
 [  0   8  97   0   1   0   1   4   0   0]
 [  0   1   0  98   0   0   0   2   2   0]
 [  0  10   1   0  82   0   1   0   0   1]
 [  1   1   0   6   0  88   1   0   0   0]
 [  0   5   0   0   0   0 101   0   0   0]
 [  0   9   0   0   0   0   0  84   0   4]
 [  0   6   0   4   0   1   0   5  71   0]
 [  0   2   0   0   4   0   0  10   0  96]]
```



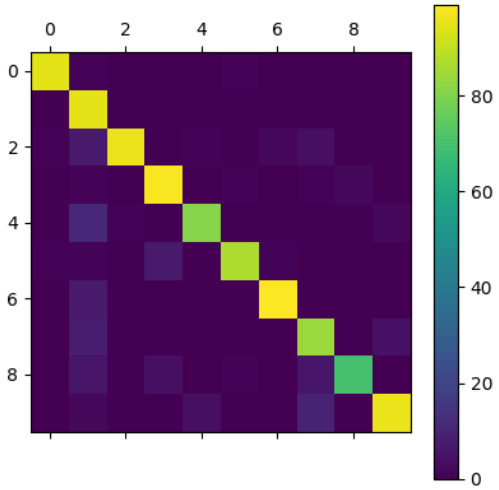
3⁰ TESTE: Classificando no conjunto de teste usando k=9

k = 9

Acurácia: 0.901

Matriz de Confusão:

[[95 1 0 0 0 1 0 0 0 0]
[0 95 0 0 0 0 0 0 0 0]
[1 7 96 0 1 0 2 4 0 0]
[0 1 0 98 0 1 0 1 2 0]
[0 11 1 0 81 0 0 0 0 2]
[1 1 0 7 0 87 1 0 0 0]
[0 7 0 0 0 0 99 0 0 0]
[0 8 0 0 0 0 0 84 0 5]
[0 6 0 4 0 1 0 6 70 0]
[0 2 0 0 4 0 0 10 0 96]]



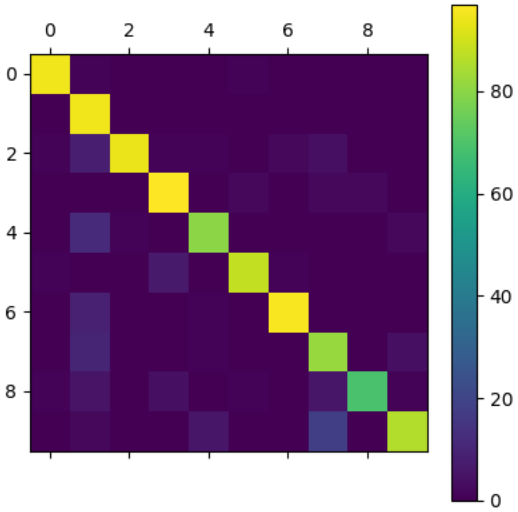
3⁰ TESTE: Classificando no conjunto de teste usando k=11

k = 11

Acurácia: 0.882

Matriz de Confusão:

[[95 1 0 0 0 1 0 0 0 0]
[0 95 0 0 0 0 0 0 0 0]
[1 8 94 1 1 0 2 4 0 0]
[0 0 0 97 0 2 0 2 2 0]
[0 12 1 0 80 0 0 0 0 2]
[1 0 0 7 0 88 1 0 0 0]
[0 9 0 0 1 0 96 0 0 0]
[0 10 0 0 1 0 0 82 0 4]
[1 5 0 4 0 1 0 6 69 1]
[0 2 0 0 6 0 0 18 0 86]]



Considerações: O segundo experimento do trabalho foi variar o número de vizinhos mais próximos k (5, 7, 9 e 11), utilizando a distância Euclidiana e fixando o tamanho das imagens em X= 28 e Y =25, para extração de características (esse foi o tamanho que gerou maior acurácia total no experimento 1). Em termos gerais, o aumento da quantidade de vizinhos mais próximos (k) a serem considerados para classificação de uma nova instância não promoveu melhorias significativas na acurácia do modelo. Assim, k = 3 continuou sendo o melhor valor a ser considerado. No entanto, o ideal é definir um vetor de k e usar validação cruzada (ou outro método de reamostragem) para a tomada de decisão correta, e evitar possível overfitting.

C) Variando a distância (Manhattan e Chebyshev), porém mantendo k = 3 e tamanho (X=28, Y=25)

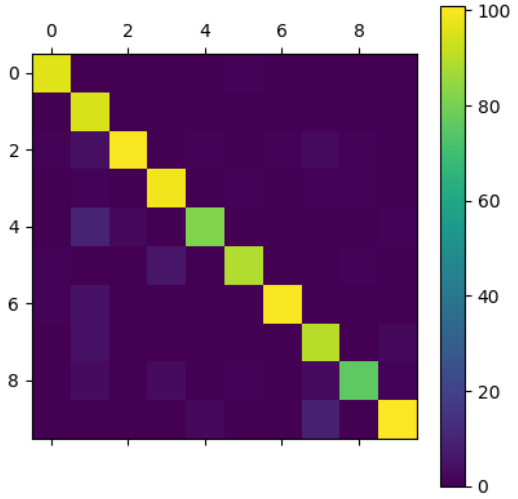
1⁰ TESTE: Classificando no conjunto de teste usando distância de Manhattan.

Distância = manhattan

Acurácia: 0.928

Matriz de Confusão:

```
[[ 96   0   0   0   0   1   0   0   0   0]
 [  0  95   0   0   0   0   0   0   0   0]
 [  1   4 100   0   1   0   1   3   1   0]
 [  0   1   0  99   0   1   0   1   1   0]
 [  0  10   2   0  82   0   0   0   0   1]
 [  1   0   0   6   0  89   0   0   1   0]
 [  1   5   0   0   0   0 100   0   0   0]
 [  0   5   0   0   0   0   0  90   0   2]
 [  0   3   0   3   0   1   0   3  76   1]
 [  0   0   0   0   2   0   0   9   0 101]]
```



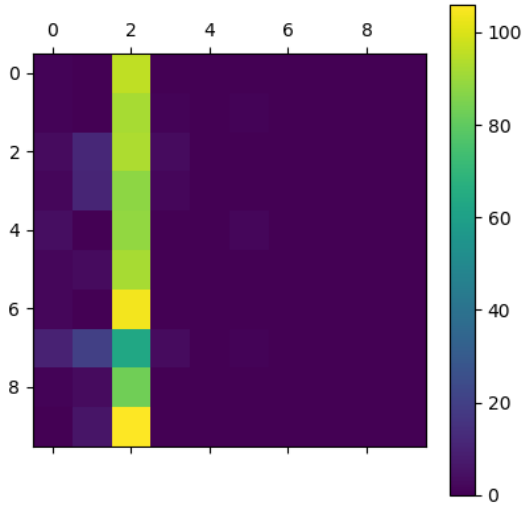
2⁰ TESTE: Classificando no conjunto de teste usando distância de Chebyshev.

Distância = chebyshev

Acurácia: 0.096

Matriz de Confusão:

```
[[  1   0  96   0   0   0   0   0   0   0]
 [  1   0  92   1   0   1   0   0   0   0]
 [  3  12  93   3   0   0   0   0   0   0]
 [  2  11  88   2   0   0   0   0   0   0]
 [  4   0  89   0   0   2   0   0   0   0]
 [  2   3  92   0   0   0   0   0   0   0]
 [  2   0 104   0   0   0   0   0   0   0]
 [ 10  20  63   3   0   1   0   0   0   0]
 [  1   3  83   0   0   0   0   0   0   0]
 [  0   6 106   0   0   0   0   0   0   0]]
```



Considerações: O terceiro experimento do trabalho consistiu em variar o tipo de métrica de distância para Manhattan e Chebyshev, mantendo k = 3 e tamanho das imagens em X= 28 e Y =25, para extração de características. Em geral, a mudança de métrica de similaridade não promoveu melhores na classificação dos dígitos. A métrica de Chebyshev mostrou-se totalmente inviável enquanto medida de similaridade para este caso. Por fim, a melhor performance do classificador KNN foi constatada quando utilizou-se de k = 3, métrica Euclidiana ou Manhattan e tamanho de imagens de X= 28 e Y =25, para extração de características.