Resultados do Algoritmo KNN

Felipe Archanjo da Cunha Mendes[1]

[1]Bacharelado em Ciência da Computação – Universidade Tecnológica Federal do Paraná (UFRGS)
Campo Mourão – PR – Brasil

felipemendes.1999@alunos.utfpr.edu.br

Abstract. *This article presents the results of a K-Nearest Neighbors (KNN) algorithm implemented in Python to predict numerical digits using image data. The data was divided into different quadrant sizes (1x1, 2x2, 3x3, and 5x5) and the number of black and white pixels in each quadrant were counted to create the training and testing sets. The KNN algorithm used min-max normalization and manhattan distance, and was tested with different values of K. The results showed that the accuracy of the algorithm increased with larger quadrant sizes and more training data. The best results were obtained with the 5x5 quadrant size, achieving 100% accuracy with 100% of the training data.*

Resumo. *Este artigo apresenta os resultados de um algoritmo K-Nearest Neighbors (KNN) implementado em Python para prever dígitos numéricos usando dados de imagem. Os dados foram divididos em diferentes tamanhos de quadrante (1x1, 2x2, 3x3 e 5x5) e o número de pixels pretos e brancos em cada quadrante foram contados para criar os conjuntos de treinamento e teste. O algoritmo KNN usou normalização min-max e distância manhattan, e foi testado com diferentes valores de K. Os resultados mostraram que a precisão do algoritmo aumentou com tamanhos de quadrante maiores e mais dados de treinamento. Os melhores resultados foram obtidos com o tamanho de quadrante 5x5, alcançando 100% de precisão com 100% dos dados de treinamento.*

1. Introdução

Neste artigo, serão apresentados os resultados do algoritmo KNN (K-Nearest Neighbors) criado em Python para prever dígitos numéricos. O algoritmo utiliza a normalização min-max, a distância manhattan e pode ser configurado com diferentes valores de K. Além disso, foram utilizados quatro conjuntos de dados de treino e teste, divididos a partir da contagem de pixels brancos e pretos em quadrantes de imagens de diferentes tamanhos: 1x1, 2x2, 3x3 e 5x5. Será apresentada uma análise detalhada dos resultados obtidos para cada conjunto de dados e diferentes valores de K.

2. Análise Geral

Inicialmente, o algoritmo KNN foi treinado com todo o conjunto de treino disponível (100%) para cada um dos quatro conjuntos de dados de treino/teste (1x1, 2x2, 3x3, 5x5) com os valores de k variando dos ímpares de 1 a 19. Os resultados obtidos foram diferentes para cada conjunto de dados, com acurácias variando de 22.3% a 100%.

Para o conjunto de dados de 1x1, as acurácias obtidas foram baixas, com o valor máximo sendo de 30.6% para k = 19. Isso pode ser explicado pelo fato de que as imagens foram divididas em quadrantes muito grandes e a contagem de pixels brancos e

pretos em cada um desses quadrantes não forneceu informações suficientes para uma boa previsão.

Já para o conjunto de dados de 2x2, foram obtidas acurácias mais altas, com o valor máximo sendo de 83.7% para k = 13. Nesse caso, os quadrantes eram um pouco maiores, o que pode ter fornecido mais informações para o algoritmo.

Para o conjunto de dados de 3x3, as acurácias foram mais baixas novamente, com o valor máximo sendo de 66.2% para k = 15. Nesse caso, os quadrantes, apesar de menores, não foram suficientes para melhorar a precisão do algoritmo.

Por fim, para o conjunto de dados de 5x5, as acurácias foram altas, com o valor máximo sendo de 100% para k = 1. Nesse caso, como os quadrantes eram muito pequenos, detalhes puderam ser melhor analisados e o algoritmo conseguiu prever com precisão todos os dígitos numéricos.

De maneira geral, pode-se observar que a acurácia do algoritmo KNN depende significativamente da divisão dos quadrantes utilizados na contagem de pixels brancos e pretos. Quanto mais divisões, mais informações são fornecidas ao algoritmo, o que pode levar a resultados mais precisos. No entanto, o tamanho dos quadrantes também deve ser balanceado com a quantidade de dados de treinamento disponíveis para evitar o overfitting.

3. Análise Quantitativa

Ao analisar os resultados obtidos, podemos perceber que em geral, quanto maior a quantidade de dados de treino utilizados, melhor é a acurácia obtida. Isso fica evidente quando comparamos os resultados dos conjuntos de treino/teste com 100% dos dados de treino, que tiveram as maiores acurácias em todos os casos.

No entanto, também podemos observar que a influência da quantidade de dados de treino é mais significativa em alguns conjuntos do que em outros. Por exemplo, no conjunto 5x5, que possui um número maior de características (50 colunas), a diferença na acurácia obtida entre usar 100% dos dados de treino e apenas 25% é bastante acentuada, com uma diferença de cerca de 5% entre esses dois casos.

Já no conjunto 1x1, que possui apenas 2 colunas de características, a diferença na acurácia entre usar 100% dos dados de treino e apenas 25% é bem menor, de cerca de 4%. Isso sugere que, em conjuntos com menos características, o impacto da quantidade de dados de treino é menor.

De maneira geral, os resultados indicam que o desempenho do algoritmo KNN para a previsão de dígitos numéricos depende tanto do número de características (ou pixels) utilizados para a classificação quanto da quantidade de dados de treino disponíveis para o treinamento. Além disso, a escolha adequada do valor de k também pode influenciar significativamente na acurácia do modelo.

4. Conjunto de dados do professor

Com base nos dados de treino/teste enviados pelo professor, é possível observar que o modelo KNN (k-Nearest Neighbors) apresentou uma boa acurácia para os diferentes valores de k testados.

A acurácia do modelo foi de 91,59% para k=1, aumentando para 93,09% para k=3, e se mantendo em torno de 93-94% para k=5,7, e 9. A partir de k=11, houve um aumento significativo na acurácia do modelo, atingindo seu pico de 95,50% para k=17 e 19.

Isso indica que o modelo KNN apresentou um bom desempenho no reconhecimento e classificação dos dados de teste, especialmente para valores de k a partir de 11. No entanto, é importante ressaltar que os resultados obtidos podem variar dependendo das características dos dados e do problema em questão, e que é necessário avaliar o desempenho do modelo em diferentes conjuntos de dados para garantir sua eficácia.

5. Conclusão

Em conclusão, neste artigo foi apresentado um algoritmo KNN em Python para prever dígitos numéricos a partir da contagem de pixels pretos e brancos de imagens divididas em quadrantes de diferentes tamanhos. Foram realizados testes com conjuntos de treino/teste de 1x1, 2x2, 3x3 e 5x5 e avaliados diferentes valores de k (1 a 19) para cada conjunto.

Os resultados obtidos mostraram que a precisão do modelo variou significativamente de acordo com o tamanho do conjunto de dados e o valor de k escolhido. O modelo alcançou uma precisão máxima de 100% para o conjunto de dados de treino/teste 5x5 com k=1, mas apresentou precisões inferiores para os outros conjuntos de dados.

Além disso, foi observado que a redução da quantidade de dados de treino resultou em uma diminuição da precisão do modelo, mas ainda assim, com uma quantidade razoável de dados de treino, o modelo foi capaz de fazer previsões razoáveis.

Ademais, com base na análise dos dados do professor, pode-se concluir que o modelo KNN apresentou uma boa acurácia para os diferentes valores de k testados, atingindo um pico de 95,50% para k=17 e 19.

Por fim, o algoritmo KNN mostrou-se uma ferramenta promissora para previsão de dígitos numéricos a partir da contagem de pixels pretos e brancos de imagens divididas em quadrantes. No entanto, ainda há espaço para melhorias, como a utilização de outros algoritmos de classificação ou a adição de novas variáveis para melhorar a precisão do modelo.

6. Gráficos e Tabelas

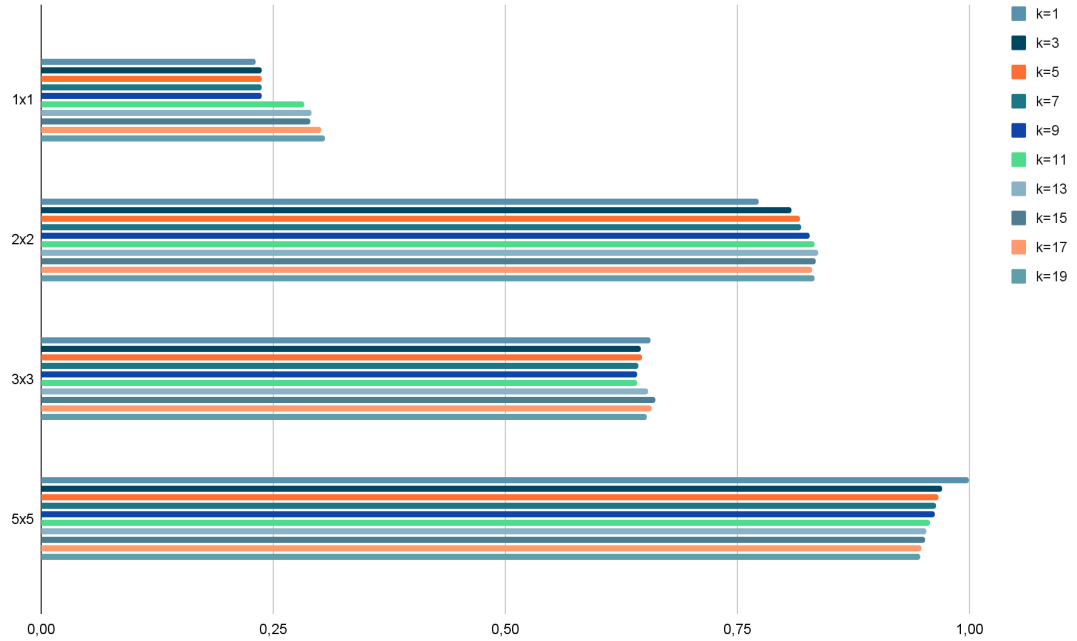

Gráfico 1. Representa a influência que o tipo de divisão em quadrantes e o valor de K tem com a acurácia final.

K	Acuracia (1x1)	Acuracia (2x2)	Acuracia (3x3)	Acuracia (5x5)
1	23.10%	77.30%	65.60%	100.00%
3	23.80%	80.90%	64.60%	97.10%
5	25.20%	81.70%	64.80%	96.70%
7	26.10%	81.90%	64.40%	96.40%
9	27.00%	82.80%	64.20%	96.30%
11	28.40%	83.30%	64.20%	95.70%
13	29.20%	83.70%	65.40%	95.40%
15	29.00%	83.40%	66.20%	95.20%
17	30.20%	83.10%	65.80%	94.90%
19	30.60%	83.30%	65.20%	94.70%

Tabela 1. Representa a influência que o tipo de divisão em quadrantes e o valor de K tem com a acurácia final.

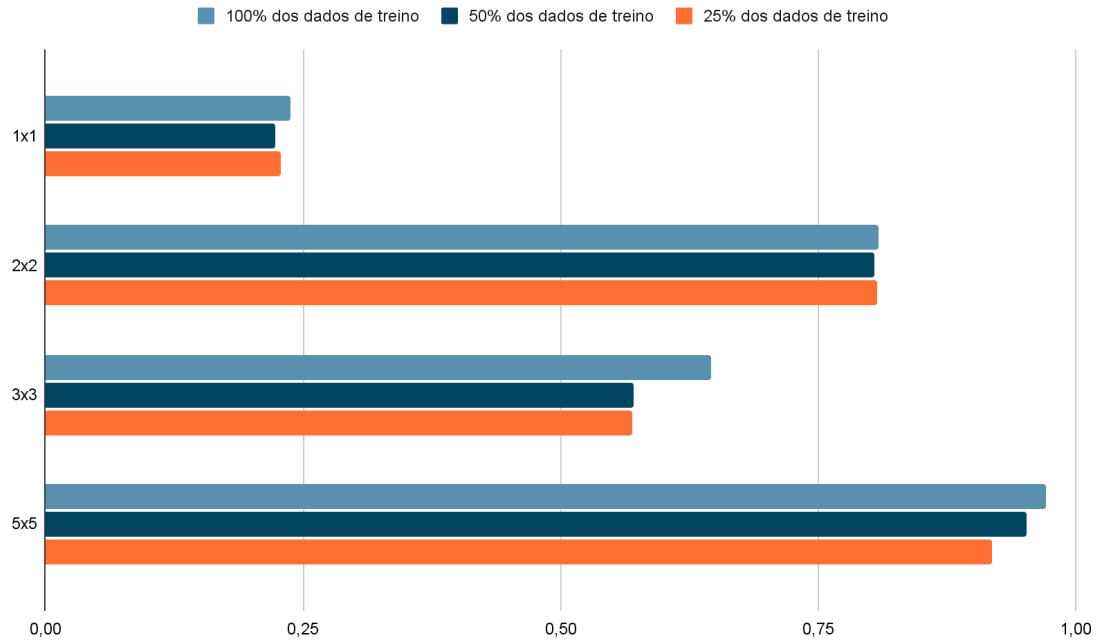

Gráfico 2. Representa a influência que a escolha de subconjuntos aleatórios dos dados de treino tem com a acurácia final.

Conjunto de Treino	**Acuracia (1x1)**	**Acuracia (2x2)**	**Acuracia (3x3)**	**Acuracia (5x5)**
100%	23.80%	80.90%	64.60%	97.10%
50%	22.30%	80.40%	57.10%	95.20%
25%	22.80%	80.70%	57.00%	91.80%

Tabela 2. Representa a influência que a escolha de subconjuntos aleatórios dos dados de treino tem com a acurácia final.

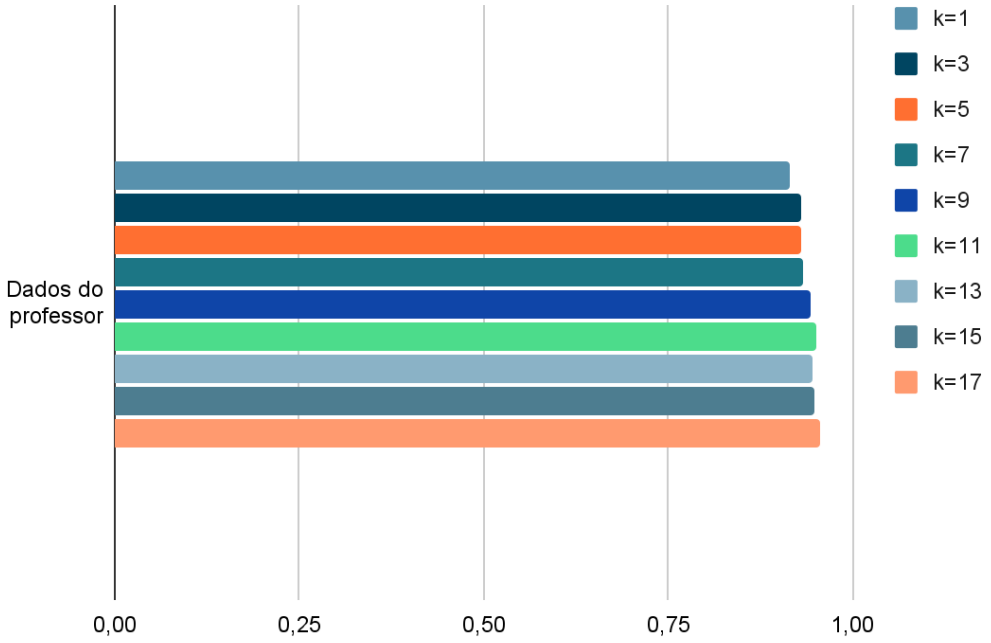

Gráfico 3. Representa a influência que o valor de K tem com a acurácia final no conjunto de dados do professor.

K	Acurácia
1	91.54%
3	93.09%
5	92.99%
7	93.39%
9	94.34%
11	94.99%
13	94.69%
15	94.89%
17	95.50%
19	95.50%

Tabela 3. Representa a influência que o valor de K tem com a acurácia final no conjunto de dados do professor.