Trabalho 1 - K-Nearest Neighbor (K-NN)

Catarine Soares Cruz[1], Gustavo Zanzin Guerreiro Martins[1]

[1]Departamento Acadêmico de Computação – Universidade Tecnológica Federal do Paraná (UTFPR) – 86812-460 – Campo Mourão – PR – Brazil

{catarinecruz, gustavozanzin}@alunos.utfpr.edu.br

Abstract. *This article describes how the k-Nearest Neighbors (k-NN) algorithm was implemented using different values of k and the Euclidean distance. The goal was to classify data in a test set based on their proximity to the training data. The results showed that the choice of k value had a significant impact on the obtained results.*

Resumo. *Este artigo descreve a forma de como foi implementado o algoritmo k-Nearest Neighbors (k-NN) utilizando diferentes valores de k e a distância Euclidiana. O objetivo foi classificar dados em um conjunto de testes com base na sua proximidade aos dados de treinamento. Os resultados mostraram que a escolha do valor de k teve um impacto significativo nos resultados obtidos.*

1. Introdução

O algoritmo dos k-vizinhos mais próximos, conhecido como *KNN* ou *k-NN*, é um classificador de aprendizado supervisionado não-paramétrico que usa proximidade para fazer classificações ou previsões sobre o agrupamento de um determinado dado. Normalmente é utilizado como um algoritmo de classificação, baseando-se na suposição de que pontos ou amostras semelhantes, ou seja, pertencentes à mesma classe, podem ser encontrados próximos uns dos outros.

Como o algoritmo utiliza a proximidade das amostras em relação umas com as outras para realizar as classificações, é extremamente importante escolher a maneira na qual a distância será calculada de forma que melhor se adeque para o caso no qual será analisado. Dessa forma, a mais habitualmente usada é a distância Euclidiana, que calcula a distância em linha reta, que por sua vez será utilizada nesta implementação. Entretanto existem outras formas para calcular tal métrica, como a distância de Manhattan.

Além disso, é agudamente pertinente salientar o que é a variável k e qual o seu impacto no algoritmo. O valor k no algoritmo dos k-vizinhos mais próximos define quantos vizinhos serão verificados para determinar a classificação de uma amostra específica em um teste. Assim, selecionar um valor para k é extremamente importante para a acurácia do algoritmo, pois diferentes valores podem levar a superajuste ou subajuste. Portanto, utilizar-se-á diferentes valores de k, para compreender melhor seus impactos nos resultados.

2. Procedimentos

Para a implementação do algoritmo k-NN, foi utilizado a linguagem de programação *Python* e as bibliotecas *Numpy*, *Pandas*, *SciKit-Learn* e *Collections*. A priori, foram definidas as funções auxiliares para calcular a distância Euclidiana e realizar o cálculo da quantidade desejada de dados aleatórios (25% e 50%). Desse modo, com as funções auxiliares definidas implementou-se o algoritmo de k-NN em si.

Primeiramente, a função *fit* armazena as amostras do treinamento, a variável *X_train* recebe as características e o *Y_train* recebe as classes. Já a função *predict*, recebe as amostras do Teste e posteriormente percorre cada linha do arquivo de Teste. Desse modo, a função *_predict* receberá cada linha do Teste e obterá as distâncias entre uma linha do Teste e cada linha do Treino, após obter as distâncias é necessário ordená-la da menor distância para a maior, isso é feito utilizando o *np.argsort* que devolve os índices ordenados e recebe até o valor de k escolhido (1, 3, 5, 7, 9, 11, 13, 15, 17, ou 19).

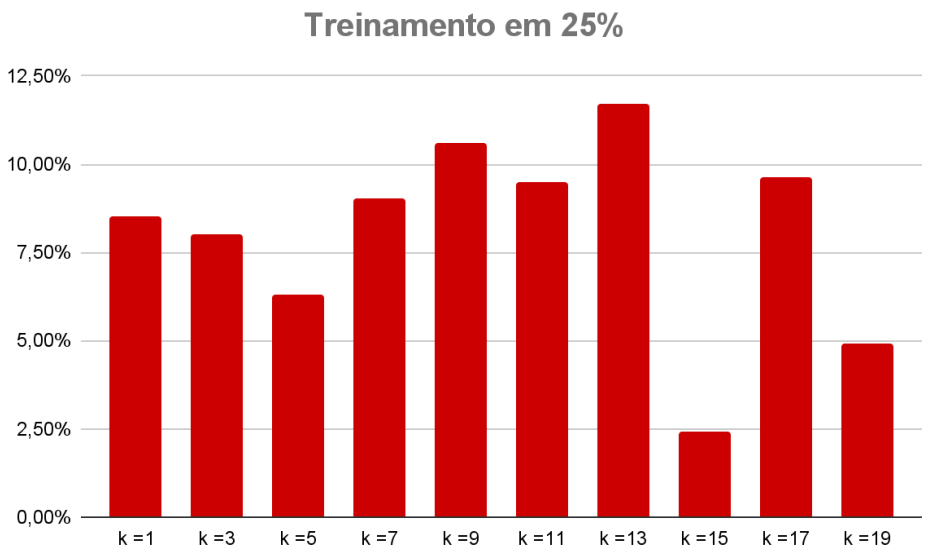
Logo após, é preciso coletar quais as classes que os *k* primeiros pertencem e obter a classe que teve maior recorrência. Ao receber esse valor é somando a quantidade de vezes que ocorre com o *np.sum.* Assim, para finalmente obter a acurácia basta dividir o resultado da quantidade de vezes pelo número de classes e visualizar o valor obtido.

3. Resultados

Após a explicação da implementação, é possível colocar o algoritmo em prática, testando para os diferentes valores de *k* e para as diferentes porcentagens da quantidade dos dados de treinamento.

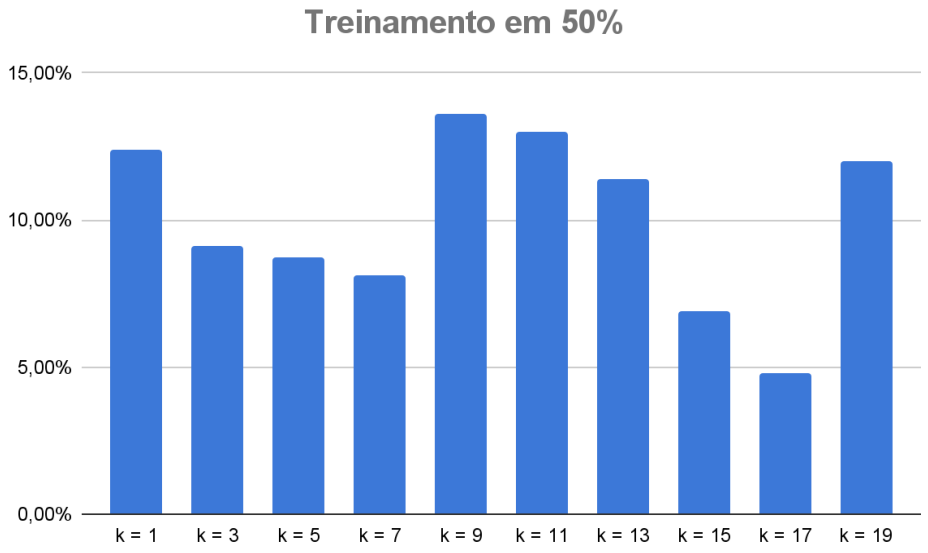
Observa-se no Gráfico 1, utilizando apenas um quarto das características, que os melhores valores para *k* foram 13 e 9, que alcançaram valores mais altos de acurácia. Contudo, é possível analisar que o valor máximo da acurácia obtido foi 11.71%, ou seja, é um resultado extremamente insatisfatório.

Gráfico 1 - utilizando-se apenas 25% dos dados.



Partindo para a análise do Gráfico 2, utilizou-se o dobro dos dados que foram utilizados no Gráfico 1, a acurácia obtém uma melhora, porém pequena. O valor máximo obtido foi de 12.41%, o que ainda deixa insatisfatório o resultado. Os valores de k que obtiveram as acurácias mais próximas do limite foram 9 e 11.

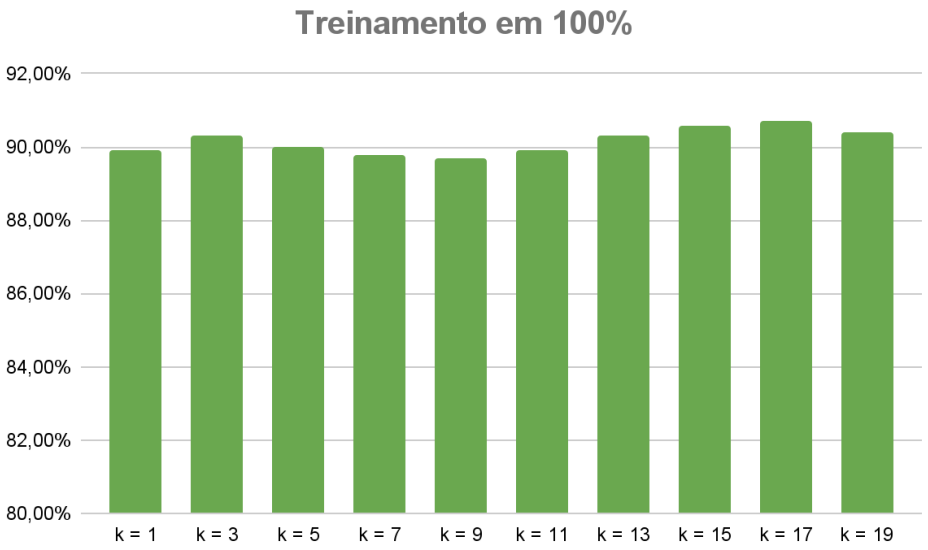
Gráfico 2 - utilizando-se apenas 50% dos dados.



Prosseguindo a análise, agora com todas as características da amostra, observamos resultados bastantes satisfatórios. Nesse caso, os valores de k quase não

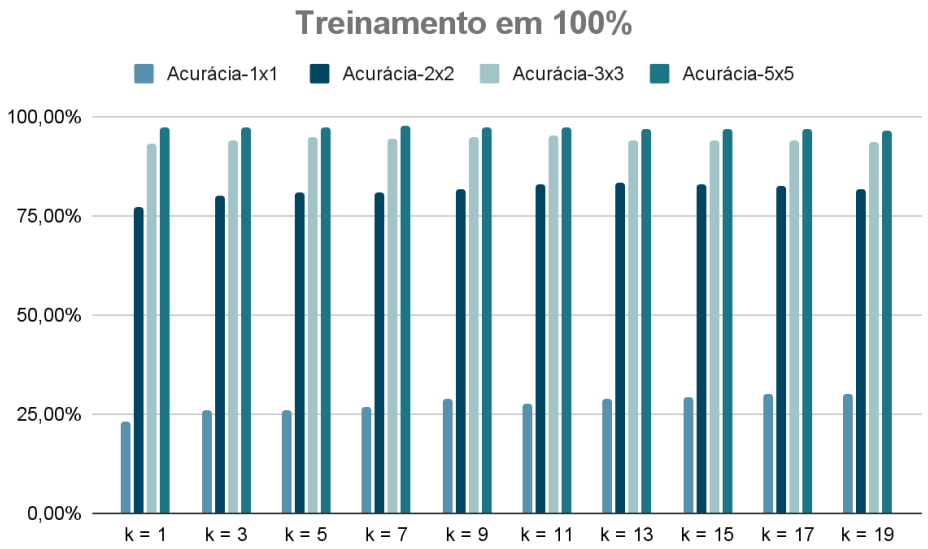
influenciaram no resultado final da acurácia, os valores ficaram bem próximos entre 89.69% e 90.69%. Porém, o maior valor foi atingido pelo k = 17, seguido pelo k = 15.

Gráfico 3 - utilizando-se 100% dos dados.



Utilizando outros arquivos de Teste e Treino, em 100%, mas divididos em diferentes quadrantes (1x1, 2x2, 3x3 e 5x5), obtemos os valores contidos no Gráfico 4. É possível inferir que as condições são muito semelhantes às anteriores. O 1x1 contém poucas classificações, assim, o resultado máximo foi de 30.13%, k = 17, e os diferentes valores de k não trouxeram uma mudança significativa na acurácia. Já para o 2x2, que contém o dobro de características do 1x1, porém, houve uma mudança bastante expressiva na acurácia, que o valor máximo a 83.28%, k = 13, mas os diferentes valores de k não influenciaram muito a acurácia final. Para os 3x3 e o 5x5, apesar da grande diferença de classificações a acurácia foi bem próxima, 3x3 com 95.10% e o 5x5 com 97.70%, para k = 11 e k = 7, respectivamente.

Gráfico 4 - utilizando-se 100% dos dados para cada arquivo.



4. Conclusão

Em suma, pode-se perceber a influência da quantidade de características e os valores do k tem sobre a acurácia final. Dessa maneira, precisa-se encontrar uma quantidade razoável de características, poucas, como observado nos exemplos de 25%, 50%, 1x1 e 2x2 não são, na maioria das vezes, bons. Logo é necessário uma quantidade maior para realizar as análises. Ademais, o valor do k, também, é extremamente importante, pois o que todos tiveram em comum é que os melhores valores foram com k mais alto entre $k = 7$ até $k = 17$. Mostrando que a quantidade de vizinhos próximos influencia diretamente na acurácia. Por conseguinte, é necessário para obter melhores resultados uma boa combinação entre o valor do k e a quantidade de características.

Referências

Loeber, Patrick, “KNN (K Nearest Neighbors) in Python - Machine Learning From Scratch 01 - Python Tutorial” ,YouTube, https://youtu.be/ngLyX54e1LU, Maio.

“What is the k-nearest neighbors algorithm?”, IBM, https://www.ibm.com/topics/knn, Maio.