INF1017 - Aprendizado de Máquina

Trabalho Prático KNN

Autores :
Natália Gubiani Rampon

Professora :
Mariana Recamonde Mendoza

7 de fevereiro de 2021

---

1 Introdução

O presente relatório contém a análise dos resultados obtidos aplicando uma implementação autoral do algoritmo *K-Neareast-Neighbors* com valores de K iguais a 1, 3, 5 e 7. Os dados usados para treino e teste foram disponibilizados pela professora, representando características de núcleos celulares mamários obtidos através do método FNA e seu respectivo diagnóstico (em malignos ou benignos). A partição de dados entre treino e teste já estava previamente executada.

2 Avaliação K-Nearest-Neighbors

Utilizando os dados de treino e teste fornecidos para previsão do diagnóstico e usando o algoritmo *K-Nearest-Neighbors*, obtemos os resultados da Figura 1. Para fins de representação numérica, esses dados também estão presentes na Tabela 1. Ressalta-se que a repartição dos dados entre subconjunto de treino e teste já havia sido previamente realizada pela professora, assim como a normalização destes. A acurácia das previsões foi analisada utilizando o método de *holdout*.

No gráfico da Figura 1, o eixo y representa a acurácia (taxa de previsões corretas feitas pelo algoritmo utilizando os dados de teste) e o eixo x representa o hiper-parâmetro K utilizado.

Notamos que para os dados normalizados, o valor de K tem um pico de desempenho no valor $K = 3$ (com uma acurácia de 95,62%), mas que sua acurácia diminui com o aumento

	K = 1	K = 3	K = 5	K = 7
Normalizados	92,11%	95,61%	94,74%	94,74%
Não-normalizados	87,72%	86,84%	89,48%	88,60%

Tabela 1: Acurácia do algoritmo KNN com diferentes valores de K para os dados normalizados e não normalizados

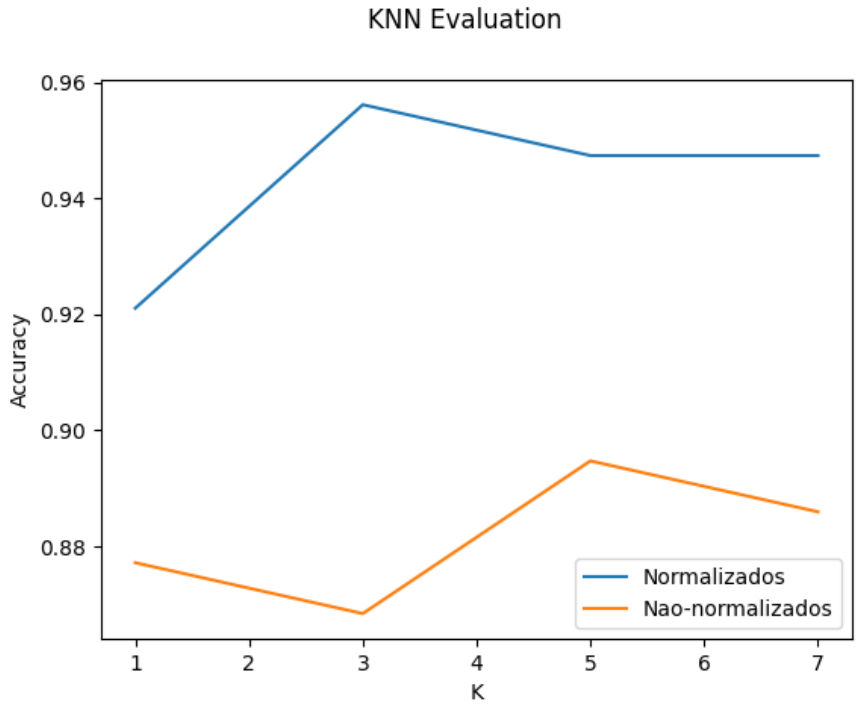

Figura 1: Avaliação do algoritmo KNN com dados normalizados e não-normalizados

para 5 ou 7. Isso faz sentido dependendo da distribuição dos dados, pois podemos imaginar pontos perto da fronteira de decisão onde um aumento demasiado do parâmetro K poderia vir a incluir dados errôneos na comparação, pegando dados muito afastados do ponto de teste sendo analisado.

O mesmo comportamento não se observa para os dados não-normalizados, que têm um pico de desempenho para $K = 5$, mas para os quais o valor de $K = 3$ é ainda pior que o valor de $K = 1$. Notamos que o pico de desempenho observado para os dados não-normalizados (acurácia de 89,47%) é menor do que o observado para os dados normalizados (acurácia de 95,62%). Os resultados obtidos com dados normalizados foram consistentemente melhores do que os obtidos sem a normalização.

Assim, a normalização realmente ajudou a igualar a importância dos atributos, o que aumentou o desempenho do algoritmo, representando um ganho de certa de 6% devido ao tratamento dos dados. Essa melhora obtida com a normalização vem da grande variação dos valores nos diferentes atributos, com um intervalo máximo entre valores de 9960.07 para os dados de treino e de 9972.322 para os dados de teste. Ambos valores de intervalo máximo vêm do atributo da coluna 22. Esse atributo normalmente apresenta valores na ordem da dezena, mas para algumas instâncias percebemos *outliers* próximos de dez

mil. Essas instâncias podem talvez representar dados com algum tipo de erro de medida ou barulho e uma etapa de pré-processamento que retirasse esses valores díspares talvez melhorasse ainda mais o desempenho do algoritmo.