Raport z analizy algorytmu k-NN

na zbiorach Iris i Wine

# Wstęp

W ramach zadania przeprowadzono analizę działania algorytmu k najbliższych sąsiadów ‘k-NN’ na dwóch popularnych zbiorach danych: Iris i Wine z sklearn.datasets. Wybór zbioru Wine uzasadnia jego złożoność: 13 różnych cech, co pozwoli w na lepsze skontrastowanie wyników ze zbiorem Iris, który składa się z 4 różnych cech. Analizowano wpływ liczby sąsiadów ‘k’ oraz wybranej metryki odległości (euclidean, manhattan, minkowski) na dokładność klasyfikacji, czas uczenia modelu oraz całkowity czas wykonania algorytmu.

# Metodologia

Wykorzystano cross-walidację z podziałem na 10 zestawów (KFold cross-validator) dla różnych kombinacji parametrów ‘k={1, 3, 5, 7}’ i metryki odległości: Euclidean, Manhattan, Minkowski.   
Dla każdej z kombinacji obliczono średnią dokładność klasyfikacji, odchylenie standardowe dokładności, średni czas uczenia oraz średni całkowity czas wykonania algorytmu.

# Wyniki

## Zbiór Iris

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| K | Metric | Avg. Accuracy | Std. Deviation | Avg. Training Time (s) | Total Execution Time (s) |
| 1 | Euclidean | 0.9600 | 0.0611 | 0.0005 | 0.0016 |
| 1 | Manhattan | 0.9600 | 0.0611 | 0.0000 | 0.0022 |
| 1 | Minkowski | 0.9600 | 0.0611 | 0.0009 | 0.0017 |
| 3 | Euclidean | 0.9667 | 0.0447 | 0.0006 | 0.0009 |
| 3 | Manhattan | 0.9667 | 0.0447 | 0.0012 | 0.0020 |
| 3 | Minkowski | 0.9667 | 0.0447 | 0.0004 | 0.0031 |
| 5 | Euclidean | 0.9733 | 0.0442 | 0.0016 | 0.0017 |
| 5 | Manhattan | 0.9667 | 0.0447 | 0.0011 | 0.0030 |
| 5 | Minkowski | 0.9733 | 0.0442 | 0.0017 | 0.0021 |
| 7 | Euclidean | 0.9533 | 0.0521 | 0.0002 | 0.0025 |
| 7 | Manhattan | 0.9533 | 0.0521 | 0.0000 | 0.0017 |
| 7 | Minkowski | 0.9533 | 0.0521 | 0.0002 | 0.0021 |

- Najlepsze średnie wyniki dokładności (accuracy) osiągnięto dla ‘k=5’ z metrykami euclidean i minkowski (0.9733) oraz dla ‘k=3’ z dowolną metryką (0.9667).

- Czas uczenia modelu i całkowity czas wykonania były bardzo krótkie dla wszystkich kombinacji, co podkreśla efektywność algorytmu k-NN dla małych zbiorów danych.

## Zbiór Wine

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| K | Metric | Avg. Accuracy | Std. Deviation | Avg. Training Time (s) | Total Execution Time (s) |
| 1 | Euclidean | 0.7310 | 0.0973 | 0.0001 | 0.0033 |
| 1 | Manhattan | 0.8039 | 0.0861 | 0.0010 | 0.0030 |
| 1 | Minkowski | 0.7310 | 0.0973 | 0.0007 | 0.0024 |
| 3 | Euclidean | 0.7036 | 0.1392 | 0.0007 | 0.0032 |
| 3 | Manhattan | 0.7542 | 0.1208 | 0.0007 | 0.0043 |
| 3 | Minkowski | 0.7036 | 0.1392 | 0.0003 | 0.0030 |
| 5 | Euclidean | 0.6641 | 0.1001 | 0.0006 | 0.0024 |
| 5 | Manhattan | 0.7542 | 0.0983 | 0.0000 | 0.0017 |
| 5 | Minkowski | 0.6641 | 0.1001 | 0.0001 | 0.0038 |
| 7 | Euclidean | 0.6588 | 0.1007 | 0.0000 | 0.0027 |
| 7 | Manhattan | 0.7422 | 0.0923 | 0.0006 | 0.0029 |
| 7 | Minkowski | 0.6588 | 0.1007 | 0.0015 | 0.0016 |

- Najwyższa średnia dokładność została osiągnięta dla ‘k=1’ z metryką manhattan (0.8039). Ogólnie, metryka manhattan wydaje się być bardziej skuteczna na tym zbiorze danych.

- Czas uczenia i całkowity czas wykonania pozostały na niskim poziomie, choć były zauważalnie wyższe niż dla zbioru Iris, co jest zgodne z oczekiwaniami ze względu na większą złożoność zbioru Wine.

# Wnioski

**1. Wybór parametru ‘k’:** Najlepsze wyniki uzyskano dla średnich wartości ‘k’ (3-5) na obu zbiorach danych. Zbyt mała wartość ‘k’ może prowadzić do przetrenowania, natomiast zbyt duża zmniejsza dokładność klasyfikacji.

**2. Wybór metryki odległości:** Metryka manhattan okazała się nieco bardziej efektywna na zbiorze Wine, podczas gdy na zbiorze Iris wszystkie trzy metryki osiągnęły porównywalne wyniki. Wybór metryki powinien być dopasowany do specyfiki danych.

**3. Czas uczenia i wykonania:** Algorytm k-NN charakteryzuje się niskim czasem uczenia, co czyni go atrakcyjnym dla zastosowań wymagających szybkiego przetwarzania. Jednakże, należy pamiętać, że czas przewidywania może rosnąć wraz ze wzrostem rozmiaru zbioru danych, ponieważ algorytm musi obliczyć odległość od nowego punktu do wszystkich punktów w zbiorze uczącym.

**4. Ogólna efektywność algorytmu:** k-NN okazuje się być skutecznym i efektywnym algorytmem dla niewielkich zbiorów danych jak Iris i Wine. Ze względu na prostotę implementacji   
i interpretacji wyników, algorytm ten może służyć jako solidna podstawa dla bardziej złożonych zastosowań w dziedzinie uczenia maszynowego.