

Metody sztucznej inteligencji - raport projektu I

T: Implementacja i badanie skuteczności algorytmu k najbliższych sąsiadów (k-NN)

Autorzy: Kornel Żaba, Mikołaj Słoń

1. Opis algorytmu:

a) k-NN klasyczne

Algorytm k najbliższych sąsiadów jest algorytmem uczenia nadzorowanego używanym w problemie klasyfikacji. Dla danego zbioru próbek treningowych T , próbka testowa t jest klasyfikowana w następujący sposób:

- 1) Dla każdej próbki treningowej jest wyliczany dystans euklidesowy do próbki testowej t .
- 2) Z próbek treningowych jest wybierane k próbek z najmniejszym dystansem do próbki testowej t .
- 3) Próbkę t jest przypisana etykieta do której należy większość z k najbliższych próbek ('głosowanie').

b) k-NN eksperymentalne

Algorytm eksperymentalny zawiera w sobie dwie zmiany w stosunku do oryginalnego algorytmu k-NN.

- 1) W fazie treningu, dla każdej klasy C jest znajdowane uśrednione centrum C_t na podstawie wartości próbek treningowych T_i należących do klasy C , a także dystans D_i każdej próbki treningowej do centrum C_t oraz największy dystans D_{max} między punktem C_t oraz jednym z punktów treningowych, należącym do klasy C .

Wzór na uśrednione centrum klasy C_t

$$C_t = \frac{1}{N} \sum T_i$$

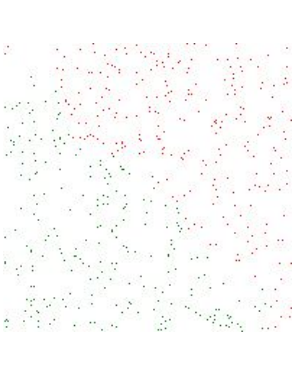
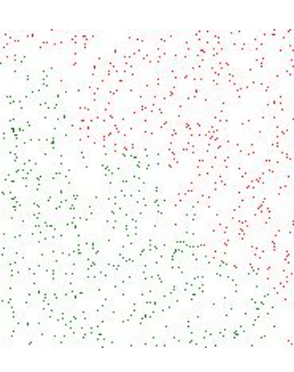
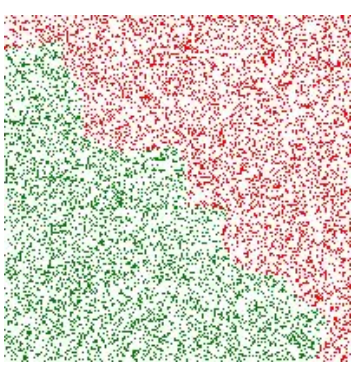
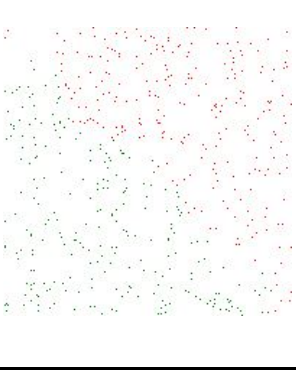
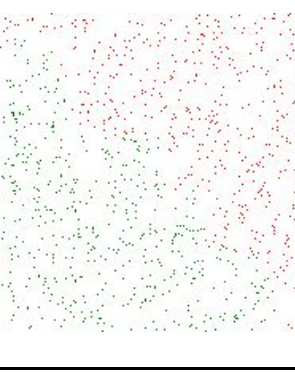
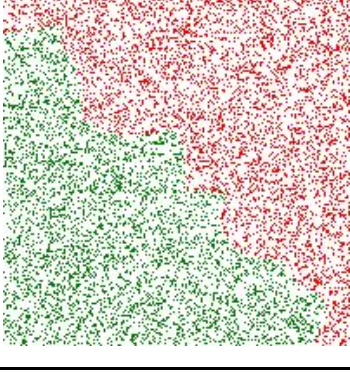
- 2) W fazie klasyfikacji, podczas przypisywania etykiety do próbki testowej t , głosy k najbliższych próbek treningowych są ważone. Waga W_i jest tym większa, im bliżej dana próbka treningowa jest centrum swojej klasy. Waga jest znormalizowana dystansem D_{max} .

Wzór na wagę głosu próbki treningowej T_i .

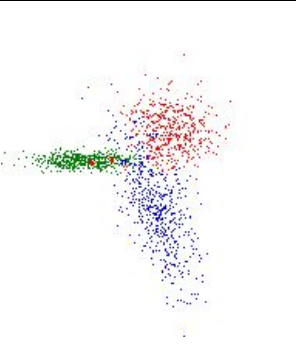
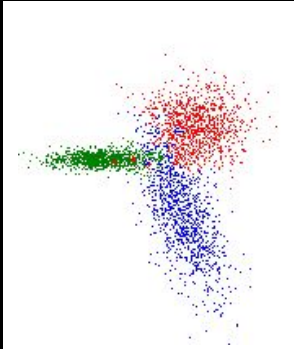
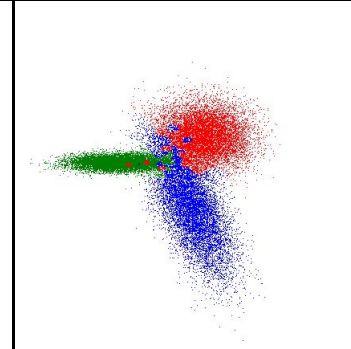
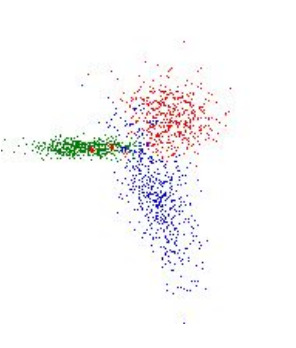
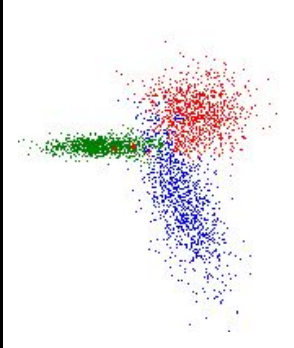
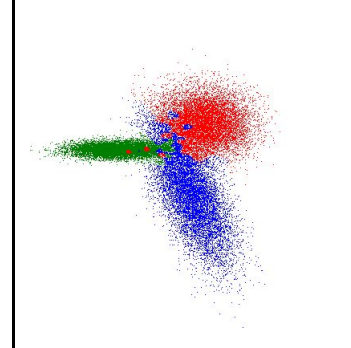
$$W_i = \frac{D_{max}}{D_i}$$

2. Tabele i Wykresy

a) Data.Simple

k = 20	100:500	100:1000	100:10 000
k-NN klasyczne			
k-NN eksperymentalne			

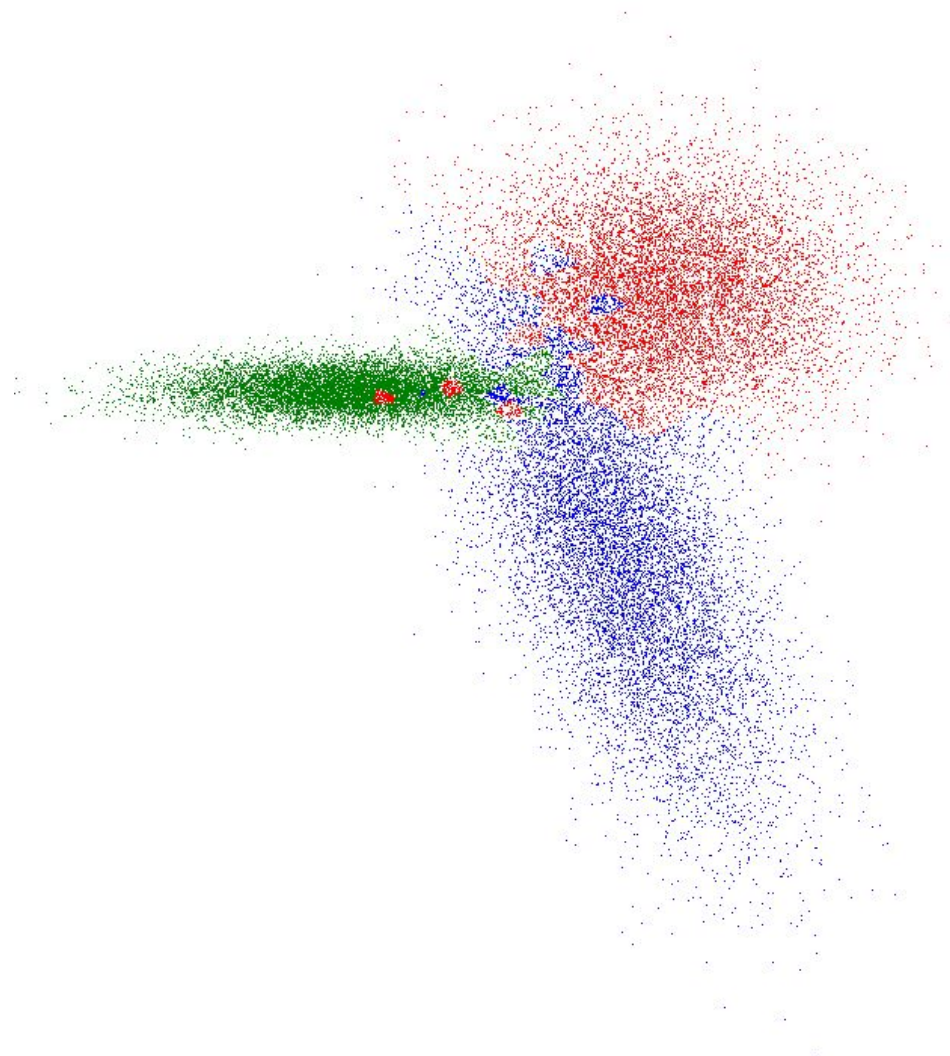
b) Data.three_gauss

k = 20	100:500	100:1000	100:10 000
k-NN klasyczne			
k-NN eksperymentalne			

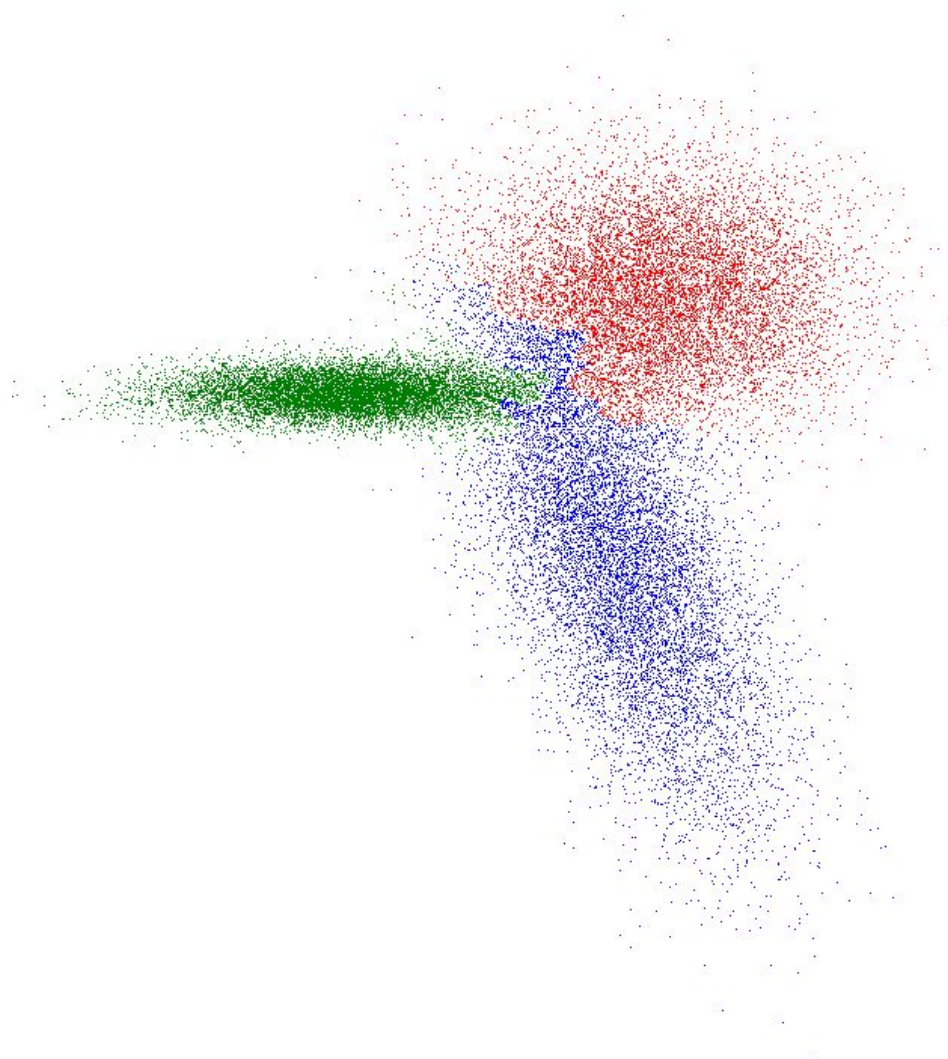
3. Wnioski

a) Zmiana wielkości sąsiedztwa

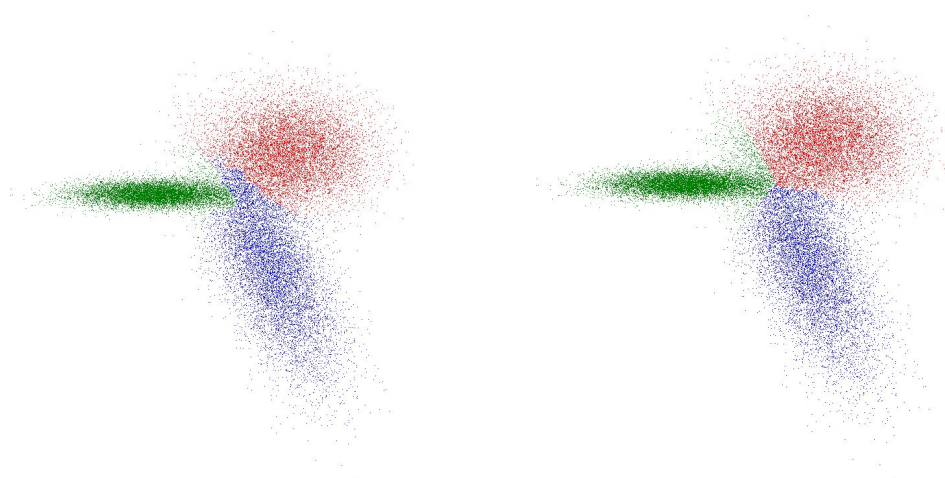
Przy próbkce treningowej `data.three_gauss.train.100` i próbkce testowej `data.three_gauss.train.10000` możemy zaobserwować jak zmiana wielkości sąsiedztwa wpływa na klasyfikację punktów. Dla ekstremalnego przypadku k równego jeden otrzymujemy następujący wynik:



Możemy zaobserwować pojedyncze enklawy punktów formułujące się w środku zbiorów. Dzieje się tak ponieważ nawet mała ilość punktów odbiegających współrzędnymi od większości punktów danej klasyfikacji może wpłynąć na klasyfikację. Zwiększając k do pięciu osiągamy następujący wykres:



Enklawy przestały istnieć bądź zmniejszyły się. Takiej zależności możemy się spodziewać zwiększając k do kolejno 20 i 100:



Zwiększając kolejno k uzyskujemy silniejszy podział i redukujemy wpływ obserwacji silnie odstających od normy ale jednocześnie uogólniamy kształt zbiorów redukując precyzję.

b) Wyniki algorytmu klasycznego i eksperymentalnego



4. Manual

Program przyjmuje cztery parametry:

- Wersję algorytmu który ma zostać zastosowany do klasyfikacji. "1" oznacza klasyczną implementację k-NN a "2" implementację eksperymentalną. W przypadku gdy pierwszy argument nie przyjmie wartości "1" albo "2" klasyczna implementacja k-NN zostanie zastosowana.
- Nazwę pliku na podstawie którego algorytm ma trenować. Plik ten musi znajdować się w folderze w którym program jest używany. W przypadku braku możliwości otwarcia danego pliku program o tym poinformuje i zakończy działanie.
- Nazwę pliku testowego
- Wielkość sąsiedztwa, w przypadku podania nieprawidłowej liczby program przyjmie domyślną wartość równą 20

W przypadku uruchomienia programu bez podania argumentów lub przy nieprawidłowej ich ilości należy postępować zgodnie z instrukcjami wyświetlanymi na konsoli.

Skrypt testowy przyjmuje zmienną liczbę parametrów.

- Względną ścieżkę do zbioru danych.
- Liczbę powtórzeń danej konfiguracji (w każdym powtórzeniu, zbiór danych jest losowo 'przetaskowany').
- Literę 't' i następujący po nim ciąg liczb oznaczający możliwe konfiguracje wielkości zbioru treningowego.
- Literę 'k' i następujący po nim ciąg liczb oznaczający możliwe konfiguracje wartości parametru sąsiedztwa.

Przykładowe wywołanie skryptu:

```
start ../KnnLib/TestScript/bin/Release/TestScript.exe
```

```
../data/data.three_gauss.train.10000.csv 12 t 5000 10000 k 8 12 16 32
```

Każdy skrypt jako wyjście tworzy 2 pliki .csv, Scenario.csv i TotalScenario.csv.

Scenario.csv zawiera wyniki każdej iteracji dla każdej kombinacji.

TotalScenario.csv zawiera podsumowanie statystyczne wyników w Scenario.csv

Każdy z plików csv jest oznaczony datą powstania.

Pliki Scenario.csv zawierają kolumny:

- 1) Path - ścieżka do danego zbioru danych
- 2) Algorithm - nazwa użytego algorytmu k-NN
- 3) K - wartość parametru sąsiedztwa
- 4) TrainingSize - wielkość zbioru treningowego
- 5) Correct - liczba prawidłowo zaklasyfikowanych próbek
- 6) TestSize - wielkość zbioru treningowego
- 7) Accuracy - liczba procentowa prawidłowo zaklasyfikowanych próbek
- 8) TrainingTime - czas trwania fazy treningu w sekundach
- 9) TestTime - czas trwania fazy klasyfikacji w sekundach

Pliki TotalScenario.csv zawierają kolumny:

- 1) Algorithm - nazwa użytego algorytmu k-NN
- 2) K - wartość parametru sąsiedztwa
- 3) TrainingSize - wielkość zbioru treningowego
- 4) AvgCorrect - średnia liczba prawidłowo zaklasyfikowanych próbek we wszystkich iteracjach
- 5) StdCorrect - standardowe odchylenie liczby prawidłowo zaklasyfikowanych próbek we wszystkich iteracjach
- 6) AvgAccuracy - średnia liczba procentowa prawidłowo zaklasyfikowanych próbek we wszystkich iteracjach
- 7) StdAccuracy - standardowe odchylenie liczby procentowej prawidłowo zaklasyfikowanych próbek we wszystkich iteracjach

W folderze scripts znajdują się przykładowe skrypty oraz foldery z gotowymi wynikami.