# K najbliższych sąsiadów - zadanie drugie

29 października 2017

Grzegorz Borkowski

## 1. WSTĘP

Zadanie zostało zaimplementowane w Pythonie 3.6.1
Do raportu został dołączony Jupyter Notebook z kodem źródłowym programu.

## 2. ZADANIE 2

### *2.1 Treść zadania*

Celem zadania jest obserwacja wpływu kompresji CNN na pracę klasyfikatora k-NN.

Przygotowujemy 2 zbiory danych, a następnie obserwujemy wygląd granicy decyzyjnej i skuteczność klasyfikacji. Tym razem jednak korzystamy z następujących klasyfikatorów:
* zwykły k-NN z k=1 i metryką Euklidesa
* CNN z k=1 i metryką Euklidesa
* zwykły k-NN z k=3 i metryką Euklidesa
* CNN z k=3 i metryką Euklidesa

Dodatkowo dla obu CNN raportujemy o ile procent został zmniejszony po jego zastosowaniu zbiór z wiedzą klasyfikatora. Warto też te usunięte elementy odpowiednio zobrazować na diagramie granicy decyzyjnej. Wyniki oczywiście należy opatrzyć komentarzem.

Dla każdej pary zbiór-klasyfikator zwizualizuj wygląd granicy decyzyjnej oraz oszacuj błąd takiego klasyfikatora

### *2.2 Zbiory danych*

Dane treningowe i testowe zostały wygenerowane metodą make classification z pakietu scikit.dataset. Metoda ta generuje losowy n-klasowy problem klasyfikacyjny z zadanymi parametrami. [1]

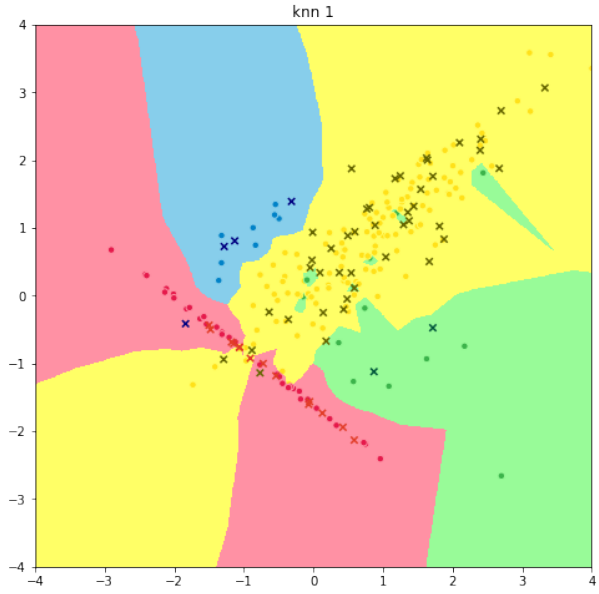
### *2.3 Pierwszy zbiór danych*

Zbiór wejściowy składa się ze 250 elementów, podzielonych na 4 klasy. Zero klasa ma liczność 51 elementów, pierwsza, 15 elementów, druga 12 elementów, trzecia klasa ma 172 elementów. Podzieliłem zbiór na dwa zbiory, treningowy i testowy (treningowy: 187, testowy: 63 elementów) w losowy sposób. Zbiór składa się z dwóch cech.

*kNN, k=1, Euklides*

| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.85 | 0.79 | 0.81 |
| Klasa 1 | 0.50 | 1.00 | 0.67 |
| Klasa 2 | 1.00 | 0.75 | 0.86 |
| Klasa 3 | 0.93 | 0.93 | 0.93 |
| avg-total | 0.90 | 0.89 | 0.89 |

Skuteczność klasyfikatora: 0.89



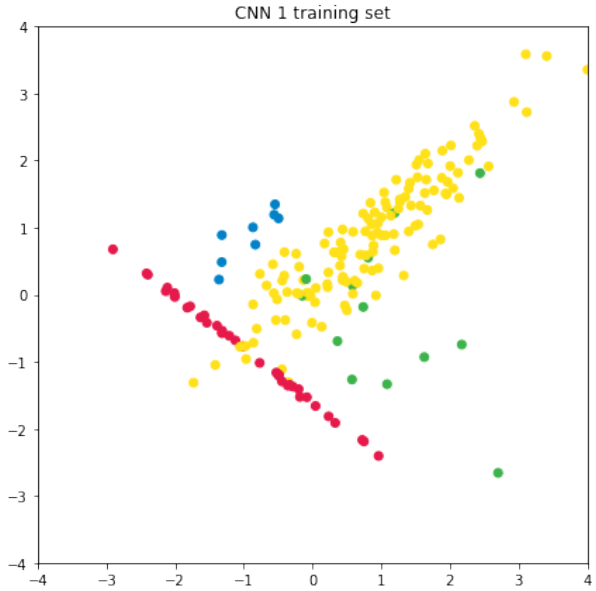

Rysunek 1. Granica decyzyjna dla kNN, k=1, z metryką Euklidesa

*CNN, k=1, Euklides*

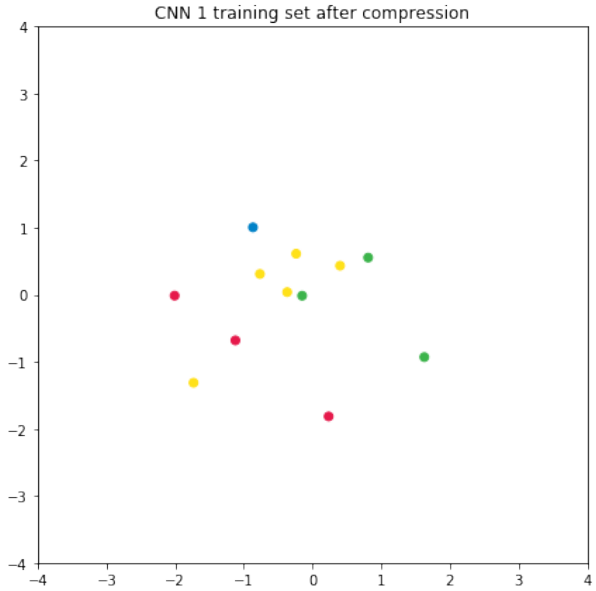
Wejściem do algorytmu CNN był zbiór treningowy o rozmiarach: 37 elementy klasy zero, 13 elementów pierwszej klasy, 8 elementów drugiej klasy i 129 elementów trzeciej klasy. Na wyjściu otrzymaliśmy: 3 elementy zero klasy, 3 elementy pierwszej klasy, 1 element drugiej klasy, 5 elementów trzeciej klasy

| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.78 | 1.00 | 0.88 |
| Klasa 1 | 0.06 | 1.00 | 0.12 |
| Klasa 2 | 1.00 | 0.75 | 0.86 |
| Klasa 3 | 1.00 | 0.26 | 0.41 |
| avg-total | 0.92 | 0.48 | 0.53 |

Stopień kompresji (stosunek liczby punktów nowego zbioru treningowego do początkowej liczby punktów zbioru treningowego): 0.06 (94 procent punktow zostalo usuniete).
Skuteczność klasyfikatora: 0.476




Rysunek 2. Zbiór treningowy dla CNN, k=1 przed kompresją




Rysunek 3. Zbiór treningowy dla CNN, k=1 po kompresji

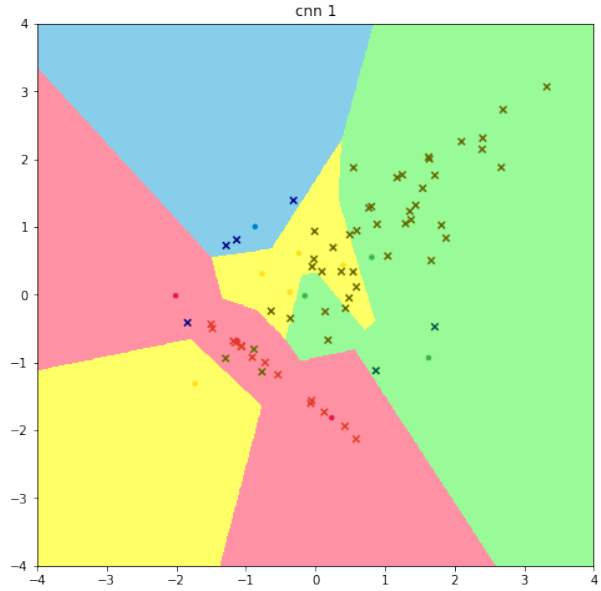
*kNN, k=3 Euklides*

| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.85 | 0.79 | 0.81 |
| Klasa 1 | 1.00 | 1.00 | 1.00 |
| Klasa 2 | 1.00 | 0.75 | 0.86 |
| Klasa 3 | 0.93 | 0.98 | 0.95 |
| avg-total | 0.92 | 0.92 | 0.92 |

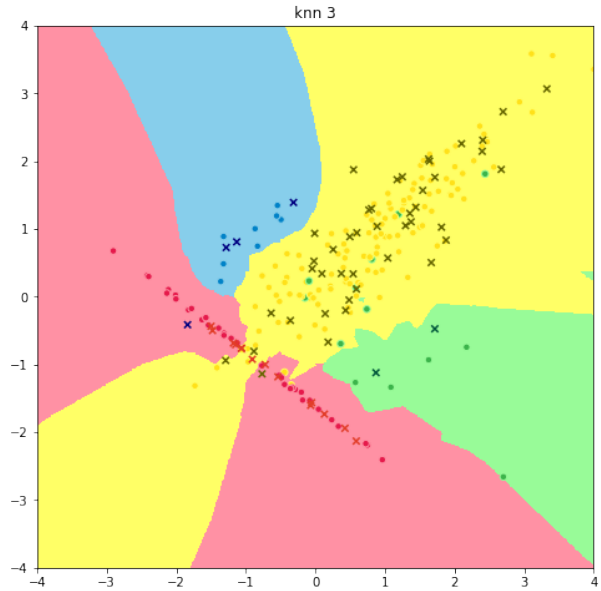
Skuteczność klasyfikatora: 0.93

*CNN, k=3, Euklides*

Wejściem do algorytmu CNN był zbiór treningowy o rozmiarach: 37 elementy klasy zero, 13 elementów pierwszej klasy, 8 elementw drugiej klasy i 129 elementów trzeciej klasy. Na wyjściu otrzymaliśmy: 5 elementów klasy 0, 3




Rysunek 4. Granica decyzyjna dla CNN z metryką Euklidesa dla k=1



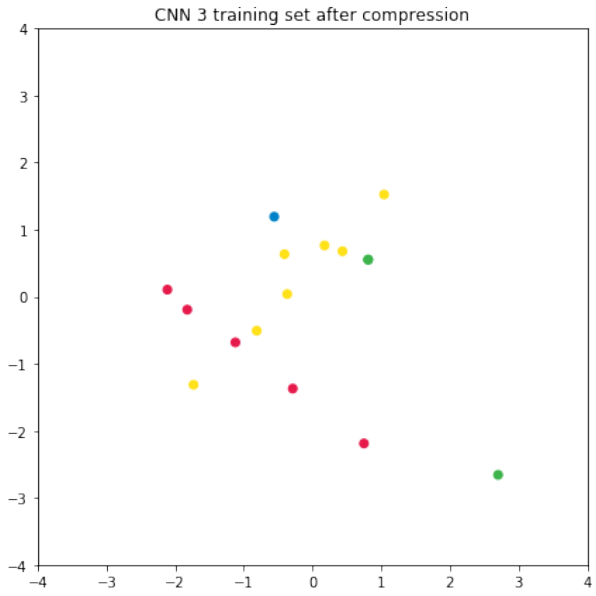

Rysunek 5. Granica decyzyjna dla CNN z metryką Euklidesa dla k=1

elementy klasy pierwszej, 1 element klasy drugiej, 7 elementów klasy trzeciej.

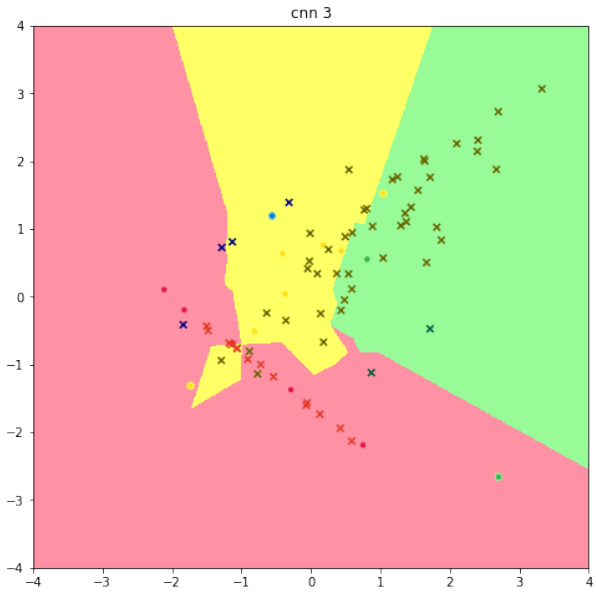
| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.69 | 0.79 | 0.73 |
| Klasa 1 | 0.04 | 0.50 | 0.07 |
| Klasa 2 | 0.00 | 0.00 | 0.00 |
| Klasa 3 | 0.75 | 0.35 | 0.48 |
| avg-total | 0.67 | 0.43 | 0.49 |

Stopień kompresji : 0.085 (91.5 procent punktow zostalo usuniete).
Skuteczność klasyfikatora: 0.4285

*kNN z metryką Euklidesa k=3*



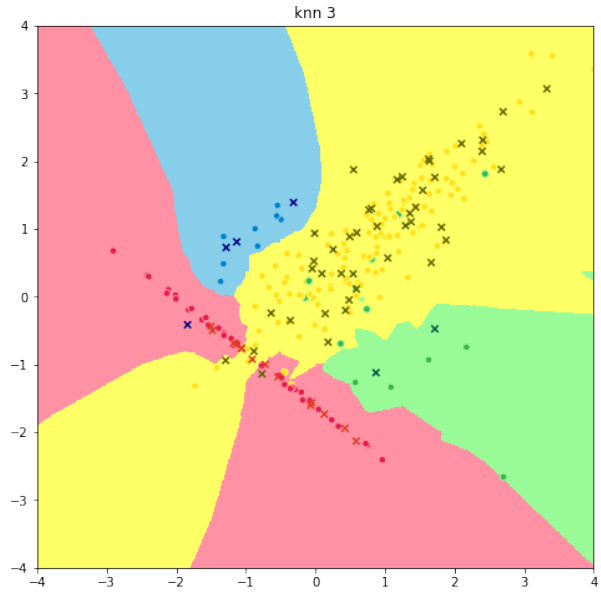

Rysunek 6. Zbiór treningowy dla CNN, k=3 po kompresji




Rysunek 7. Granica decyzyjna dla CNN, k=3

| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.85 | 0.79 | 0.81 |
| Klasa 1 | 1.00 | 1.00 | 1.00 |
| Klasa 2 | 1.00 | 0.75 | 0.86 |
| Klasa 3 | 0.93 | 0.98 | 0.95 |
| avg-total | 0.92 | 0.92 | 0.92 |

Skuteczność klasyfikatora: 0.9206




Rysunek 8. Granica decyzyjna dla kNN z metryką Euklidesa, k=3

### *2.4 Drugi zbiór danych*

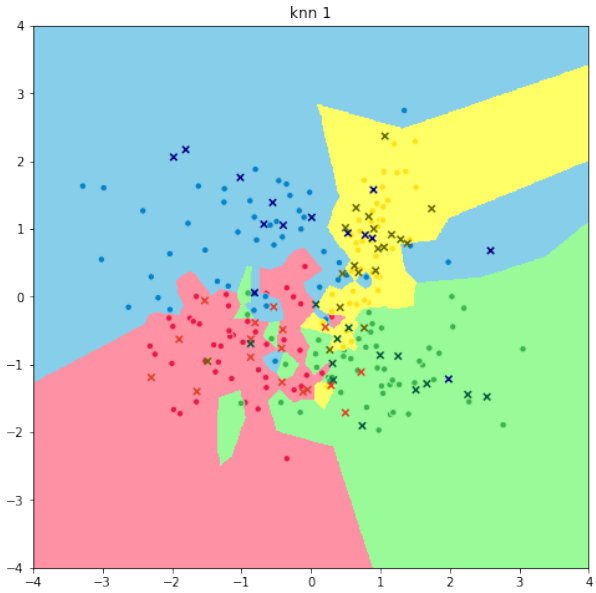
*Opis zbioru danych*

Zbiór danych składa się z czterech klas o liczności odpowiednio: 62, 63, 62, 63 elementy. Podzieliłem ten zbiór na zbiór treningowy i testowy (187 elementów i 63 elementy).

*kNN, k=1, Euklides*

| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.80 | 0.71 | 0.75 |
| Klasa 1 | 0.69 | 0.69 | 0.69 |
| Klasa 2 | 0.75 | 0.64 | 0.69 |
| Klasa 3 | 0.65 | 0.79 | 0.71 |
| avg-total | 0.72 | 0.71 | 0.71 |

Skuteczność klasyfikatora: 0.71




Rysunek 9. Granica decyzyjna dla kNN, k=1, z metryką Euklidesa

*CNN, k=1, Euklides*

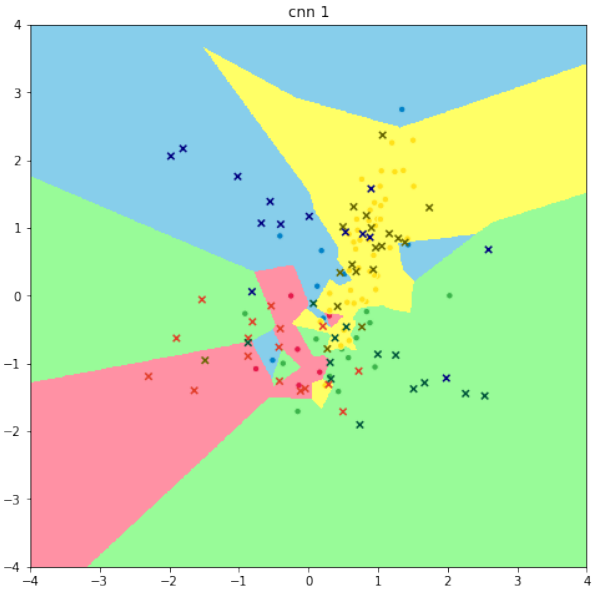
Wejściem do algorytmu CNN był zbiór treningowy o rozmiarach: 45 elementy klasy zero, 50 elementów pierwszej klasy, 48 elementów drugiej klasy i 44 elementów trzeciej klasy. Na wyjściu otrzymaliśmy: 6 elementów zero klasy, 13 elementów pierwszej klasy, 8 elementów drugiej klasy, 44 elementów trzeciej klasy


Rysunek 10. Zbiór treningowy dla CNN, k=1 przed kompresją

| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.78 | 0.41 | 0.54 |
| Klasa 1 | 0.45 | 0.69 | 0.55 |
| Klasa 2 | 0.70 | 0.50 | 0.58 |
| Klasa 3 | 0.62 | 0.79 | 0.70 |
| avg-total | 0.65 | 0.60 | 0.60 |


Rysunek 11. Zbiór treningowy dla CNN, k=1 po kompresji

Stopień kompresji (stosunek liczby punktów nowego zbioru treningowego do początkowej liczby punktów zbioru treningowego): 0.379 (62 procent punktow zostalo usuniete). Skuteczność klasyfikatora: 0.603

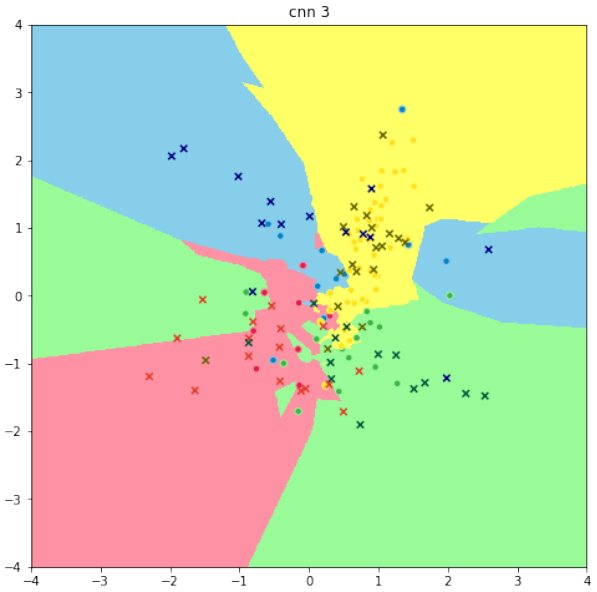



Rysunek 12. Granica decyzyjna dla CNN z metryką Euklidesa dla k=1

*CNN, k=3, Euklides*

Wejściem do algorytmu CNN był zbiór treningowy o rozmiarach: 45 elementów klasy zero, 50 elementów pierwszej klasy, 48 elementow drugiej klasy i 44 elementów trzeciej klasy. Na wyjściu otrzymaliśmy: 8 elementów klasy 0, 15 elementów klasy pierwszej, 11 elementów klasy drugiej, 44 elementów klasy trzeciej.


Rysunek 13. Zbiór treningowy dla CNN, k=3 po kompresji




Rysunek 14. Granica decyzyjna dla CNN, k=3

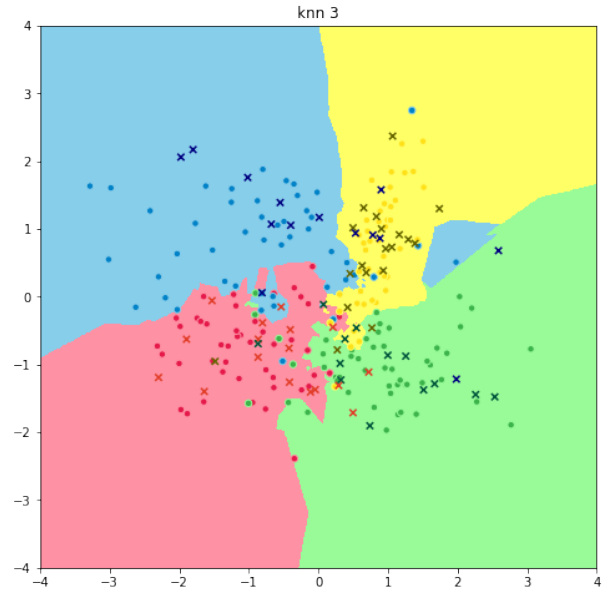
| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.80 | 0.71 | 0.75 |
| Klasa 1 | 0.50 | 0.69 | 0.50 |
| Klasa 2 | 0.89 | 0.57 | 0.70 |
| Klasa 3 | 0.71 | 0.79 | 0.75 |
| avg-total | 0.73 | 0.7 | 0.7 |

Stopień kompresji : 0.41 (59 procent punktow zostalo usuniete).
Skuteczność klasyfikatora: 0.698

*kNN, k=3 Euklides*

| klasa | precision | recall | f1score |
|---|---|---|---|
| Klasa 0 | 0.81 | 0.76 | 0.79 |
| Klasa 1 | 0.56 | 0.69 | 0.62 |
| Klasa 2 | 0.80 | 0.57 | 0.67 |
| Klasa 3 | 0.71 | 0.79 | 0.75 |
| avg-total | 0.73 | 0.71 | 0.72 |

Skuteczność klasyfikatora: 0.71




Rysunek 15. Granica decyzyjna dla kNN z metryką Euklidesa dla k=3

*Wnioski*

(1) CNN w swoim działaniu może usunąć kluczowe punkty ze zbioru treningowego, co może mieć kolosalny wpływ na skuteczność klasyfikatora (Dla pierwszego zbioru testowego CNN granica decyzyjna dla zbioru żółtego jest całkowicie błędna - elementy w znaczącej jego części zostałyby zaklasyfikowane jako elementy klasy zielonej). Np. CNN dla pierwszego zestawu danych osiąga skuteczność rzędu 45 procent, gdy naiwny klasyfikator, który dla każdego elementu ze zbioru testowego przyporządkowywałby mu klasę która była najliczniejsza w zbiorze testowym ma skuteczność rzędu 70 procent).
(2) Wybór liczby sąsiadów ma ogromne znaczenie dla algorytmu CNN. Dla pierwszego zbioru danych, CNN z k=1 poprawnie przyporządkował punkty dla których wartość pierwszej cechy była w zakresie od -4 do 0, a drugiej cechy od 2 do 4 jako punkty z klasy niebieskiej, tymczasem CNN z k=3 ustalił, ze punkty leżące w tym przedziale to punkty czerwone lub żółte.
(3) CNN może osiągać wysokie stopnie kompresji (ok 90 procent dla pierwszego zbioru danych i ok 60 procent dla drugiego zestawu danych)!
(4) Dla pewnych danych CNN może osiągać wysoki stopień kompresji bez znaczącej straty skuteczności kla-

syfikatora (CNN z k=3, usuwa 60 procent punktów i osiąga skuteczność 0.698, w porównaniu do kNN z k=3 który osiąga skuteczność 0.71)!

(5) Eksperyment podkreśla rolę dokonywania cross-validation, aby dobierać odpowiednie metaparametry dla klasyfikatora

## LITERATURA

[1] http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html