Projekt zaliczeniowy z SSI

Rozpoznawanie cyfr z pisma odręcznego za pomocą algorytmu k-NN

Szymon Hankus
Antoni Jaszcz
Bartłomiej Pacia

Politechnika Śląska
Wydział Matematyki Stosowanej

2022/2023

Temat projektu

Projekt dotyczy rozpoznawanie cyfr z pisma odręcznego za pomocą algorytmu k-NN oraz jego modyfikacji.

Opis programu oraz instrukcja obsługi

Projekt składa się z dwóch głównych plików:

- pliku `knn.py`, zawierającego implementację klasyfikatora.
- pliku `knn_project.ipynb`, zawierającego demonstrację klasy i jej możliwości oraz wygenerowane wyniki.

Do uruchomienia projektu niezbędne jest środowisko `Python3` oraz `ipykernel` (lub zainstalowany w środowisku Pythonowym pakiet `ipykernel`). Klasyfikator `knn` został przez nas napisany zgodnie z założeniami modułu Stickit-Learn.

Założenia modułu

- **Jednolitość** - jednolitego interfejsu dla obiektów. Nasz klasyfikator posiada m. in. następujące metody:
  - `fit()` - dopasowującą dane uczące do modelu.
  - `predict()` - zwracającą przewidzianą klasę dla nowej próbki.
  - `score()` - zwracającą skuteczność, macierz pomyłek oraz inne wybrane metryki do oceny modelu na podstawie predykcji zestawu danych walidacyjnych.
  - `cross_val_score()` - dokonujący sprawdzianu krzyżowego dla zestawu danych walidacyjnych i zadanego podziału.
- **Nierozprzestrzenialność klas** - zbiory danych są reprezentowane przez macierze `NumPy`, a parametry to standardowe typy języka `Python` służące do opisu zmiennych.
- **Kompozycja** - klasyfikator podzielony jest na elementy składowe, które są wielokrotnie wykorzystywane.
- **Rozsądne wartości domyślne** - w klasyfikatorze zostały dobrane odpowiednie wartości domyślne (m.in. domyślna ilość sąsiadów $k = 5$).

Algorytm k-nn

k-NN jest prostym, dobrze działającym algorytmem używanym w statystyce do klasyfikacji i regresji. Zasada działania algorytmu jest następująca: biorąc pod uwagę zbiór danych N próbek, wstępnie oznaczonych jako jedna z C klas, z P parametrami liczbowymi, algorytm daje predykcję dla nowej próbki, biorąc pod uwagę jej k "najbliższych sąsiadów (punkty w $dim(P)$ wymiarowej przestrzeni metrycznej, najmniej oddalone) i zwracając odpowiednie dane wyjściowe:

- **Klasyfikacja:** wektor przynależności do klasy, z którego oceniana klasa jest wybierana podczas głosowania pluralistycznego wśród sąsiadów.
- **Regresja:** wektor średnich parametrów obliczonych na podstawie sąsiadów.

Najczęściej jako metrykę odległości w P-wymiarowej przestrzeni wybiera się normę euklidesową. Z tego powodu początkowa normalizacja parametrów jest kluczowa dla prawidłowego działania algorytmu.

Problemy w k-nn

Algorytm w dużym stopniu zależy od liczby próbek. Im więcej próbek k-NN ma do wykorzystania, tym większa szansa na wybranie podobnych obiektów, co skutkuje większą pewnością przewidywanej klasy, a tym samym ogólnie lepszą dokładnością. Niestety, stwarza to następujące problemy:

- Wszelkie anomalie lub zatruwanie danych są bardzo uciążliwe.
- Wysoka złożoność obliczeniowa i duża ilość danych wymaganych do prawidłowego funkcjonowania skutkują stosunkowo wysoką złożonością czasową, co utrudnia implementację w scenariuszach czasu rzeczywistego.

Politechnika Śląska

Tworzenie uśrednionych reprezentantów

Biorąc pod uwagę liczbę próbek n_{c_i} pewnej klasy c_i, możemy przekształcić je w próbki $n_{samples}$ (1), każda utworzona z n_{layer} (2) kolejnych próbek klasy (z wyjątkiem ostatniej, która jest utworzona z pozostałych n_{left} (3) próbek), poprzez ich uśrednienie. Jest to dalej przedstawione na przykładzie w rozdziale.

$$n_{samples} = \lceil \log_2 n_{c_i} \rceil \tag{1}$$

$$n_{layer} = \lceil \frac{n_{c_i}}{n_{samples}} \rceil \tag{2}$$

$$n_{left} = n_{c_i} - ((n_{samples} - 1) * n_{layer}) \tag{3}$$

$$\hat{x}_{i_j} = \begin{cases} \frac{\sum_{k=1}^{n_{layer}} x_k}{n_{layer}} & , for \quad j < n_{samples} \\ \frac{\sum_{k=1}^{n_{left}} x_k}{n_{left}} & , for \quad j = n_{samples} \end{cases} \tag{4}$$

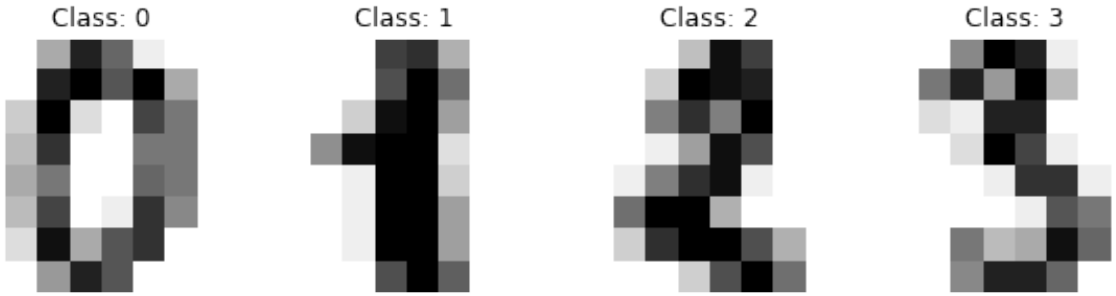
Redukcja wymiarowości poprzez rzutowanie związanych parametrów

Aby zmniejszyć liczbę wymiarów, można zastosować rzutowanie zestawu powiązanych parametrów na nową cechę. Produktem takiej operacji jest nowy zestaw parametrów, reprezentujący nowo utworzone cechy. Można to osiągnąć na przykład poprzez uśrednienie podzbioru powiązanych parametrów P' do nowego pojedynczego parametru $\hat{p}'$ (przedstawionego jako wartość liczbowa).

Politechnika Śląska

Dane oraz ustawienia treningowe

W moich badaniach wykorzystałem UCL Handwritten Digits Data Set, popularny zbiór danych do optycznego rozpoznawania pisma odręcznego. Składa się on z 1797 próbek obrazów cyfr 8x8 pikseli w skali szarości (z intensywnością pikseli w zakresie [0,16]). Każda klasa zawiera około 180 próbek. Przykłady klas (od 0 do 4) pokazano na rysunku.
Ponieważ wszystkie wartości uwzględniane przez algorytm mieszczą się w zakresie [0,16], surowy zbiór danych jest już znormalizowany.

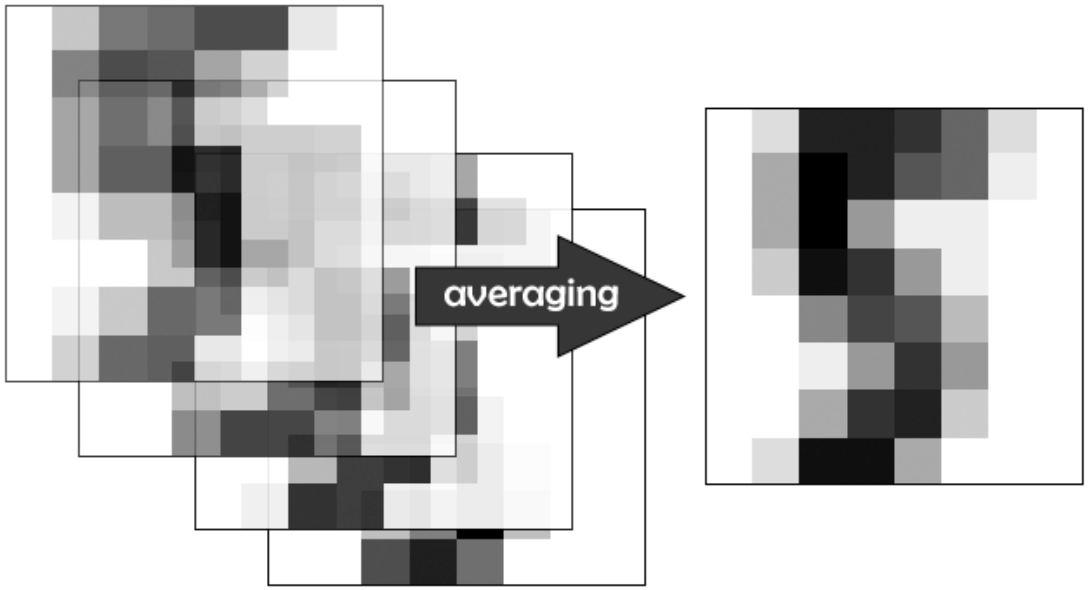


Rysunek: Przykładowe próbki danych

Politechnika Śląska

Reprezentanci klas

W eksperymentach przetestowano wpływ zastosowania metody średniej reprezentatywnej, przekształcając próbki treningowe w odpowiednią liczbę reprezentantów dla każdej klasy. Innymi słowy, tacy reprezentanci zostali utworzeni poprzez nałożenie na siebie pewnej liczby próbek danej klasy, a następnie podzielenie iloczynu przez ich ilość. Najlepiej ilustruje to poniższy rysunek.



Rysunek: Tworzenie przedstawicieli z próbek klasy 5

Politechnika Śląska

Rzutowanie kolumn

Nnowe cechy zostały utworzone przez uśrednienie powiązanych parametrów. W tym przypadku za powiązane parametry uznałem piksele w tej samej kolumnie. Uśredniając wartości pikseli w każdej kolumnie, liczba parametrów została zmniejszona z $8x8 = 64$ do zaledwie $8x1 = 8$. Musiało to jednak zostać wykonane dla próbek w zestawie treningowym, a także dla każdej próbki, która była przewidywana.

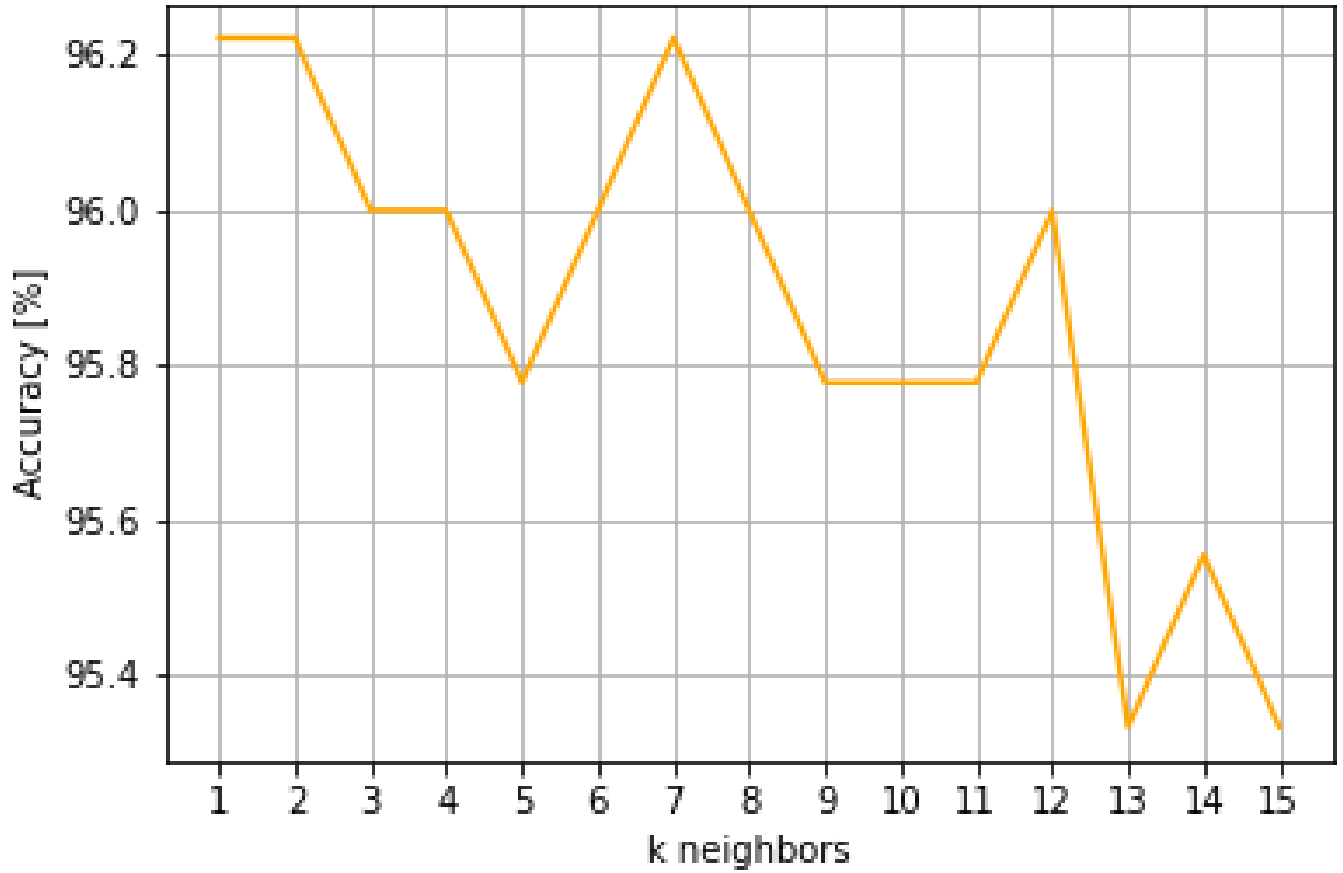
Politechnika Śląska

Dobór parametru k

Kolejną kwestią związaną z algorytmem jest wybór odpowiedniego parametru k (liczba branych pod uwagę sąsiadów). Aby wybrać go efektywnie, przetestowałem wydajność każdego proponowanego modelu z różnymi parametrami k, w zakresie od 1 do 15, i zdecydowałem się na k na podstawie uzyskanej dokładności modelu. Każdy model został wytrenowany i przetestowany na podzbiorach podzielonych 3:1 z oryginalnego zbioru danych. Zestawy treningowe i testowe były takie same dla każdego modelu. Na podstawie uzyskanych wyników wybrano następujące parametry k:

- 7 dla bazowego modelu k-nn
- 2 dla modelu k-nn z zaimplementowanym uśrednianiem reprezentantów
- 2 dla modelu k-nn z zaimplementowanym rzutowaniem kolumn

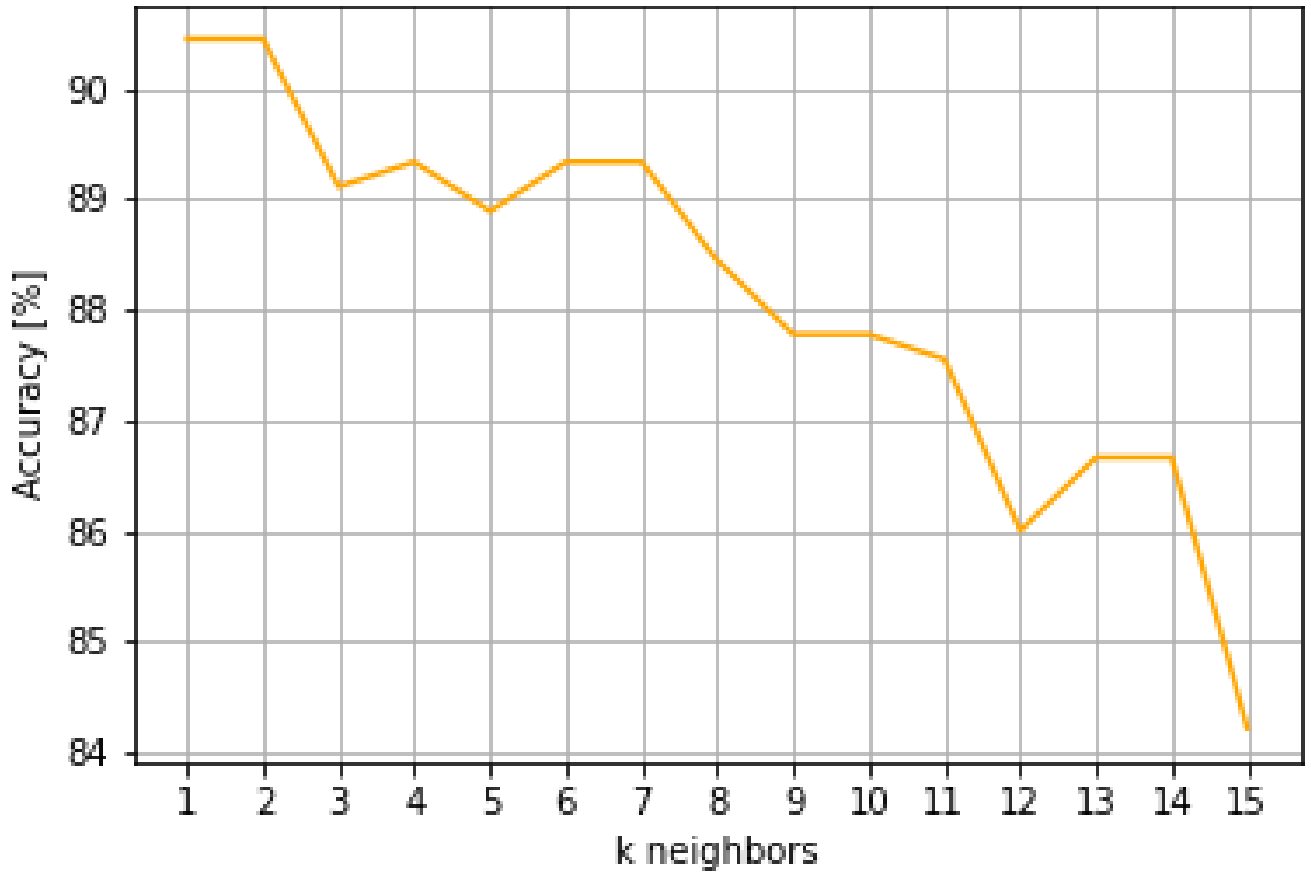
Dobór parametru k – wykresy



Rysunek: Zależność dokładności bazowego modelu k-nn od różnych wartości k

Politechnika Śląska

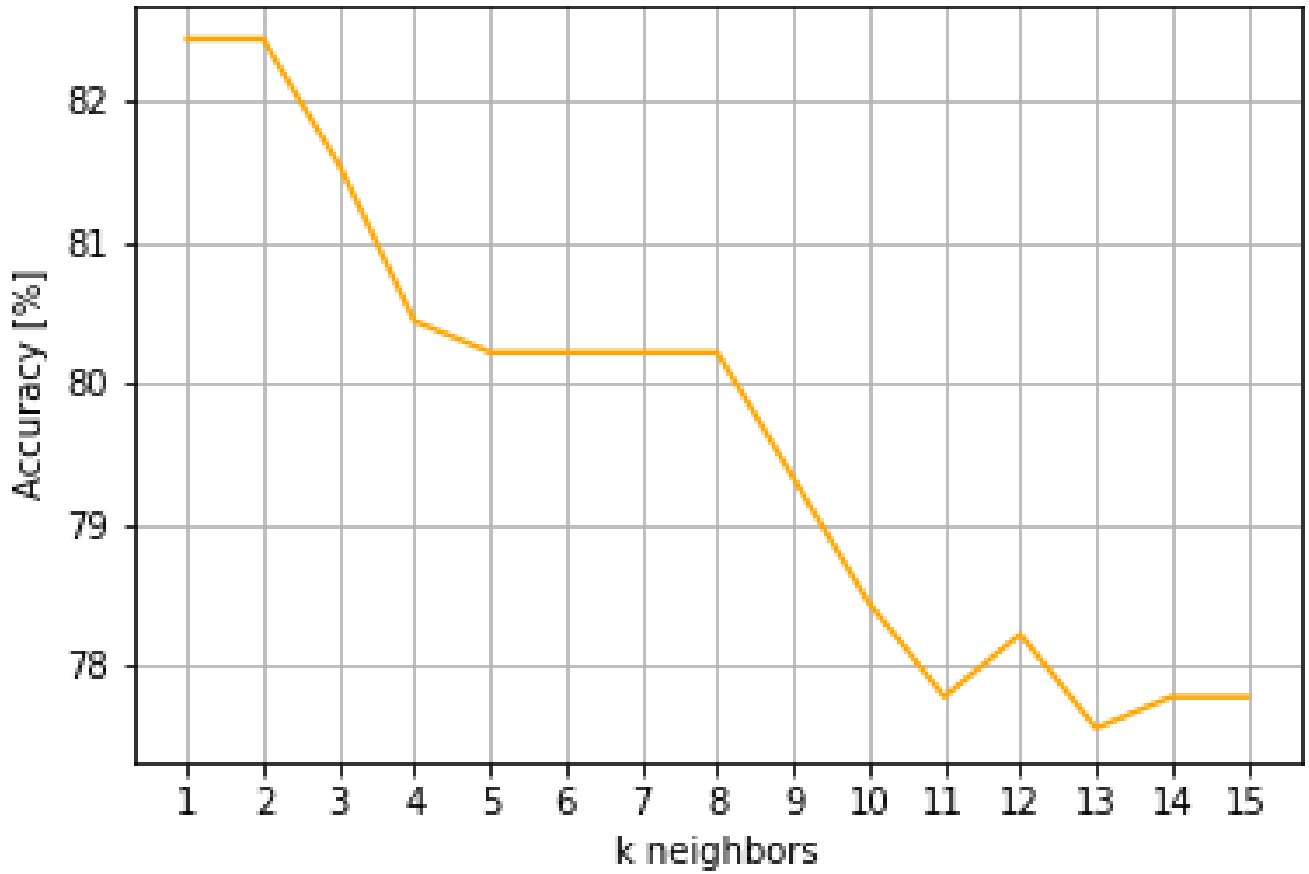
Dobór parametru k – wykresy



Rysunek: Zależność dokładności modelu k-nn z zaimplementowaną metodą średniej reprezentatywnej od różnych wartości k

Politechnika Śląska

Dobór parametru k – wykresy



Rysunek: Zależność dokładności modelu k-nn z zaimplementowaną metodą rzutowania cech od różnych wartości k

Politechnika Śląska

Wyniki

Aby właściwie ocenić wydajność modeli, przeprowadzono 4-krotną walidację krzyżową. Modele zostały ocenione za pomocą czterech wskaźników:

- Dokładność
- Precyzja (macro śr.)
- Pełność (macro śr.)
- wynik f1 (macro śr.)

Wyniki – tabele

Tabela: Dokładność

Model	fold 1	fold 2	fold 3	fold 4
bazowy k-nn	96.44	95.55	97.33	96.21
		śr = 96.38		
reprezentanci	91.76	91.54	93.76	90.42
		śr = 91.87		
rzutowanie	73.50	75.95	81.74	82.63
		śr = 78.45		

Wyniki – tabele

Tabela: Precyzja

Model	fold 1	fold 2	fold 3	fold 4
bazowy k-nn	96.54	95.58	97.40	96.25
		śr = 96.44		
reprezentanci	92.44	92.04	93.93	90.72
		śr = 92.28		
rzutowanie	74.63	76.03	82.15	82.59
		śr = 78.85		

Politechnika Śląska

Dopełnienie

Tabela: Models' recall score (macro)

Model	fold 1	fold 2	fold 3	fold 4
bazowy k-nn	96.43	95.53	97.39	96.14
		śr = 96.37		
reprezentanci	91.69	91.47	93.89	90.41
		śr = 91.87		
rzutowanie	73.83	75.67	81.93	82.40
		śr = 78.46		

Wynik f1

Tabela: Models' f1-score (macro)

Model	fold 1	fold 2	fold 3	fold 4
bazowy k-nn	96.41	95.52	97.39	96.10
		śr = 96.36		
reprezentanci	91.81	91.50	93.88	90.32
		śr = 91.88		
rzutowanie	73.54	75.54	81.82	82.13
		śr = 78.26		

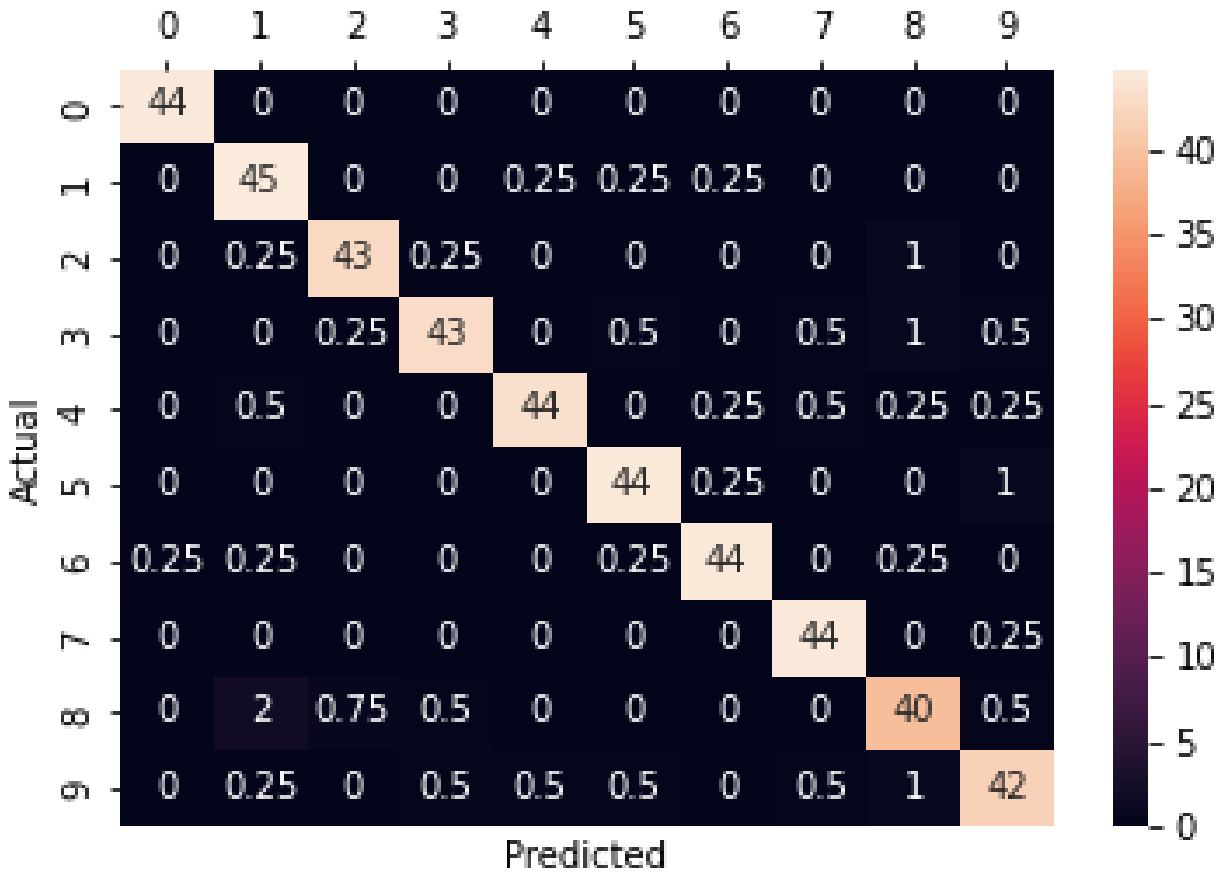
Politechnika Śląska

Wyniki – tabele

Tabela: Zmierzony czas przy sprawdzianie krzyżowym

Model	computational time
bazowy k-nn	19.875s
reprezentanci	1.250s
rzutowanie	19.531s

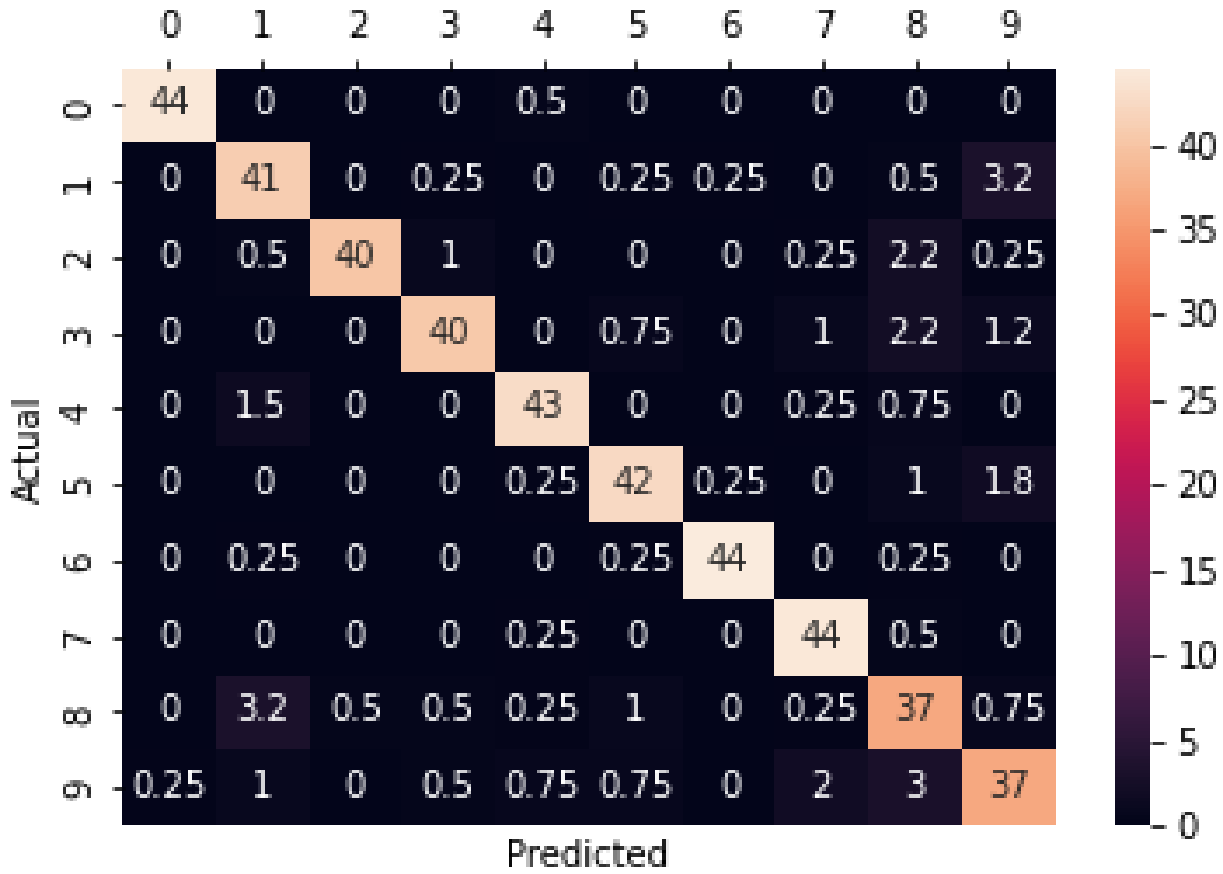
Wyniki – macierze pomyłek



Rysunek: Średnia macierz pomyłek modelu bazowego

Politechnika Śląska

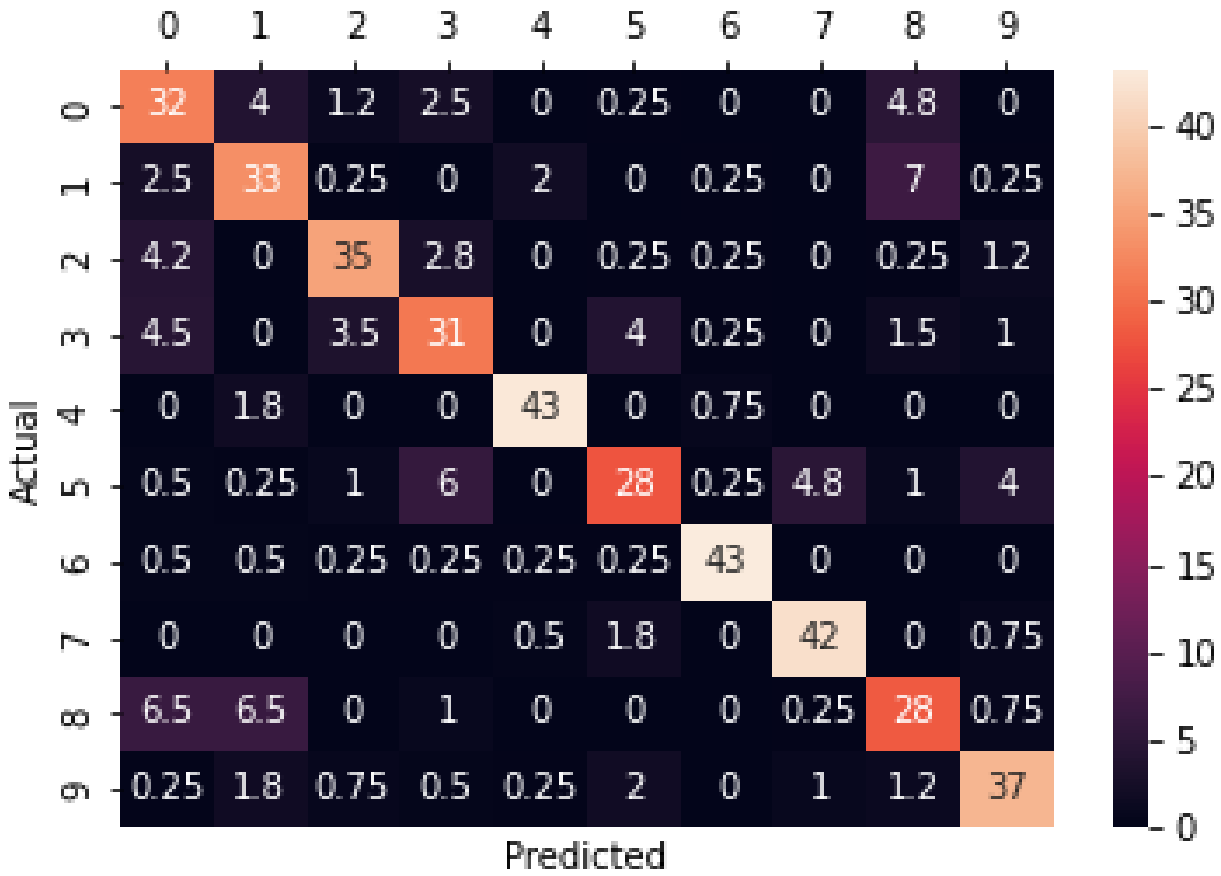
Wyniki – macierze pomyłek



Rysunek: Średnia macierz pomyłek modelu z zaimplementowanymi średnimi reprezentantami

Politechnika Śląska

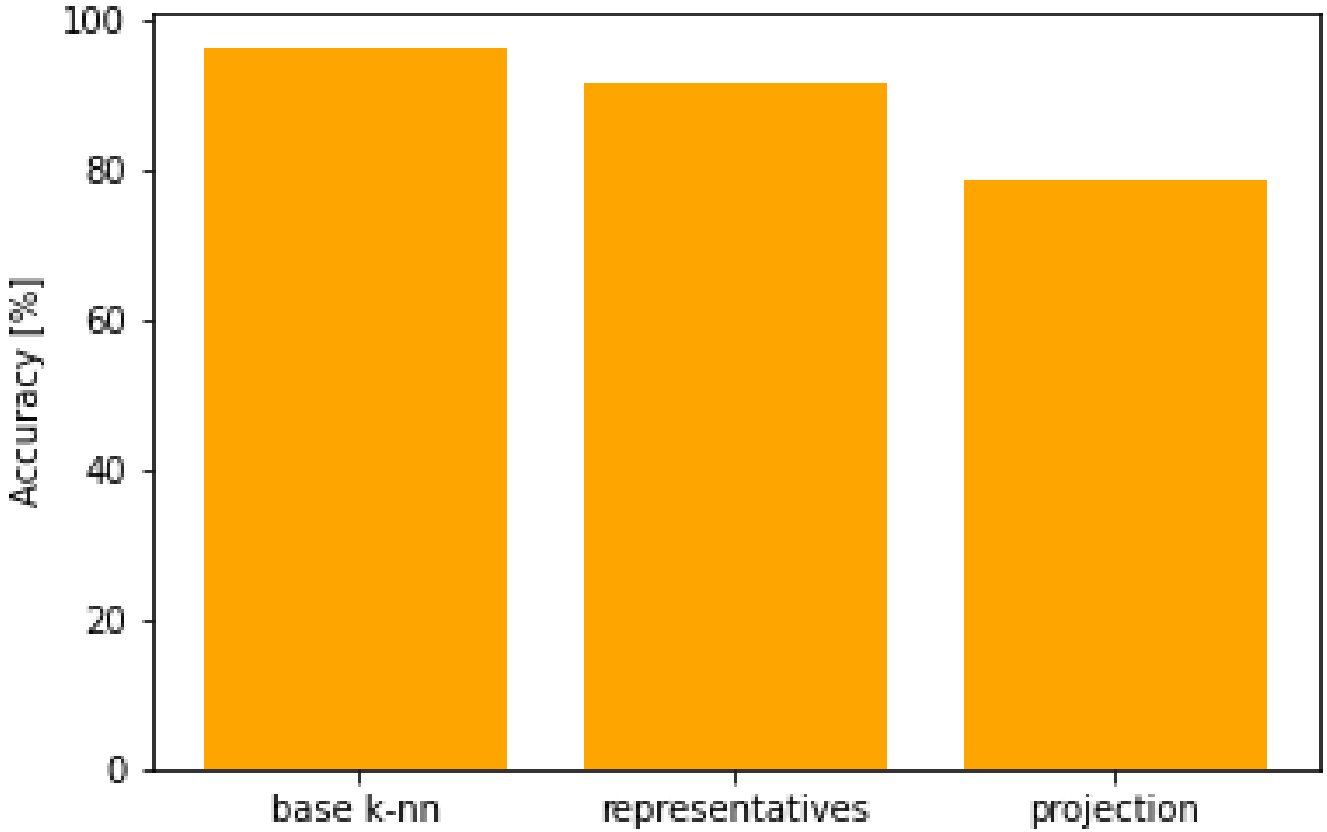
Wyniki – macierze pomyłek



Rysunek: Średnia macierz pomyłek modelu z zaimplementowanym rzutowaniem cech

Politechnika Śląska

Wyniki – porównanie dokładności rozwiązań



Rysunek: Porównanie średniej dokładności modeli

Politechnika Śląska

Wnioski

Model k-nn wypadł wyjątkowo dobrze w danym zadaniu. Nie tylko dokładność, ale wszystkie inne mierzone wskaźniki pozostają na wysokim poziomie. Jak na tak prosty model, wynik jest zdumiewający. Jednak jednym z największych problemów jest jego złożoność obliczeniowa, skutkująca wysokim czasem obliczeń wraz ze wzrostem ilości danych. Problem ten można rozwiązać poprzez implementację średnich reprezentantów. Jak można zauważyć w wynikach, zastosowanie tej metody znacznie skraca czas obliczeń wymagany do przewidywania nowych próbek. Nie jest to jednak pozbawione konsekwencji. Metoda reprezentantów wypadła o około 4.5 punktu procentowego gorzej pod względem dokładności niż model podstawowy. Chociaż kompromis jakości na rzecz szybkości jest nieunikniony, wzrost szybkości znacznie przewyższa spadek dokładności, a metoda może być stosowana jako prawidłowe rozwiązanie, w którym szybkość jest najważniejsza. Metoda rzutowania wypadła raczej słabo zarówno pod względem szybkości, jak i dokładności. Zmniejszenie liczby parametrów nie poprawiło znacząco ani szybkości, ani dokładności.