Podstawy uczenia maszynowego
Eksperymenty z k-NN – raport

Mateusz Kocot

9 kwietnia 2021

Spis treści

1 Wstęp **1**

2 Zbiór danych **1**

3 Zadanie 1 **2**

4 Zadanie 2 **2**
4.1 Wybór konfiguracji 2
4.2 Wyniki 3
4.3 Wnioski 4

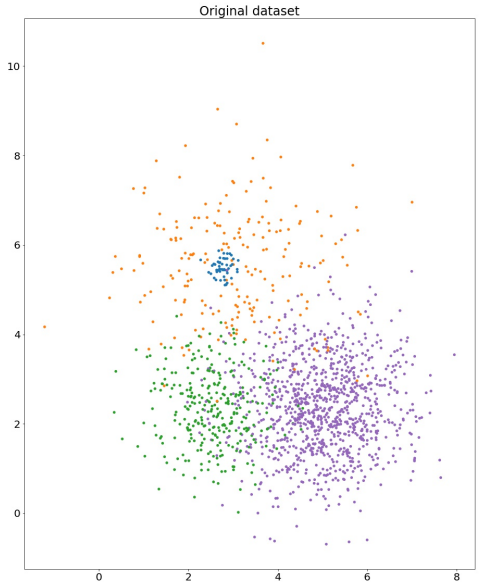
1 Wstęp

Zadanie zaimplementowano w języku Python, w postaci notatnika Jupyter. Wykorzystano pakiety: scikit-learn, NumPy oraz Matplotlib.

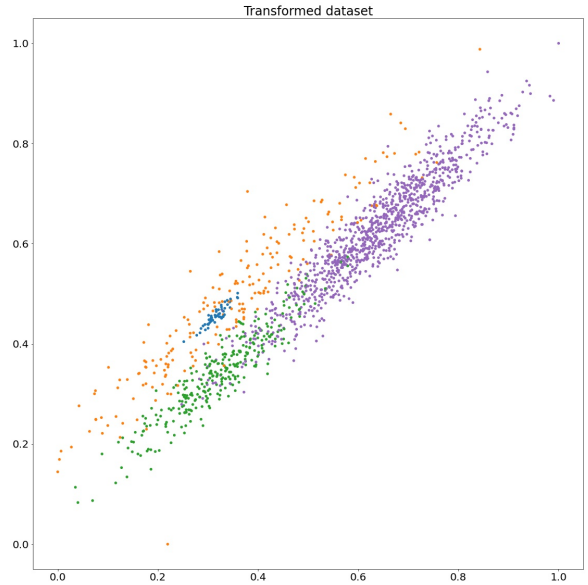
2 Zbiór danych

Najpierw wygenerowano zbiór składający się z 4 klas (rys. 1) oznaczanych kolorami: niebieskim, pomarańczowym, zielonym i fioletowym. Zbiór obserwacji, tj. współrzędnych będzie oznaczany przez macierz $\mathbf{X}$, natomiast zbiór etykiet, czyli klas – przez wektor $\mathbf{y}$. Wygenerowany zbiór ma następujące własności:

- Wszystkie klasy w pewnych miejscach nachodzą na siebie. Najbardziej widać to pomiędzy klasą zieloną i fioletową.
- Klasa niebieska w całości znajduje się wewnątrz klasy pomarańczowej.
- Gęstość obserwacji nie jest równa. Zbiór pomarańczowy jest rzadszy od zielonego, a ten z kolei jest rzadszy od zbioru fioletowego.



Rys. 1: Domyślnie wygenerowany zbiór. Każdy kolor oznacza inną klasę.



Rys. 2: Zbiór po zmodyfikowaniu. Każdy kolor oznacza inną klasę.

Następnie, w celu zmniejszenia regularności, zbiór został zmodyfikowany w dwóch krokach (rys. 2):

- Najpierw zbiór został „rozciągnięty” poprzez pomnożenie współrzędnej x-owej każdego punktu przez 7.
- Później wykonano rotację każdego punktu względem początku układu współrzędnych, o kąt $\pi/6$.
- Na koniec zbiór został przeskalowany do zakresu $(0, 1) \times (0, 1)$.

W ten sposób uzyskano zbiór danych zgodny z wymaganiami.

3 Zadanie 1

Na zbiorze wytrenowano klasyfikator k-NN w odpowiednich konfiguracjach. Metrykę Mahalanobisa sparametryzowano macierzą kowariancji transponowanej macierzy obserwacji $\mathbf{X}$. Wizualizację klasyfikacji poszczególnych elementów przestrzeni przedstawiono na rys. 3.

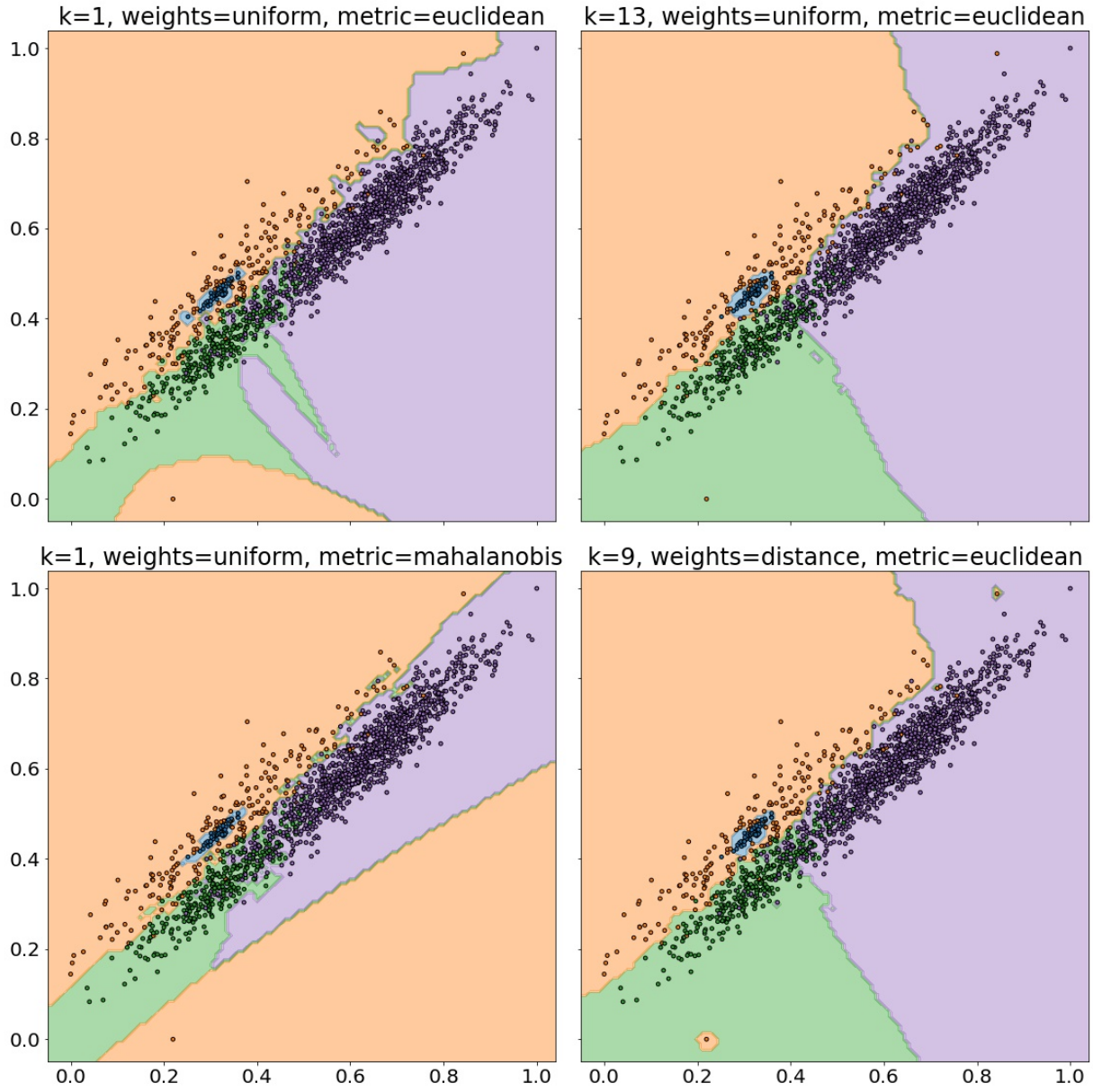
4 Zadanie 2

4.1 Wybór konfiguracji

Na oko, najlepszą konfiguracją wydaje się być metryka Mahalanobisa i głosowanie większościowe. Mimo że wykorzystane zostało $k = 1$ i granica zbiorów fioletowego i zielonego nie jest dobrze odwzorowana, to ta konfiguracja najlepiej radzi sobie z oznaczeniem zbioru niebieskiego oraz wyznaczeniem granicy między zbiorami: fioletowym i zielonym a zbiorem

pomarańczowym. Nieregularne, rozciągnięte wzdłuż prostej rozmieszczenie punktów sprzyja użyciu metryki Mahalanobisa.

Najgorszą konfiguracją wydaje się być metryka Euklidesa i głosowanie większościowe. Granice między klasami są mocno „pofałdowane" dla obu wartości k. Dla $k = 1$, model został przetrenowany.



Rys. 3: Klasyfikacja poszczególnych elementów przestrzeni.

4.2 Wyniki

Wykonano procedurę zgodnie z instrukcją. Wyniki zostały przedstawione w tab. 1. W tabeli przedstawiono także wyniki pozostałych konfiguracji (metryka Mahalanobisa, głosowanie

ważone odległością; metryka Euklidesa, głosowanie ważone odległością). Ciekawsze szczegóły implementacyjne:

- Zbiory dzielone są w taki sposób, by proporcje pomiędzy klasami zostały zachowane.
- Dla każdej konfiguracji, dla każdego k, klasyfikator trenowany jest 20-krotnie na zbiorze walidacyjnym. Później klasyfikator trenowany jest na całym zbiorze treningowym. Cała procedura powtarzana jest 10 razy.

metryka	**głosowanie**	μ	σ
Mahalanobisa	**większościowe**	0.892581	0.008781
Euklidesa	**większościowe**	0.881935	0.013927
Mahalanobisa	waż. odległością	0.889355	0.010896
Euklidesa	waż. odległością	0.884516	0.013440

Tab. 1: Uśredniona skuteczność klasyfikatorów k-NN na wygenerowanym zbiorze.

4.3 Wnioski

Metryka Mahalanobisa z głosowaniem większościowym rzeczywiście okazała się być najlepszą konfiguracją dla wygenerowanego wcześniej zbioru. W przeciwieństwie do metryki Euklidesa, uwzględnia ona nieregularność zbioru.

Wyniki nie różnią się mocno. Większość punktów znajduje się w środku klas, więc zawsze klasyfikowane są poprawnie. Z drugiej strony, część punktów znajduje się na „terytorium" innej klasy i często mogą być klasyfikowane niepoprawnie. Istnieją także przemieszane rejony, gdzie klasyfikacja równie dobrze mogłaby być losowa. Niemniej jednak, użycie metryki Mahalanobisa sprawia, że punkty leżące na granicach równoległych do „prostej rozciągnięcia" są częściej klasyfikowane poprawnie.

Warto zwrócić uwagę na zależność średniej skuteczności i odchylenia standardowego. Im większa skuteczność, tym mniejsze odchylenie. W związku z tym, im lepszy klasyfikator, tym mniejsze wahania wyników w zależności od wylosowanego podziału. Wynika to z faktu, że granica decyzyjna słabszego modelu zmienia się bardziej w zależności od dobranych danych, w stosunku do modelu lepszego.