K-NEAREST NEIGHBORS

KNN

Algorytm kNN jest tzw. algorytmem leniwego uczenia maszynowego. Jest to przykład uczenia nadzorowanego (uczenia z nauczycielem) w którym dostępne dane uczące są etykietowane. Algorytm kNN jest wykorzystywany do budowy klasyfikatorów. Pseudokod algorytmy kNN przedstawia Rys1.

k-Nearest Neighbor
Classify (**X**, **Y**, *x*) // **X**: training data, **Y**: class labels of **X**, *x*: unknown sample
for $i = 1$ **to** m **do**
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set I containing indices for the k smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$

Rys1. Pseudokod algorytmu kNN

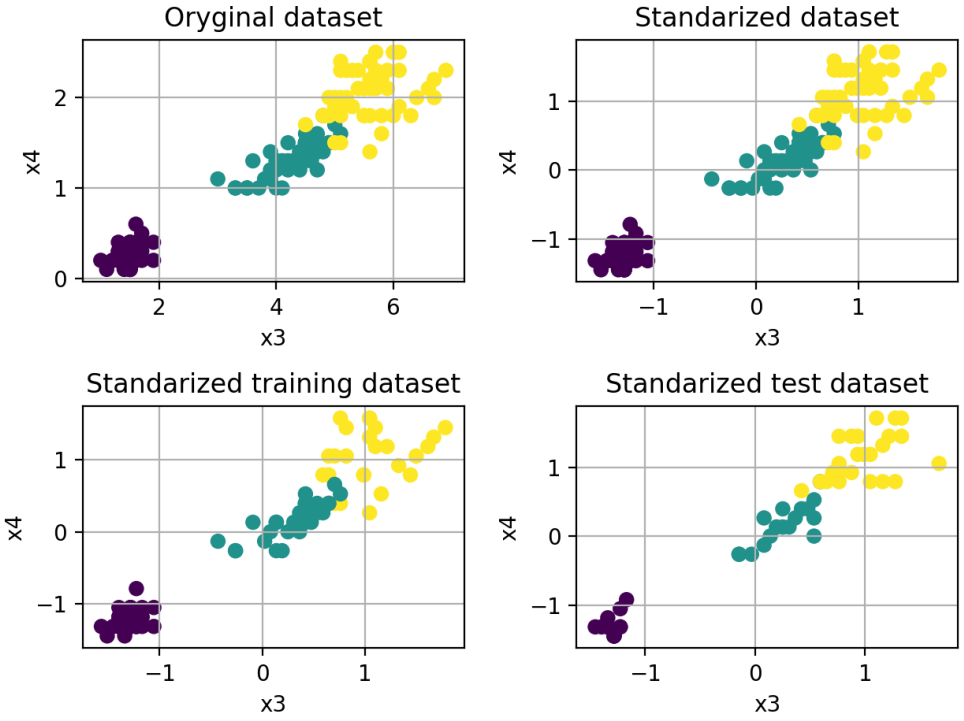
ZADANIA

Korzystając z biblioteki **`sklearn`** (`http://scikit-learn.org`), wykonaj następujące zadania:

1. Załaduj dane „`iris`”
   `datasets`
2. Zwizualizuj dane (3 i 4 wymiar) z wykorzystaniem wykresu punktowego
   `matplotlib` i metoda **`scatter`**
3. Podziel dane na zbiór uczący i testowy
   `train_test_split`

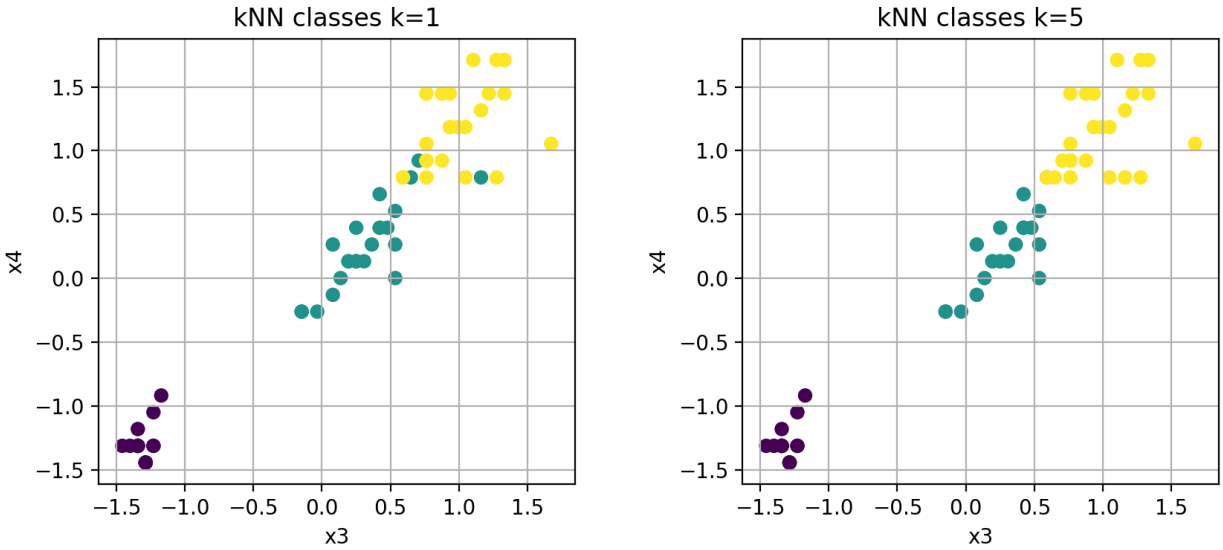
4. Ustandaryzuj[1] dane uczące i testowe
   `StandardScaler`
5. Wytrenuj model kNN na danych uczących
   KNeighborsClassifier
6. Zwizualizuj otrzymane rezultaty dla różnych wartości parametru k=1, k=5
7. Przetestuje działanie klasyfikatora na danych uczących różnej wielkości i różnej wartości parametru k oraz różnych metryk! Porównaj jakość klasyfikacji korzystając z metody **`score`** (metoda instancyjna klasyfikatora). Większa wartość zwracana przez metodę **`score`** świadczy o klasyfikatorze lepszej jakości. Następnie przetestuj działanie metody **`classification_report`** do oceny poprawności klasyfikacji (**`from sklearn.metrics import classification_report`**)

Przykładowe wyniki wizualizacji:



Rys 2. Oryginalny zbiór, ustandaryzowany zbiór, zbiór uczący, zbiór testujący

[1] Standaryzacja to proces wstępnego przetworzenia danych w celu wyzerowania wartości średniej $(\mu = 0)$ cech oraz normalizacji standardowego odchylenia $(std = 1)$ poprzez zastąpienie $x_{new\,j}(i) = \frac{x_j(i) - mean(x_j)}{std(x_j)}$ oraz $std(x_{new_j}) = \sqrt{\frac{\sum_{i=1}^{n}(x_j(i) - mean(x_j))^2}{n-1}}$



Rys 3. Klasyfikacja obiektów dla różnych wartości parametru k.

ZADANIE DODATKOWE:

Zastosuj algorytm kNN dla problemu rozpoznawania ręcznie pisanych cyfr.

Baza ‘digits’ zawiera 1797 przykładów odręcznie napisanych cyfr o wymiarze 8x8 pikseli (sic!:D) w odcieniach szarości. Pojedyncze obrazki zapisane są w postaci macierzowej, a cała baza przykładów tworzy wielowymiarową macierz o wymiarze 1797x8x8. Stąd przed trenowaniem modelu należy ‘spłaszczyć’ obrazki do reprezentacji wektorowej (np. z wykorzystaniem metody reshape)

Cyfry można wyświetlić, korzystając z metody **`matshow`** z modułu **`matplotlib`**:

```
from sklearn.datasets import load_digits
digits = load_digits()
plt.gray()
plt.matshow(digits.images[0])
plt.show()
```