

k -Nearest Neighbors

1 Klasyfikator k -NN

1. Wczytujemy zbiór treningowy i wybieramy dodatnią liczbę całkowitą k .
2. Dany jest przykład do zaklasyfikowania.
3. Ze zbioru treningowego wybieramy k punktów o najmniejszym dystansie od klasyfikowanego przykładu.
4. Klasyfikujemy przykład zgodnie z klasą, która wystąpiła najwięcej razy wśród wybranych punktów.

Zadania

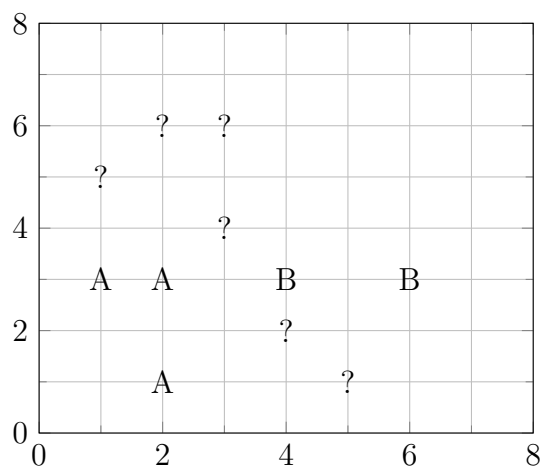
Zadanie 1.

Przypisz następujące przykłady do klasy A lub B przy użyciu metody k -NN z $k = 3$.

Zbiór treningowy: A(1, 3), A(2, 1), A(2, 3), B(4, 3), B(6, 3).

Przykłady do zaklasyfikowania:

- | | | |
|----------|----------|----------|
| • (1, 5) | • (2, 6) | • (3, 4) |
| • (3, 6) | • (4, 2) | • (5, 1) |



Mini-projekt: k -NN

Celem jest napisanie programu, który pobiera następujące argumenty:

k: dodatnia liczba naturalna będąca hiperparametrem k -NN.

train-set: nazwa pliku zawierającego zbiór treningowy w postaci csv.

test-set: nazwa pliku zawierającego zbiór testowy w postaci csv.

Program ma dokonać klasyfikacji k -NN wszystkich obserwacji z pliku **test-set** na podstawie pliku **train-set** oraz podać dokładność (accuracy) tej klasyfikacji (proporcję poprawnie zaklasyfikowanych przykładów testowych).

Program ma też dostarczać testowy interfejs (niekoniecznie graficzny), który umożliwia (zapętlone) podawanie przez użytkownika pojedynczych wektorów do klasyfikacji i podaje ich etykietę k -NN na podstawie **train-set**.

Opcjonalne rozszerzenie (dodatkowy punkt za aktywność): dowolną techniką (excel, python, etc.) zrobić wykres zależności dokładności (accuracy) od wartości k .

Przetestować na danych ze zbiorów treningowego i testowego znajdujących się w plikach **iris.data** i **iris.test.data** (ale program powinien umożliwiać wykorzystanie dowolnego zbioru).

Termin oddania: Następny tydzień.