

## Uczenie maszynowe

### Lista 1

#### Zadanie1:

Napisz skrypt, który generuje przykładowe dane oraz przeprowadza regresję liniową.

Opis działań do realizacji:

- Wygeneruj 100 losowych wartości cechy  $X$  z zakresu od 0 do 10 oraz wartości docelowe  $y$  z liniową zależnością od  $X$  dodając szum gaussowski, co oznacza, że  $y$  ma rozkład normalny z dodanym szumem.
- Dodaj kolumnę z jedynkami (bias) do macierzy cech  $X$  (pozwala to na uwzględnienie stałego składnika w modelu liniowym). Oblicz parametry  $\theta$  (theta) przy użyciu metody najmniejszych kwadratów. parametry te reprezentują współczynniki prostej liniowej, która jest dopasowywana do danych.
- Wyświetl obliczone parametry  $\theta$ .
- Oblicz wartości przewidywane  $y_{pred}$  na podstawie danych  $X$  i parametrów  $\theta$ .
- Oblicz i wyświetl błąd średniokwadratowy (RMSE) i błąd średni absolutny (MAE) między wartościami rzeczywistymi  $y$  a przewidywanymi  $y_{pred}$ . Dokonaj interpretacji otrzymanych wyników.
- Wygeneruj wykres punktowy danych i krzywej reprezentującej dopasowany model regresji liniowej. Na wykresie zaznacz legendę oraz opisy osi i tytuł.

Wynikowy skrypt powinien generować dane, obliczać parametry regresji liniowej, prezentować wyniki i wykres wizualizujący dopasowany model do danych.

Przydatne funkcje MATLAB: randn, ones, pinv, scatter, plot

#### Zadanie2:

Napisz skrypt, który implementuje algorytm k-means dla grupowania danych.

Opis działań do realizacji:

- Wygeneruj przykładowe dane (przy użyciu rozkładu normalnego) składające się z dwóch zestawów punktów skupionych wokół dwóch różnych średnich.
- Ustaw parametry algorytmu k-means, takie jak liczba klastrów i maksymalna liczba iteracji.
- Zaimplementuj algorytm k-means, który przypisuje każdy punkt danych do najbliższego centroidu, aktualizuje położenia centroidów na podstawie przypisanych klastrów i powtarza ten proces przez określoną liczbę iteracji.
- Po zakończeniu procesu grupowania, wizualizuj wyniki poprzez wygenerowanie wykresu punktowego danych, gdzie punkty są kolorowane na podstawie przypisanych klastrów, a centroidy są oznaczone na wykresie krzyżykami. W skrypcie należy również zawrzeć legendę, opisy osi i tytuł na wykresie.

Przydatne funkcje MATLAB: randn, randperm, zeros, mean

### Zadanie 3.

Napisz skrypt, który przeprowadzi analizę danych zestawu Iris. Skrypt powinien zawierać następujące kroki:

Opis działań do realizacji:

- Wczytaj zestaw danych Iris. Przypisz cechy do zmiennej `X` oraz etykiety klas do zmiennej `y`.
- Wyświetl podstawowe informacje o danych, takie jak liczba przykładów, liczba cech i unikalne klasy.
- Stwórz wykres, który wizualizuje długość i szerokość działki kielicha kwiatu dla różnych klas.
- Usuń brakujące wartości, tj. wiersze zawierające braki danych. Znormalizuj cechy do zakresu [0,1].
- Podziel dane na zbiór treningowy (70%) i testowy (30%) przy użyciu funkcji `cvpartition`.
- Wyświetl podsumowanie po przygotowaniu danych, w tym liczbę wierszy po usunięciu brakujących wartości oraz liczbę wierszy w zbiorach treningowym i testowym.

Przydatne funkcje MATLAB: load fisheriris (dane), normalize, gscatter, isnan, cvpartition

### Zadanie 4.

Napisz skrypt, który generuje przykładowe dane, dzieli je na zbiór treningowy i testowy, buduje prosty model sieci neuronowej, klasyfikuje dane testowe za pomocą sieci neuronowej i ocenia skuteczność klasyfikacji.

Opis działań do realizacji:

- Wygeneruj przykładowe dane składające się z dwóch zestawów punktów skupionych wokół dwóch różnych średnich, korzystając z rozkładu normalnego.
- Przypisz etykiety klas dla wygenerowanych danych (pierwszy zestaw punktów oznacz jako klasa "1", a drugi zestaw jako klasa "0").
- Podziel dane na zbiór treningowy (80%) i testowy (20%).
- Zbuduj prosty model sieci neuronowej z jedną warstwą ukrytą zawierającą 10 neuronów.
- Dokonaj treningu sieci neuronowej na danych treningowych, włączając okno pokazujące postęp uczenia.
- Dokonaj klasyfikacji danych testowych przy użyciu wytrenowanej sieci neuronowej, w opcji z zaokrągleniem wyników klasyfikacji do 0 lub 1 i w opcji bez zaokrąglenia.
- Oceń skuteczność klasyfikacji poprzez porównanie przewidywanych etykiet z rzeczywistymi etykietami testowymi.
- Wyświetl informacje o dokładności klasyfikacji.

Przydatne funkcje MATLAB: randn, ones, zeros, cvpartition, patternnet, numel