WSI - ćwiczenie 4.
Regresja i klasyfikacja

13 kwietnia 2023

1 Sprawy organizacyjne

1. Ćwiczenie realizowwane jest samodzielnie.
2. Ćwiczenie wykonywane jest w języku Python.
3. Ćwiczenie powinno zostać oddane najpóźniej na 9. zajęciach. W ramach oddawania ćwiczenia należy zademonstrować prowadzącemu działanie kodu oraz utworzyć pull request (z kodem oraz raportem) który prowadzący będzie mógł komentować.
4. Rozwiązanie ćwiczenia powinno być zgodne z szablonem z repozytorium `https://gitlab-stud.elka.pw.edu.pl/jlyskawa/wsi-template`.
5. Raport powinien być w postaci pliku .pdf, .html albo być częścią notebooka jupyterowego. Powinien zawierać opis eksperymentów, uzyskane wyniki wraz z komentarzem oraz wnioski.
6. Na ocenę wpływa poprawność oraz jakość kodu i raportu.
7. Można korzystać z pakietów do obliczeń numerycznych, takich jak *numpy*
8. Implemenetacja algorytmu powinna być ogólna. W szczególności powinna być możliwa do zastosowania dla dowolnego zbioru danych o dyskretnych wartościach atrybutów.
9. Można skorzystać z pakietów *pandas* i *scikit-learn* w celu załadowania zbioru danych oraz jego podziału na części.

2 Ćwiczenie

Celem ćwiczenia jest implementacja drzew decyzyjnych tworzonych algorytmem ID3 z ograniczeniem maksymalnej głębokości drzewa.

Następnie należy wykorzystać stworzony algorytm do stworzenia i zbadania jakości klasyfikatorów dla zbioru danych Cardio Vascular Disease Detection

(`https://www.kaggle.com/datasets/bhadaneeraj/cardio-vascular-disease-detection`). Klasą jest pole *cardio*. Należy znaleźć taką wartość parametru maksymalnej głębokosci, która da najlepszy wynik.

Część atrybutów w zbiorze danych nie jest dyskretnych - wiek (*age*), waga (*weight*), wzrost (*height*), ciśnienie skurczowe (*ap_hi*), ciśnienie rozkurczowe (*ap_lo*). Należy je zdyskretyzować poprzez wybrane przez siebie podzielenie wartości na zakresy.

Implementacja powinna obsługiwać sytuację, w której w zbiorze trenującym nie ma wszystkich wartości jakiegoś atrybutu.

Należy pamiętać o podziale danych na zbiory trenujący, walidacyjny i testowy. Można użyć w tym celu gotowych funkcji.