

# Indukcja drzew decyzyjnych

Słowa kluczowe: uczenie maszynowe, uczenie nadzorowane, python

## Opis

Implementacja aplikacji w języku Python, która dokona klasyfikacji przypadków przy użyciu algorytmu indukcji drzew decyzyjnych ID3.

## Wymagania obowiązkowe (max 4/6 pkt)

- Algorytm ma być zaimplementowany całkowicie własnoręcznie. Własnoręczność oczywiście w granicach rozsądku, można wykorzystać na przykład biblioteczną funkcję liczącą logarytm.
- Program powinien wypisać na wyjście, możliwie estetycznie, wygenerowane drzewo decyzyjne.
- Należy przetestować algorytm, tzn. sprawdzić wyprodukowane drzewa, dla przynajmniej dwóch zbiorów danych:
  - a) Cars (<https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>)
  - b) Komputery (<http://www.cs.put.poznan.pl/ibladek/students/ed/dane/komputery.csv>)
- Format danych to CSV z przecinkiem jako separatorem.
- Aplikacja powinna na wejście przyjmować następujące argumenty linii poleceń (konsolka):
  - a) **--gainRatio** (na 5.0) flaga mówiąca o tym, że zamiast InfoGain użyte zostanie GainRatio.
  - b) **--minInfoGain** (na 4.0) podana liczba (domyślnie 0.0) to minimalny zysk informacyjny, przy którym dokonywany może być podział na atrybucie (kiedy decyzje nie są jednolite).
  - c) **[file with data]** plik z danymi, obowiązkowy argument
- Przykłady uruchomienia:

```
python3 dd.py komputery.csv
python3 dd.py --minInfoGain 0.1 komputery.csv
python3 dd.py --minInfoGain 0.1 --gainRatio komputery.csv
```
- Do implementacji obsługi linii poleceń należy wykorzystać bibliotekę *argparse*.

## Wymagania dodatkowe cz. 1 (max 5/6 pkt)

- Należy zaimplementować upraszczanie drzewa, tzn. przerywanie dalszych podziałów w danej gałęzi jeżeli zysk informacji jest zbyt mały. Należy wtedy dać w liściu najbardziej prawdopodobną decyzję.

## Wymagania dodatkowe cz. 2 (max 6/6 pkt)

- Należy użyć w obliczeniach ilorazu przyrostu informacji (GainRatio) zamiast InfoGain.