

Inteligentne Systemy Wspomagania Decyzji

Laboratorium 1

mgr inż. Andrii Shekhovtsov

7 października 2025

Zasady oceniania

Program, który powstał w ramach tego zadania, powinien zostać przesłany za pośrednictwem Moodle jako plik tekstowy z kodem w Python w formacie `.py`. W przesyłanym pliku z kodem proszę umieścić na pierwszej linii komentarz ze swoim imieniem, nazwiskiem, numerem albumu. Dodatkowo, należy przygotować i wysłać raport w formacie `.pdf`, wymagania co do raportu są opisane w zadaniach.

Plik ten należy przesłać za pośrednictwem systemu Moodle w wyznaczonym tam terminie.

W przesyłanym pliku z kodem proszę umieścić na pierwszej linii komentarz ze swoim imieniem, nazwiskiem, numerem albumu.

Przykładowe formatowanie pliku:

```
1 # Jan Kowalski, nr. alb. 12345
2
3 # tutaj umieszczamy cały kod programu...
```

UWAGA: Termin oddania zadania jest ustawiony w systemie Moodle. W przypadku nieoddania zadania w terminie, uzyskana ocena będzie zmniejszana o 0,5 za każdy zaczęty tydzień opóźnienia. Zadania oddawane później niż miesiąc po terminie ustawionym na Moodle mogą zostać niesprawdzone lub ocenione na ocenę niedostateczną.

UWAGA: W przypadku wysłania zadania w formie niezgodnej z opisem w instrukcji prowadzący zastrzega prawo do wystawienia oceny negatywnej za taką pracę. Przykład: wysłanie `.zip` lub `.pdf` tam, gdzie był wymagany plik tekstowy z rozszerzeniem `.py`.

UWAGA: W przypadku podejrzeń wysłania zadania wykonanego w dużym stopniu przy pomocy AI (LLMs, ChatGPT, etc.), prowadzący zastrzega prawo do wystawienia oceny negatywnej za taką pracę. Ocenę tą można poprawić tylko w trakcie dopytki ustnej.

1 Zadanie

Celem zadania jest samodzielne przygotowanie problemu decyzyjnego oraz jego rozwiążanie metodą TOPSIS, z udokumentowaniem wyników w raporcie.

1. Przygotowanie problemu decyzyjnego:

- wybrać temat problemu decyzyjnego (np. wybór samochodu, wybór dostawcy, wybór laptopa do pracy),
- określić kryteria oceny (C_j), ich typ (zysk/koszt),
- wskazać alternatywy (A_i),
- ustalić wagę w_j kryteriów.

2. Implementacja metody TOPSIS w Pythonie:

- zaimplementować metodę TOPSIS przy użyciu operacji wektorowych biblioteki `numpy`,
- normalizację przeprowadzić według jednej z metod przedstawionych na wykładzie (zalecane: min-max lub wektorowa),

- nie wolno korzystać z gotowych implementacji bibliotek MCDA.

3. Rozwiążanie problemu:

- zastosować przygotowaną implementację TOPSIS do własnego problemu decyzyjnego,
- uzyskać ranking alternatyw (uszeregowanie alternatyw od najlepszej (największe C_i) do najgorszej (najmniejsze C_i)).

4. Analiza wyników:

- porównać ranking alternatyw z oczekiwaniami,
- ocenić, czy wyniki są logiczne i spójne z przyjętymi kryteriami.

5. Wizualizacja wyników:

- przygotować wykres słupkowy (np. w Pythonie z użyciem `matplotlib`),
- oś X: kolejne alternatywy (A_1, A_2, \dots lub nazwy własne),
- oś Y: wartości rankingowe (wysokość słupków odpowiada wartości C_i).

6. Raport końcowy:

- opisać problem decyzyjny i podać dane wejściowe,
- przedstawić kolejne etapy obliczeń metody TOPSIS (w postaci tabelek lub screenshotów z obliczeń w Pythonie),
- zaprezentować wyniki: ranking oraz wykres,
- przygotować podsumowanie z wnioskami: czy wyniki są zgodne z oczekiwaniami, jakie są mocne i słabe strony metody w tym przypadku.

2 Skala ocen

- Ocena 3.0: Zadania 1-3,
- Ocena 4.0: Zadania 1-5,
- Ocena 5.0: Zadania 1-6.