

Wprowadzenie do biblioteki `pymcdm`

lab 4

1 Wprowadzenie

Biblioteka `pymcdm` (ang. *Python Multiple-Criteria Decision Making*) umożliwia rozwiązywanie zagadnień wielokryterialnego podejmowania decyzji w języku Python. Dostarcza ona różnorodne metody, takie jak TOPSIS, SPOTIS, VIKOR, PROMETHEE oraz narzędzia służące do normalizacji danych, wyznaczania wag i agregowania kryteriów.

Celem niniejszych zajęć jest zapoznanie się z funkcjonalnością biblioteki `pymcdm` oraz wykonanie przykładowych analiz i porównań metod decyzyjnych.

2 Cel zadania

1. Przyswojenie podstawowych koncepcji z zakresu wielokryterialnego podejmowania decyzji (MCDM).
2. Zapoznanie się z dokumentacją biblioteki `pymcdm` i jej strukturą.
3. Przećwiczenie implementacji wybranych metod MCDM (koniecznie TOPSIS i SPOTIS, opcjonalnie inne np. VIKOR).
4. Porównanie wyników różnych metod decyzyjnych dla określonego zbioru danych.

3 Zakres zadania

1. Instalacja i konfiguracja biblioteki `pymcdm`.
2. Przygotowanie przykładowego zestawu danych decyzyjnych wraz z odpowiednimi wagami oraz informacją o tym, które kryteria mają być maksymalizowane/minimalizowane.

3. Zastosowanie co najmniej dwóch metod MCDM (koniecznie TOPSIS i SPOTIS) do wyznaczenia rankingu alternatyw.
4. (Opcjonalnie) Wykorzystanie innych metod decyzyjnych dostępnych w `pymcdm` (np. VIKOR, PROMETHEE).
5. Wykorzystanie funkcjonalności biblioteki do normalizacji danych i/lub wyznaczania wag (np. metodą entropy lub AHP).
6. Analiza porównawcza uzyskanych wyników i wniosków z porównania.

4 Instrukcje – krok po kroku

4.1 Instalacja biblioteki `pymcdm`

1. Zainstaluj bibliotekę `pymcdm` (np. przez `pip install pymcdm`).
2. Zapoznaj się z oficjalną dokumentacją biblioteki (dostępną m.in. w repozytorium GitHub lub dokumentacji online).

4.2 Przygotowanie danych

1. Zdefiniuj macierz decyzyjną (np. w postaci listy list lub obiektu typu `numpy.ndarray`), w której wiersze będą odpowiadały rozpatrywanym alternatywom (np. opcjonalnie inwestycyjnym), a kolumny kryteriom (np. koszty, zyski, czas, ryzyko itp.).
2. Określ wektor wag dla poszczególnych kryteriów (jeśli są one znane a priori) lub skorzystaj z metod biblioteki `pymcdm` do ich wyznaczenia.
3. Ustal, które kryteria mają być maksymalizowane, a które minimalizowane (np. koszty minimalizowane, zyski maksymalizowane).

4.3 Wykorzystanie metod decyzyjnych

1. Zimportuj z biblioteki `pymcdm` odpowiednie metody, w szczególności TOPSIS i SPOTIS.
2. Dokonaj normalizacji danych (np. min-max, wektorowa) lub skorzystaj z funkcji dostępnych w `pymcdm.normalizations`.
3. Uruchom wybrane metody MCDM (TOPSIS i SPOTIS), przekazując macierz decyzyjną, wagi i informację o typach kryteriów.

- Odbierz wyniki (ranking lub oceny punktowe dla każdej alternatywy) i zapisz je w wygodnej do analizy formie (np. lista lub DataFrame).

4.4 Porównanie wyników i wnioski

- Uruchom co najmniej dwie różne metody decyzyjne (obowiązkowo TOP-SIS i SPOTIS) na tym samym zbiorze danych.
- Porównaj otrzymane rankingi – sprawdź, czy są podobne, czy też mocno się różnią. Jeśli się różnią, spróbuj znaleźć przyczyny.
- Dokonaj interpretacji wyników: która alternatywa okazała się najlepsza i dlaczego.
- Przygotuj krótki raport z wnioskami w pliku tekstowym (np. Markdown lub PDF).

5 Raport i przesyłanie zadania

- Opracowany kod (skrypt Jupyter lub pliki .py) wgraj do repozytorium GitHub.
- Dodaj plik z krótkim raportem (np. `raport.md` lub `raport.pdf`) opisujący konfigurację analizy, uzyskane wyniki i wnioski.
- Link do repozytorium oraz pliki załącz do zadania w systemie Moodle.
- Termin oddania i szczegółowe warunki oceny znajdziesz w opisie kursu.

6 Kryteria oceny

- Poprawna instalacja i użycie biblioteki `pymcdm`.
- Prawidłowo przygotowana macierz decyzyjna, wagi i typy kryteriów (max/min).
- Zastosowanie metod decyzyjnych: TOPSIS oraz SPOTIS (obowiązkowe) i ewentualnie inne dostępne w `pymcdm`.
- Uwzględnienie normalizacji oraz (opcjonalnie) wyznaczenie wag metodami `pymcdm`.
- Jasne i zrozumiałe wnioski z eksperymentu, zawarte w raporcie.

6. Złożenie kodu i raportu w repozytorium GitHub oraz na Moodle.
7. (Opcjonalnie) Dodatkowe punkty za wykorzystanie więcej niż dwóch metod, automatyczne porównanie lub wizualizację wyników.