

Algorytmy genetyczne

ZADANIA KWALIFIKACYJNE

Zuzanna Opala

opala.zuzanna@gmail.com

Poniżej znajduje się 6 zadań (a właściwie jedno zdanie podzielone na 6 części). Zaimplementuj ich rozwiązanie w Pythonie¹, a kod źródłowy oraz krótkie opisy do każdego punktu (na przykład wyniki swoich porównań, jakieś wnioski, co kierowało Tobą przy wyborze pewnych elementów i oczywiście odpowiedzi na pytania jeśli takie w zadaniach się znajdują) prześlij na mojego maila. Postaraj się zrobić wszystkie zadania, a jeśli masz z czymś problem nie bój się napisać do mnie wcześniej – spróbuję Ci pomóc. Zanim zaczniesz przejrzyj przydatne materiały na stronie warsztatów.

Pełen przykład do zadania dostępny tu: <https://repl.it/I4Wm/2>

ZADANIE 1. Wyobraź sobie, że prowadzisz sklep GMO i chcesz stworzyć „bazę” swoich przyszłych pracowników.

- Stwórz klasę, której pola będą zawierały cechy (co najmniej 5), na które chcesz zwrócić uwagę przy zatrudnieniu.

Przykład:

```
1. class kandydat:
2.     def __init__(self, symp, punkt):
3.         self.sympatycznosc = symp # w skali od 0 do 10
4.         self.punktualnosc = punkt # w skali od 0 do 10
```

- Stwórz 10-elementową tablicę takich obiektów i wypełnij ją losowymi danymi.

ZADANIE 2. Zastanów się jak chcesz oceniać pracowników (na podstawie pól z Twojej klasy) i stwórz funkcję, której wynikiem będzie liczba z zakresu od 0 do 1 wyrażająca jak bardzo dany pracownik spełnia Twoje wymagania.

Przykład:

```
1. def ocena(kand): # zauważ, że ta implementacja nie zwraca wartości z zakresu 0-1
2.     ideal_symp = 8
3.     ideal_punkt = 10
4.     return -(abs(ideal_symp - kand.sympatycznosc) + abs(ideal_punkt - kand.punktualnosc))
```

ZADANIE 3. Sprawdź którego ze swoich 10 potencjalnych pracowników byś zatrudnił. Wykonaj program kilka razy i sprawdź jak bardzo różnią się wyniki. Spróbuj tego samego dla 100 i 1000 pracowników. Co możesz powiedzieć o wynikach dla różnych wartości?

ZADANIE 4. Napisz drugą funkcję oceny, która zamiast liniowo będzie oceniać na podstawie kwadratu różnicy.

Przykład:

```
1. def ocena2(kand):
2.     return -((ideal_symp - kand.sympatycznosc) ** 2 +
3.              (ideal_punkt - kand.punktualnosc) ** 2)
```

ZADANIE 5. Oceń swoich 10 potencjalnych pracowników używając funkcji liniowej i kwadratowej. Czy ich kolejność się różni? Dlaczego? Która kolejność, Twoim zdaniem, lepiej oddaje Twoje wymagania?

ZADANIE 6. Zaproponuj inne funkcje oceny i porównaj rezultaty. Opisz swoje pomysły i wnioski.

¹ Może być też inny język, przy czym wszystkie przykłady są napisane w Pythonie (będzie tak też na warsztatach), a także jest dużo bardziej prawdopodobne, że będę w stanie pomóc w razie jakiś problemów implementacyjnych, więc myślę, że dobrze jest wykorzystać te zadania jako okazję do nauki Pythona.