



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**NAZWA WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII
BIOMEDYCZNEJ**

Informatyka w sterowaniu i zarządzaniu

Optymalizacja wielokryterialna

*Aplikacja do rekomendacji filmów z wykorzystaniem metod
optymalizacji wielokryterialnej*

<i>L.p.</i>	Członek	Numer albumu	Adres e-mail
<i>1</i>	Artur Mzyk	400658	arturmzyk@student.agh.edu.pl
<i>2</i>	Joanna Nużka	400561	joannanuzka@student.agh.edu.pl
<i>3</i>	Adrian Poniatowski	401346	adrianponiat@student.agh.edu.pl

Spis treści

1. Wstęp.....	3
1.1. Cel projektu	3
1.2. Akwizycja danych.....	3
2. Badany problem.....	3
2.1. Opis problemu	3
3. Propozycja rozwiązania.....	4
3.1. TOPSIS.....	4
3.2. UTA	4
3.3. Metoda zbiorów referencyjnych	4
4. Aplikacja	4
5. Eksperymenty	5
6. Podsumowanie/wnioski.....	5
6.1. Zrealizowane punkty.....	5
6.2. Napotkane problemy.....	5
6.3. Kroki dalszego rozwoju	5
7. Spis literatury.....	5
8. Podział pracy	6

1. Wstęp

1.1. Cel projektu

Celem projektu było zbudowanie aplikacji do rekomendacji użytkownikowi filmów na podstawie jego preferencji. Wykorzystywane są poznane na zajęciach metody optymalizacji wielokryterialnej, głównie metody rankingowe, takie jak Topsis, UTA czy metoda zbiorów referencyjnych. Użytkownikowi proponowane jest dziesięć filmów najbardziej zbliżonych do jego wymagań, na które składają się:

- gatunek,
- rok produkcji,
- obsada,
- ogólna ocena użytkowników portalu.

1.2. Akwizycja danych

Dane zaczerpnięte zostały z ogólnodostępnej bazy filmów IMDB za pośrednictwem pakietu Pythona IMDbPy. Oprócz zbioru filmów zawiera ona kluczowe informacje na ich temat i udostępnia szeroki zakres funkcji ułatwiających filtrowanie.

Dane zostały na początku poddane preprocessingowi i zweryfikowane pod względem kompletności. Interfejs graficzny pozwala na filtrowanie filmów ze względu na gatunek.

2. Badany problem

2.1. Opis problemu

Ważne jest zdefiniowanie funkcji dopasowania określającej, jak dobre jest dopasowanie danego filmu do podanych przez użytkownika preferencji. Pozwala ona ocenić film na podstawie trzech kryteriów, z których każde ma określoną skalę:

- liczba pokrywających się z preferencjami aktorów i reżyserów: 0-10,
- ocena na portalu IMDB: 0-10,
- odległość od średniego roku produkcji obejrzanych przez użytkownika filmów: 0-3 (danemu interwałowi przypisana jest odpowiednia ocena ze skali).

3. Propozycja rozwiązania

3.1. TOPSIS

3.2. UTA

3.3. Metoda zbiorów referencyjnych

4. Aplikacja

Wykorzystany został język wysokiego poziomu Python, gdyż pozwala na proste tworzenie interfejsów graficznych i łatwy dostęp do bazy danych filmów IMDB.

Dane pobrano, wykorzystując REST API z dedykowanej do filmów biblioteki Pythona IMDbPy, a następnie poddane je parsowaniu, co zostało przedstawione na *Rys. 1*.

Zbudowany został interfejs graficzny pozwalający na intuicyjne modyfikowanie preferencji w celu znalezienia odpowiedniego filmu. Ponadto, umożliwia on pobranie listy filmów na podstawie preferowanego gatunku. Szkic interfejsu znajduje się na *Rys. 2*.

```
def get_top_movies(genre: str, n_movies: int) -> Dict[str, str]:

    # Stworzenie obiektu klasy IMDb
    imdb = IMDb()

    # Konkatenacja URL potrzebnego do pobrania danych
    criteria = {'genres': genre, 'count': str(n_movies), 'sort': "num_votes,desc"}
    params = '&'.join(['s=%s' % (k, v) for k, v in criteria.items()])
    url = imdb.urls['search_movie_advanced'] % params

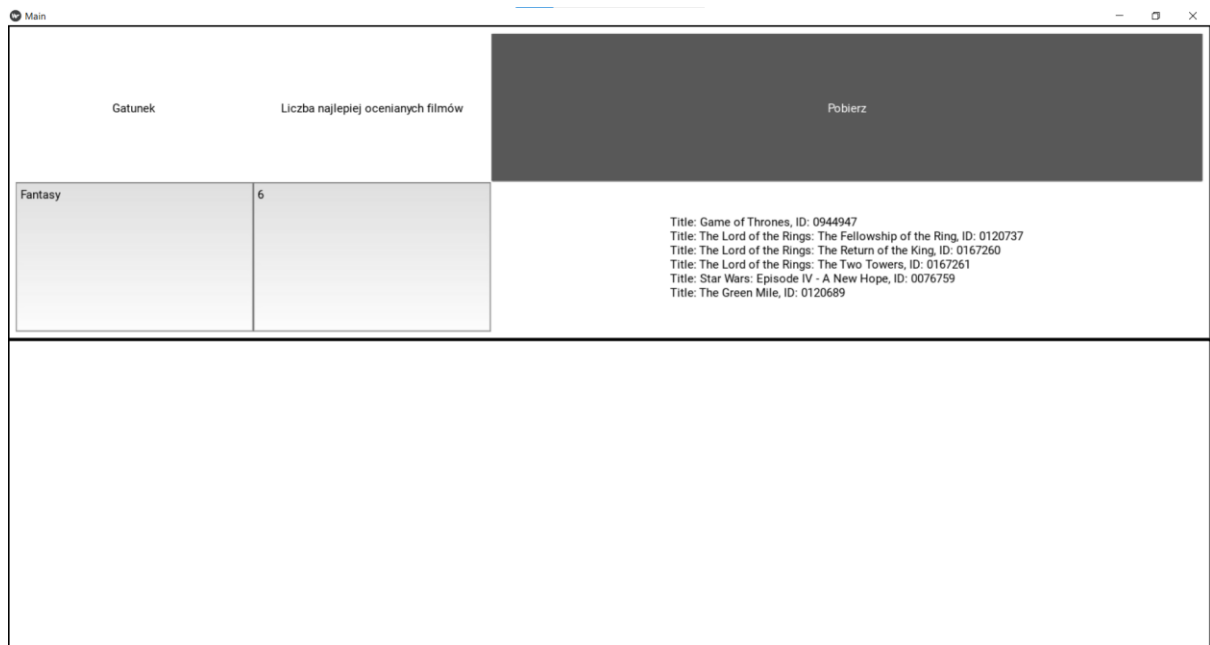
    # Pobranie danych
    content = imdb.retrieve(url)

    # Parsowanie danych
    soup = BeautifulSoup(content, 'html.parser')
    response = json.loads(soup.find('script', {"id": "__NEXT_DATA__"}).text)
    movies: Dict[str, str] = {}

    for i in range(n_movies):
        item = response['props']['pageProps']['searchResults']['titleResults']['titleListItems'][i]
        movies[item['titleId'][2:]] = item['titleText']

    return movies
```

Rys. 1. Skrypt do pobrania i parsowania danych



Rys. 2. Interfejs graficzny

5. Eksperymenty

6. Podsumowanie/wnioski

6.1. Zrealizowane punkty

6.2. Napotkane problemy

6.3. Kroki dalszego rozwoju

7. Spis literatury

<https://www.sciencedirect.com/science/article/pii/S187705091631273X>

https://www.researchgate.net/publication/236153959_Decision_Support_Systems_Based_on_Reference_Sets

<https://imdbpy.readthedocs.io/en/latest/>

8. Podział pracy

Funkcjonalność	Artur [wkład w %]	Joanna [wkład w %]	Adrian [wkład w %]
Preprocessing danych	-	50	50
Wstępne GUI do pobierania filmów	100	-	-