



**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**  
**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>	<b>Członek</b>	<b>Numer albumu</b>	<b>Adres e-mail</b>
1	Artur Mzyk	400658	arturmzyk@student.agh.edu.pl
2	Joanna Nużka	400561	<a href="mailto:joannanuzka@student.agh.edu.pl">joannanuzka@student.agh.edu.pl</a>
3	Adrian Poniatowski	401346	<a href="mailto:adrianponiat@student.agh.edu.pl">adrianponiat@student.agh.edu.pl</a>

## 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 .....	5
4. Aplikacja.....	6
5. Eksperymenty .....	7
5.1. Eksperymenty 1 .....	7
5.2. Eksperymenty 2.....	22
6. Podsumowanie/wnioski .....	33
6.1. Zrealizowane punkty .....	33
6.2. Napotkane problemy .....	33
6.3. Kroki dalszego rozwoju .....	33
7. Spis literatury .....	34
8. Podział pracy.....	34

# 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 kilka filmów najbardziej zbliżonych do jego wymagań, na które składają się:

- gatunek,
- rok produkcji,
- obsada,
- ilość zastosowanych efektów specjalnych,
- autor książki, na podstawie której powstał film,
- 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

Badamy problem optymalizacji wielokryterialnej, w którym dopasowanie filmu do preferencji podanych przez użytkownika oceniane jest na podstawie 6 kryteriów:

- liczba pokrywających się z preferencjami aktorów i reżyserów – za każde pokrycie wartość zwiększana jest o 1, kryterium maksymalizowane,
- liczba pokrywających się z preferencjami autorów książek, na podstawie których powstały filmy – za każde pokrycie wartość zwiększana jest o 1, kryterium maksymalizowane,
- popularność obsady – oceniana na podstawie liczby filmów, w których grał główny aktor, kryterium jest maksymalizowane,
- ocena na portalu IMDB, kryterium jest maksymalizowane,

- różnica między rokiem produkcji filmu a preferowanym – kryterium jest minimalizowane,
- liczba firm odpowiedzialnych za efekty specjalne – kryterium jest maksymalizowane.

### 3. Propozycja rozwiązania

#### 3.1. TOPSIS

Metoda TOPSIS to metoda optymalizacji wielokryterialnej bazująca na wyznaczaniu odległości badanych punktów od danego punktu idealnego i antyidealnego dla badanego problemu. Na podstawie tych odległości wyliczany jest współczynnik scoringowy służący do uszeregowania wyników. Największa wartość tego współczynnika odpowiada najlepszemu obiektowi dla danego problemu.

Zaimplementowana przez nas metoda TOPSIS składa się z opisanych dalej etapów i elementów. Pierwszym krokiem jest zbadanie kierunków optymalizacji dla każdego kryterium. Dla kryterium maksymalizowanego następuje odwrócenie wszystkich wartości tak, aby otrzymać problem minimalizacji. Następnie dokonywane jest odfiltrowanie punktów zdominowanych a dalsza analiza występuje tylko dla punktów niezdominowanych. Punkty niezdominowane zapisywane są w macierzy ewaluacji. Macierz ta jest normalizowana przy użyciu normy euklidesowej oraz skalowana za pomocą przemnożenia przez wektor wag. Wagi są wprowadzane przez użytkownika przy pomocy interfejsu. Następnie wyznaczany jest punkt idealny (wektor najmniejszych wartości dla każdego kryterium) oraz antyidealny (wektor największych wartości) przeskalowanej macierzy. W kolejnym kroku dla każdego z punktów niezdominowanych mierzone są odległości od punktu idealnego i antyidealnego przy użyciu metryki euklidesowej. Na podstawie otrzymanych odległości obliczany jest współczynnik topsis z zależności:

$$s = \frac{d_w}{d_w + d_b}, \text{ gdzie:}$$

$s$  – współczynnik topsis

$d_w$  – odległość od punktu antyidealnego

$d_b$  – odległość od punktu idealnego.

Ostatnim krokiem jest uszeregowanie punktów niezdominowanych według malejących wartości współczynnika oraz zwrócenie wyników.

#### 3.2. UTA

Metoda UTA jest narzędziem analizy decyzji, która pomaga w podejmowaniu decyzji w przypadku optymalizacji wielokryterialnej. Jest używana do oceny różnych alternatyw w kontekście ich wpływu na cele i preferencje osoby decydującej. Za pomocą oceny każdego z kryterium czy też każdej z alternatyw, jest tworzony ranking z uszeregowanymi wynikami.

Zaimplementowana przez nas metoda UTA, a konkretniej UTA Star składa się z opisanych dalej etapów i elementów. Pierwszym niezbędnym działaniem w tej metodzie jest zidentyfikowanie kryteriów, które są istotne dla decydenta oraz celów decyzyjnych, a więc ich maksymalizacja bądź też minimalizacja. Następnie dla każdego kryterium oraz propozycji rozwiązań (alternatyw) przypisywane są wartości oraz wagi, które odzwierciedlają kolejno stopień spełnienia danego kryterium przez konkretne alternatywy oraz ważność poszczególnych kryteriów w kontekście osiągnięcia celów. Kolejnym krokiem jest dokonanie oceny każdej alternatywy na podstawie przypisanych wartości, które uwzględniają preferencje osoby decydującej oraz wpływ każdej możliwości rozwiązania na maksymalizację bądź minimalizację czyli osiągnięcie celów. Ostatnim działaniem w realizacji metody UTA jest użycie matematycznych operacji, na przykład sumy ważonej do agregacji ocen dla każdej alternatywy, uzyskanie ogólnej oceny oraz końcowo otrzymanie wynikowego rankingu z ułożonymi propozycjami od tych najlepiej spełniających preferencje decydenta, do tych najmniej.

Metoda UTA, według informacji znalezionych w internecie, jest metodą dość często stosowaną, szczególnie użyteczna w sytuacjach z wieloma kryteriami decyzyjnymi.

### 3.3. Metoda zbiorów referencyjnych

Metoda zbiorów referencyjnych, bądź też metoda zbiorów odniesienia (RSM), to technika analizy decyzji używana w procesie podejmowania decyzji w optymalizacji wielokryterialnej. Podejście to jest oparte na porównywaniu alternatyw względem jakiegoś zdefiniowanego zestawu wzorców, które mają za zadanie reprezentować preferencje osoby podejmującej decyzję w odniesieniu do poszczególnych kryteriów.

Zaimplementowana przez nas metoda RSM składa się z opisanych dalej etapów i elementów. W pierwszym kroku dokonywana jest identyfikacja kryteriów decyzyjnych, które powinny odzwierciedlać cele i priorytety decydenta, a które są istotne dla oceny propozycji końcowych. Następnie dla każdego kryterium tworzone są zbiory referencyjne reprezentujące skrajne wartości tego kryterium. Mogą to być wartości ekstremalne, optymalne lub też inne punkty odniesienia. Kolejno każda alternatywa jest oceniana w kontekście każdego zbioru referencyjnego, a jej ocena przyjmuje formę jej zgodności z utworzonymi zbiorem referencyjnym. Otrzymane wartości ocen są następnie łączone w celu otrzymania ogólnej oceny. Na jej podstawie jest tworzony ranking, z alternatywami uzyskującymi najwyższe wyniki na szczycie.

Ta metoda jest szczególnie pomocna w optymalizacji wielokryterialnej w sytuacjach, gdy ocena alternatyw względem skrajnych przypadków jest bardziej intuicyjna i dostępna dla decydenta.

## 4. Aplikacja

Aplikacja została napisana w języku Python wraz z bibliotekami, ponieważ pozwala on na proste tworzenie interfejsów graficznych i łatwy dostęp do bazy danych filmów IMDB. W celu jej uruchomienia należy posiadać zainstalowane następujące wersje oprogramowania i bibliotek języka Python:

- Python w wersji 3.10,
- IMDbPY w wersji 2022.7.9,
- BeautifulSoup4 w wersji 4.12.2,
- matplotlib w wersji 3.6.2,
- pandas w wersji 2.1.2,
- Kivy w wersji 2.1.0,
- Kivy-Garden w wersji 0.1.5.

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

```
def get_top_movies(self, genre: str, n_movies: int) -> List[Movie.Movie]:

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

    # Pobranie danych
    print(f"[DOWNLOAD] From {url}")
    content = self.imdb._retrieve(url)

    # Parsowanie danych
    soup = BeautifulSoup(content, features='html.parser')
    response = json.loads(soup.find(name='script', attrs={"id": "__NEXT_DATA__"}).text)
    items = response['props']['pageProps']['searchResults']['titleResults']['titleListItems']
    movies: List[Movie.Movie] = [self.imdb.get_movie(item['titleId'][2:], info=["main"]) for item in items]

    return movies
```

Rys. 1. Skrypt do pobrania i parsowania danych

Zbudowany został interfejs graficzny pozwalający na intuicyjne ustawienie i modyfikowanie preferencji w celu znalezienia odpowiedniego filmu. Umożliwia on wybór gatunków, z jakich pobierane będą filmy, wpisanie preferowanego autora, roku produkcji i obsady, a także ustawienie preferowanych wag dla poszczególnych kryteriów. Po pobraniu listy filmów możliwy jest wybór algorytmu, za pomocą którego będzie rozwiązany problem. Po zakończeniu działania algorytmu wyświetlane są 3 – 5 (w zależności od wyników zwróconych przez algorytm) najlepsze filmy zwrócone przez dany algorytm. Możliwe jest także wyświetlenie wykresów przedstawiających wykresy punktowe wartości dla każdego kryterium. Na czerwono zaznaczone są wartości dla najlepszego punktu zwróconego przez wybrany algorytm. Projekt interfejsu graficznego został przedstawiony na Rys. 2. oraz Rys. 3.

Zaznacz gatunki

Action

Adventure

Animation

Biography

Comedy

Crime

Documentary

Drama

Family

Fantasy

Film-Noir

History

Horror

Music

Musical

Mystery

Romance

Sci-Fi

Short

Sport

Thriller

War

Western

Wybierz liczbę najwyżzej ocenianych filmów

Autor

Rok produkcji

Obsada

Pobierz listę filmów

Kryterium

Waga

Autor

0.1

Obsada

0.1

Popularność obsady

0.2

Ocena

0.3

Rok produkcji

0.2

Efekty specjalne

0.1

TOPSIS

UTA

RSM

Wyświetl

Rys. 2. Interfejs graficzny – wprowadzenie danych



## Pierwszy zestaw wag – konfiguracja bazowa:

Kryterium	Waga
Autor	0.1
Obsada	0.1
Popularność obsady	0.2
Ocena	0.3
Rok produkcji	0.2
Efekty specjalne	0.1

Rys. 4. Zestaw wag w eksperymencie

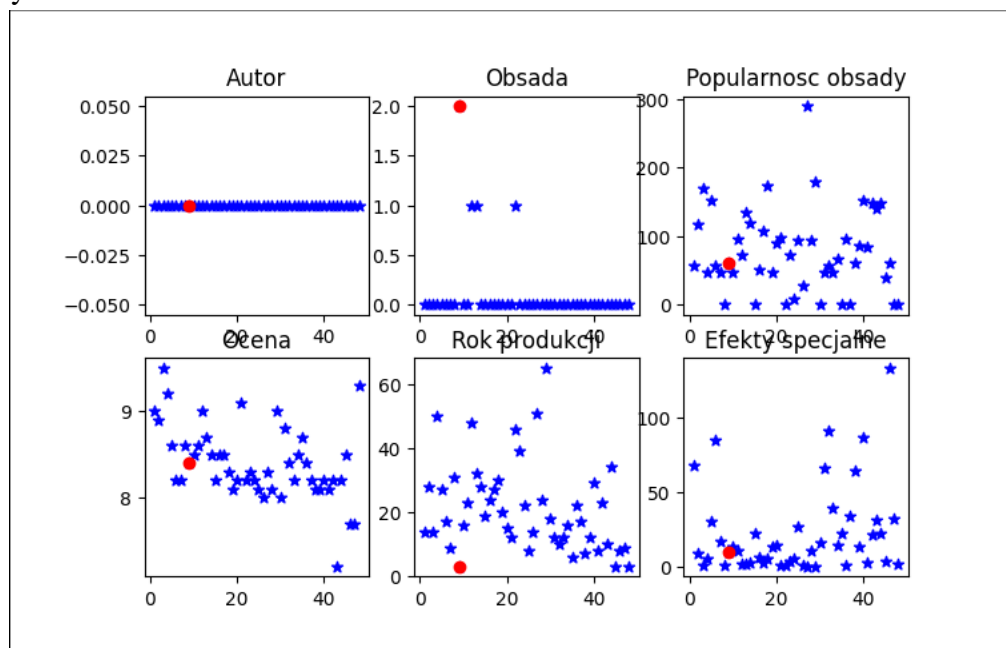
Algorytm TOPSIS:

Czas wykonania: 2ms

Najlepsze filmy:

1. Joker
2. The Dark Knight
3. Breaking Bad
4. Pulp Fiction

Wykresy:



Rys. 5. Wykresy dla algorytmu TOPSIS

Algorytm UTA:

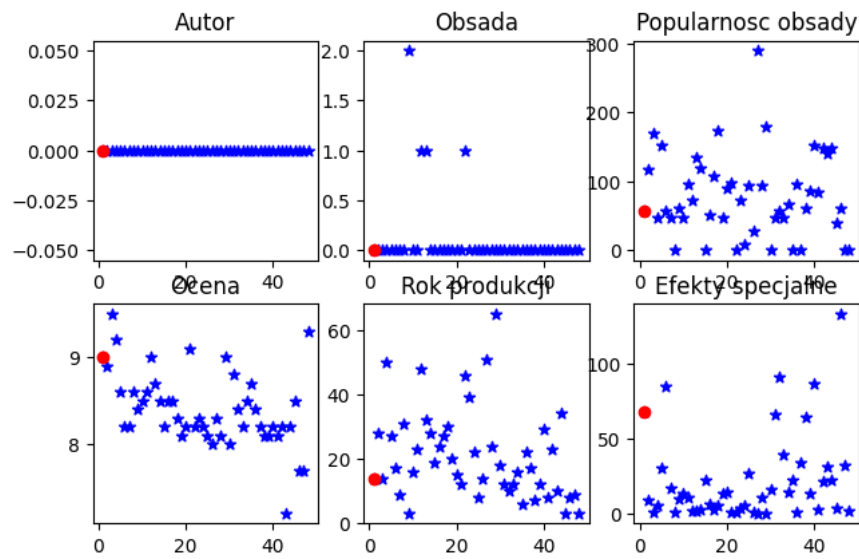
Czas wykonania: 6ms

Najlepsze filmy:

1. The Dark Knight
2. Pulp Fiction
3. Breaking Bad



Wykresy:



Rys. 6. Wykresy dla algorytmu UTA

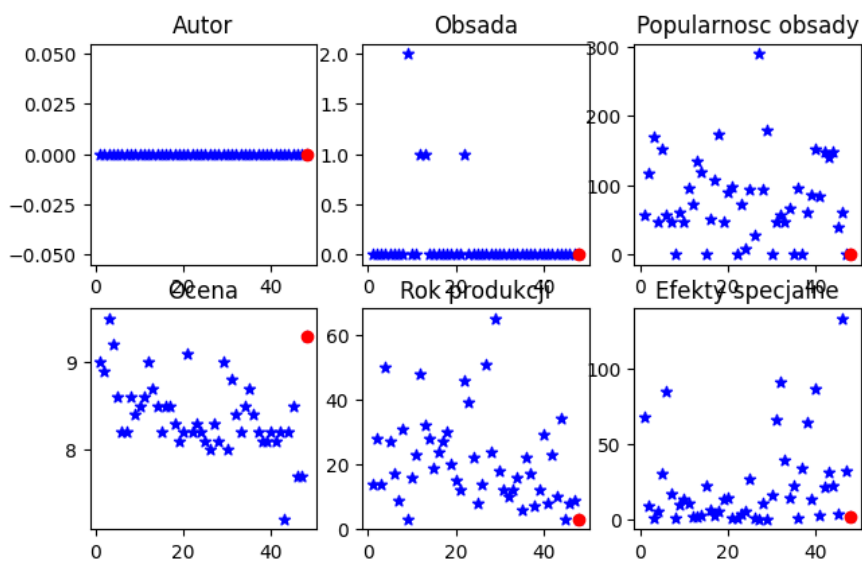
Algorytm RSM:

Czas wykonania: 15ms

Najlepsze filmy:

1. Chernobyl
2. Captain America: The Winter Soldier
3. Parasite

Wykresy:



Rys. 7. Wykresy dla algorytmu RSM

## Drugi zestaw wag – duża waga dla autora:

Kryterium	Waga
Autor	0.9
Obsada	0.02
Popularność obsady	0.02
Ocena	0.02
Rok produkcji	0.02
Efekty specjalne	0.02

Rys. 8. Zestaw wag w eksperymencie

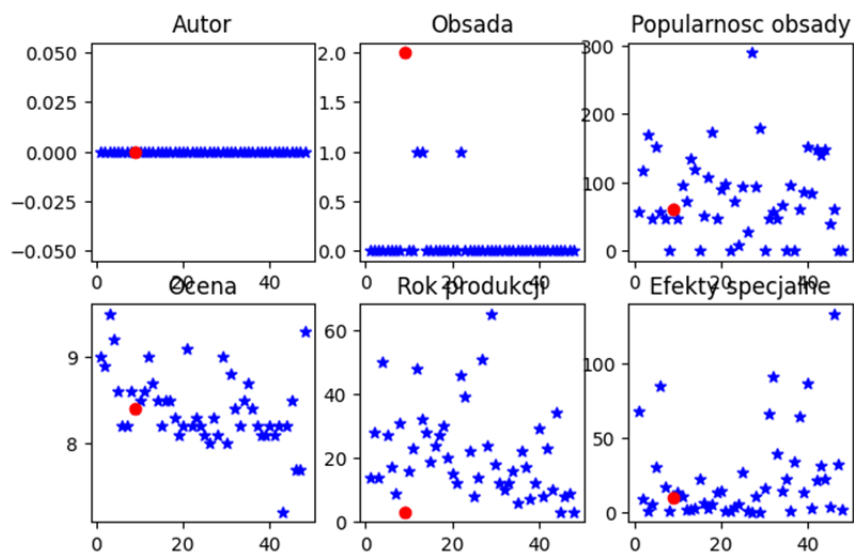
Algorytm TOPSIS:

Czas wykonania: 1ms

Najlepsze filmy:

1. Joker
2. The Dark Knight
3. Breaking Bad
4. Pulp Fiction

Wykresy:



Rys. 9. Wykresy dla algorytmu TOPSIS

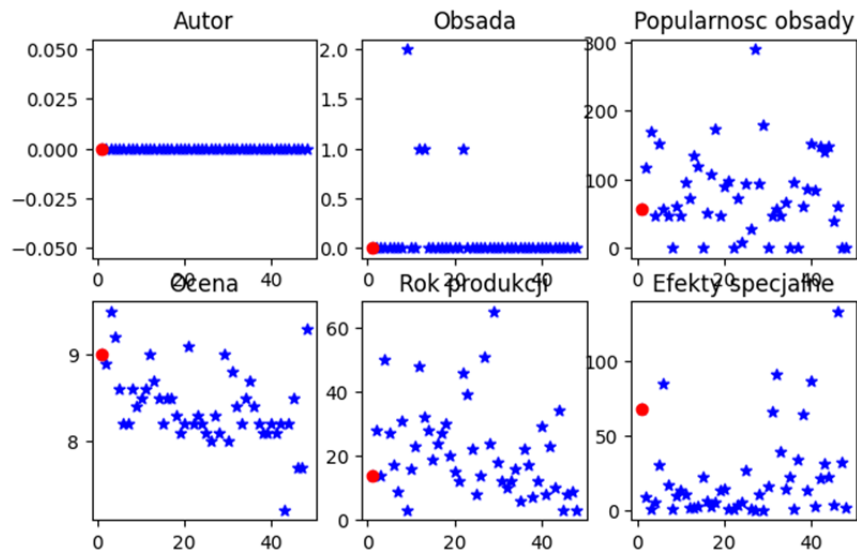
Algorytm UTA:

Czas wykonania: 7ms

Najlepsze filmy:

1. The Dark Knight
2. Pulp Fiction
3. Breaking Bad

Wykresy:



Rys. 10. Wykresy dla algorytmu UTA

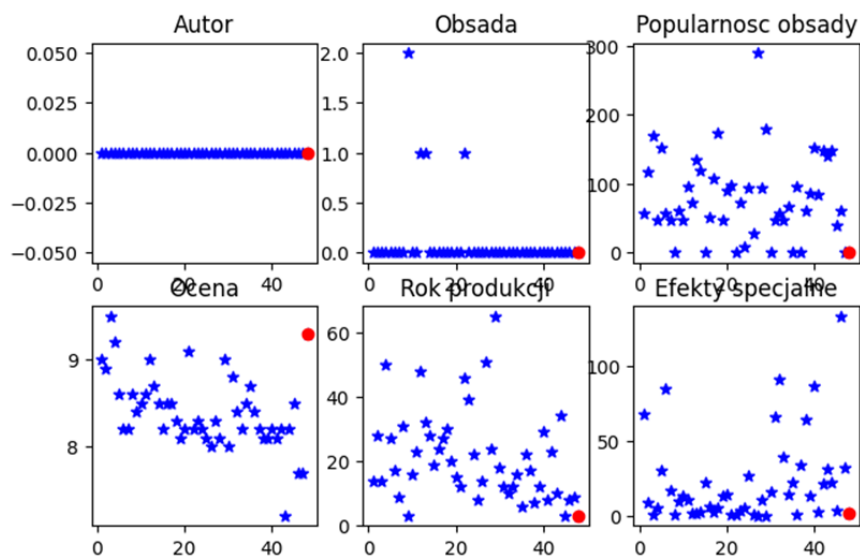
Algorytm RSM:

Czas wykonania: 13ms

Najlepsze filmy:

1. Chernobyl
2. Captain America: The Winter Soldier
3. Parasite

Wykresy:



Rys. 7. Wykresy dla algorytmu RSM

### Trzeci zestaw wag – duża waga dla obsady:

Kryterium	Waga
Autor	0.02
Obsada	0.9
Popularność obsady	0.02
Ocena	0.02
Rok produkcji	0.02
Efekty specjalne	0.02

Rys. 8. Zestaw wag w eksperymencie

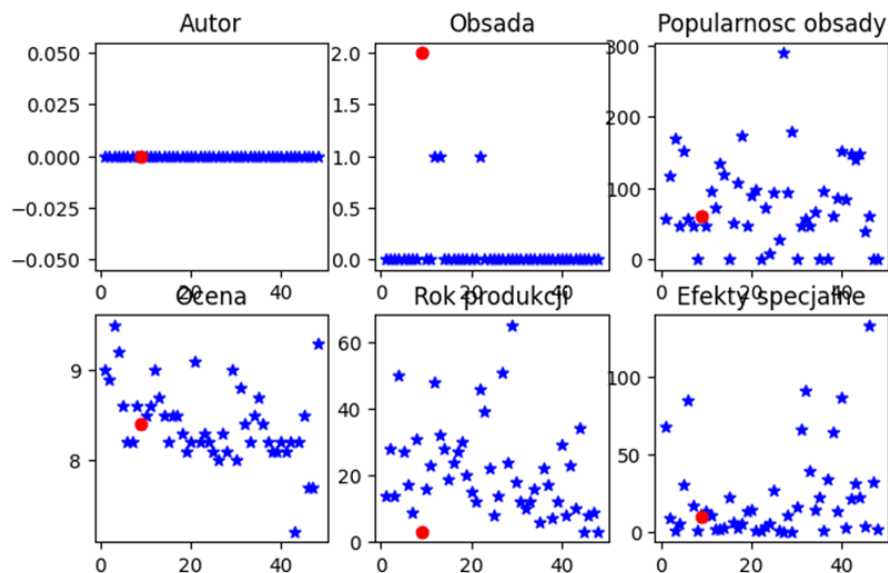
Algorytm TOPSIS:

Czas wykonania: 0ms

Najlepsze filmy:

1. Joker
2. The Dark Knight
3. Breaking Bad
4. Pulp Fiction

Wykresy:



Rys. 9. Wykresy dla algorytmu TOPSIS

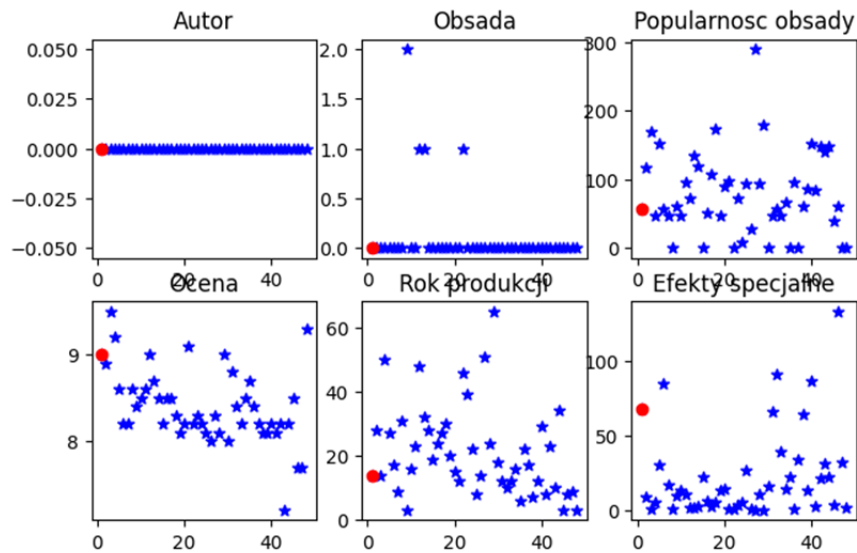
Algorytm UTA:

Czas wykonania: 4ms

Najlepsze filmy:

1. The Dark Knight
2. Pulp Fiction
3. Breaking Bad

Wykresy:



Rys. 10. Wykresy dla algorytmu UTA

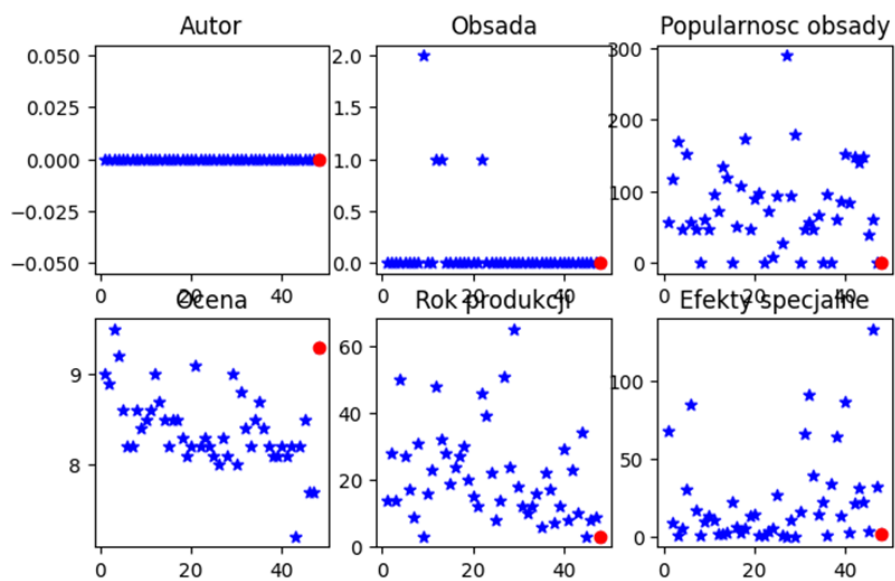
Algorytm RSM:

Czas wykonania: 14ms

Najlepsze filmy:

1. Chernobyl
2. Captain America: The Winter Soldier
3. Parasite

Wykresy:



Rys. 11. Wykresy dla algorytmu RSM

#### Czwarty zestaw wag – duża waga popularności obsady:

Kryterium	Waga
Autor	0.02
Obsada	0.02
Popularność obsady	0.9
Ocena	0.02
Rok produkcji	0.02
Efekty specjalne	0.02

Rys. 12. Zestaw wag w eksperymencie

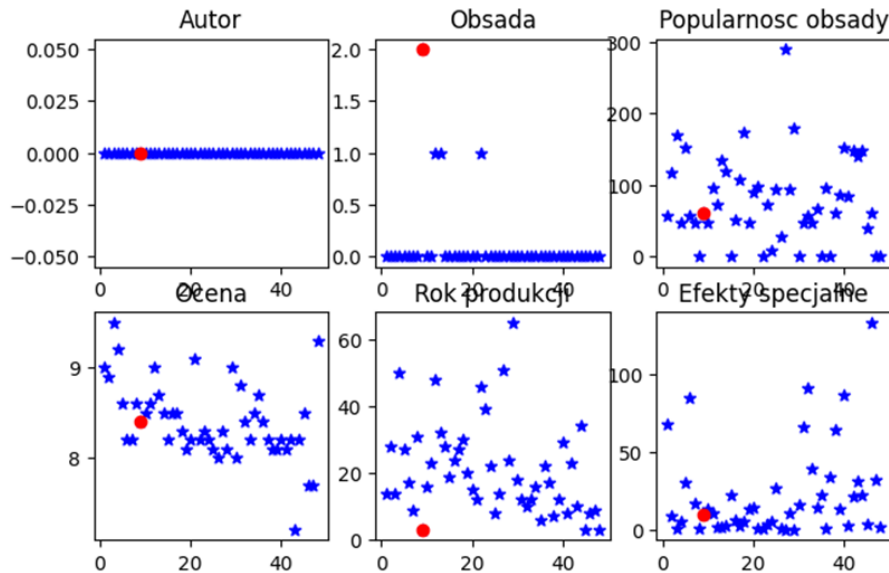
Algorytm TOPSIS:

Czas wykonania: 2ms

Najlepsze filmy:

1. Joker
2. Breaking Bad
3. The Dark Knight
4. Pulp Fiction

Wykresy:



Rys. 13. Wykresy dla algorytmu TOPSIS

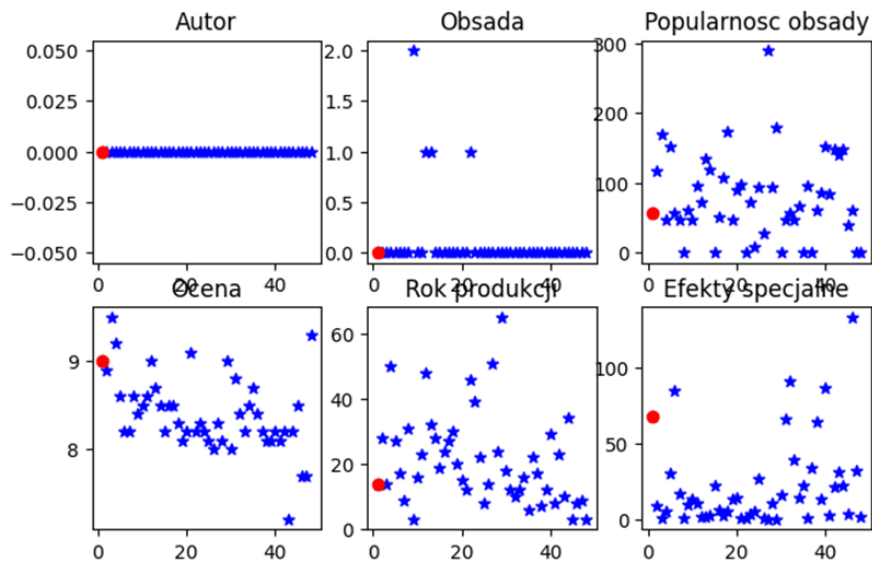
Algorytm UTA:

Czas wykonania: 6ms

Najlepsze filmy:

1. The Dark Knight
2. Pulp Fiction
3. Breaking Bad

Wykresy:



Rys. 14. Wykresy dla algorytmu UTA

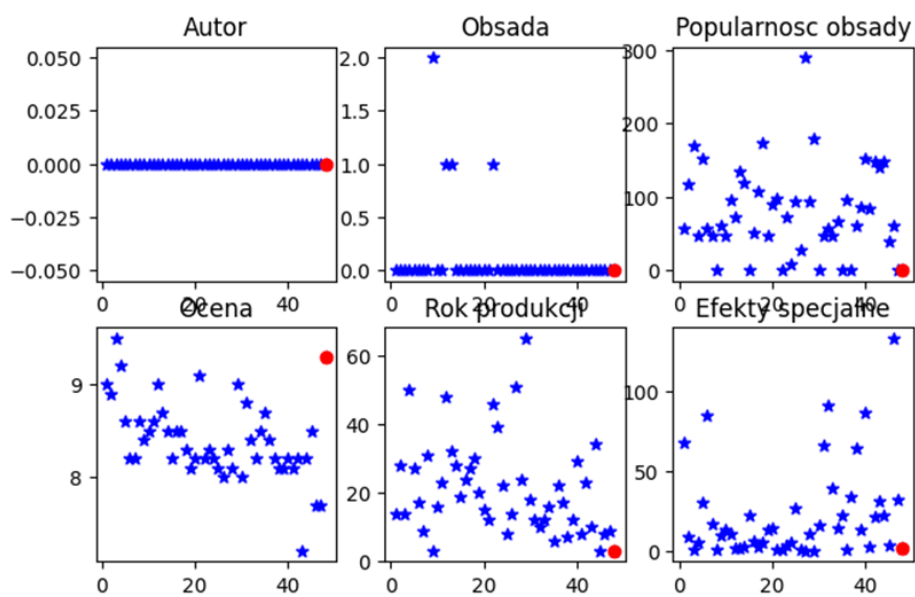
Algorytm RSM:

Czas wykonania: 14ms

Najlepsze filmy:

1. Chernobyl
2. Captain America: The Winter Soldier
3. Parasite

Wykresy:



Rys. 15. Wykresy dla algorytmu RSM

### Piąty zestaw wag – duża waga oceny:

Kryterium	Waga
Autor	0.02
Obsada	0.02
Popularność obsady	0.02
Ocena	0.9
Rok produkcji	0.02
Efekty specjalne	0.02

Rys. 16. Zestaw wag w eksperymencie

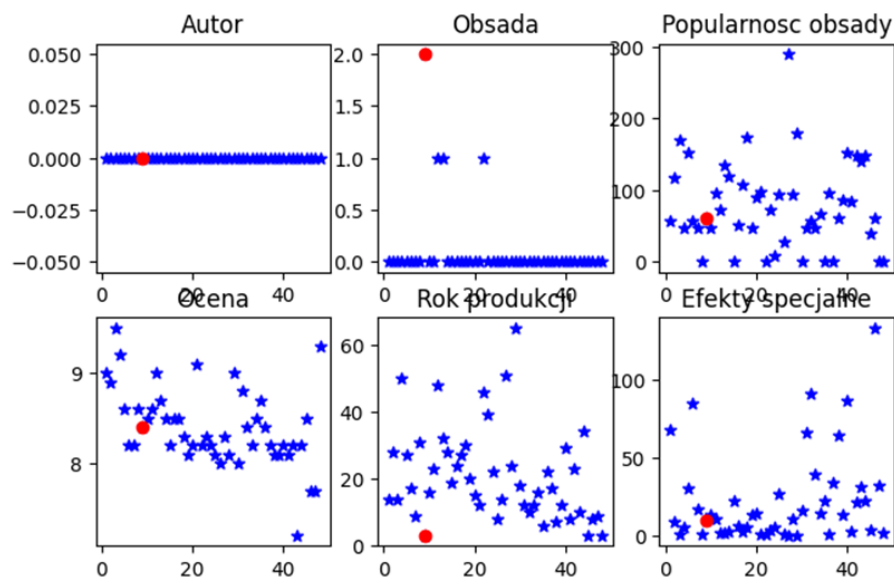
Algorytm TOPSIS:

Czas wykonania: 1ms

Najlepsze filmy:

1. Joker
2. The Dark Knight
3. Breaking Bad
4. Pulp Fiction

Wykresy:



Rys. 17. Wykresy dla algorytmu TOPSIS

Algorytm UTA:

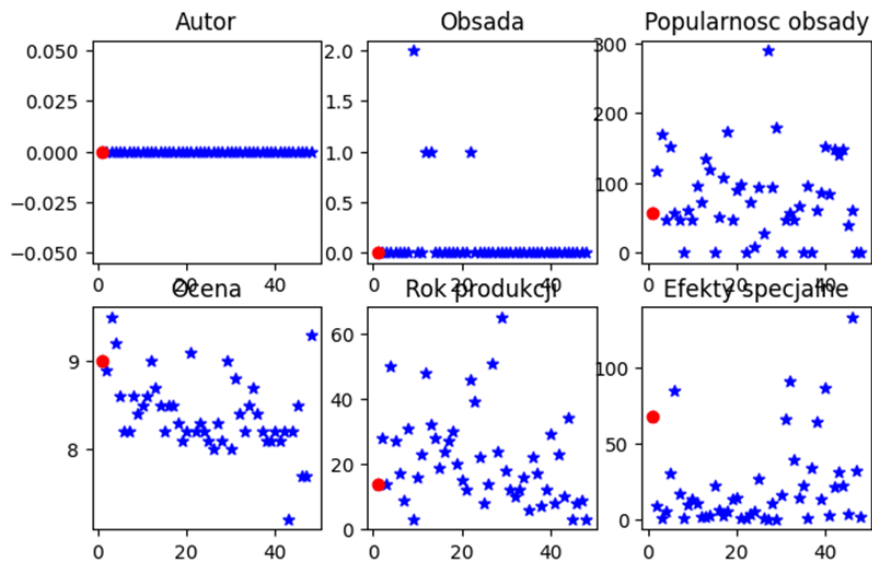
Czas wykonania: 6ms

Najlepsze filmy:

1. The Dark Knight
2. Pulp Fiction
3. Breaking Bad



Wykresy:



Rys. 18. Wykresy dla algorytmu UTA

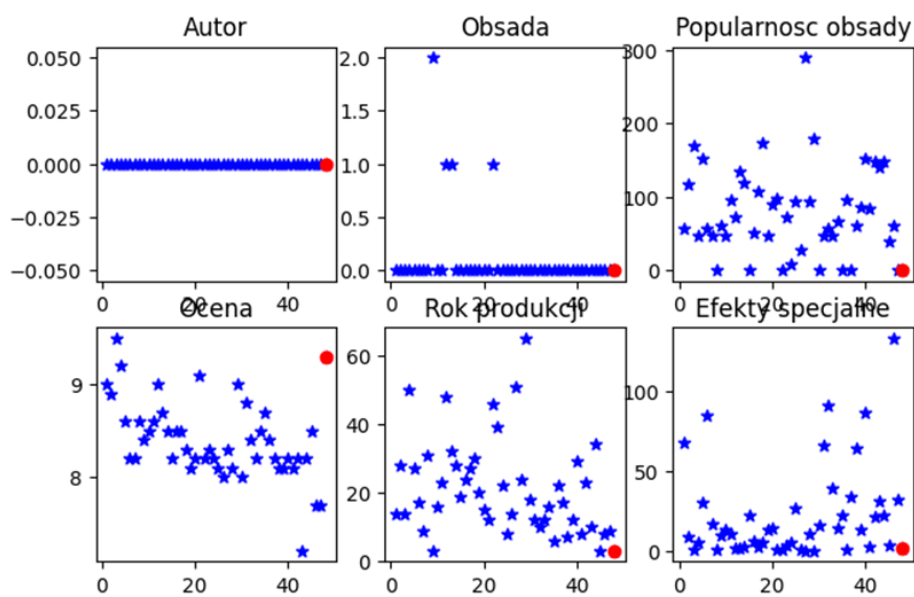
Algorytm RSM:

Czas wykonania: 14ms

Najlepsze filmy:

1. Chernobyl
2. Captain America: The Winter Soldier
3. Parasite

Wykresy:



Rys. 19. Wykresy dla algorytmu RSM

### Szósty zestaw wag – duża waga roku produkcji:

Kryterium	Waga
Autor	0.02
Obsada	0.02
Popularność obsady	0.02
Ocena	0.02
Rok produkcji	0.9
Efekty specjalne	0.02

Rys. 20. Zestaw wag w eksperymencie

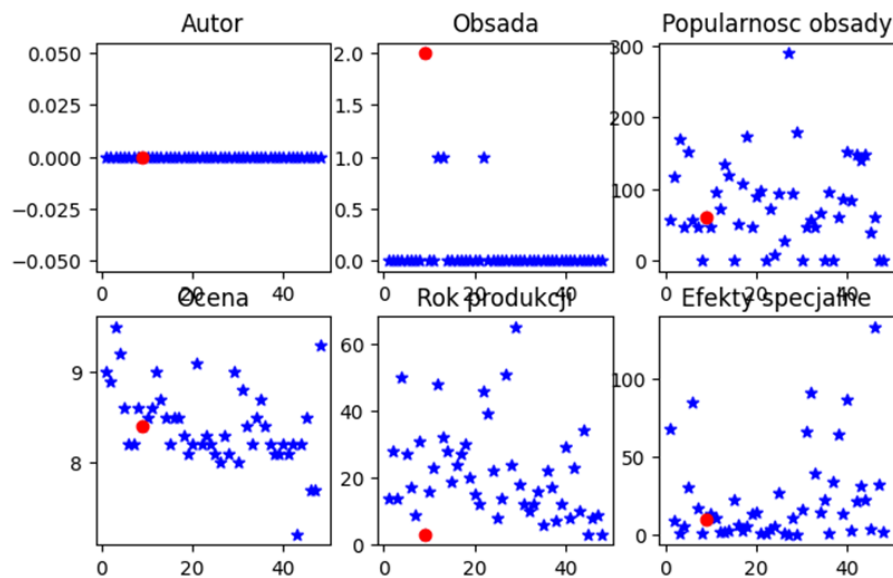
### Algorytm TOPSIS:

Czas wykonania: 0ms

Najlepsze filmy:

1. Joker
2. The Dark Knight
3. Breaking Bad
4. Pulp Fiction

Wykresy:



Rys. 21. Wykresy dla algorytmu TOPSIS

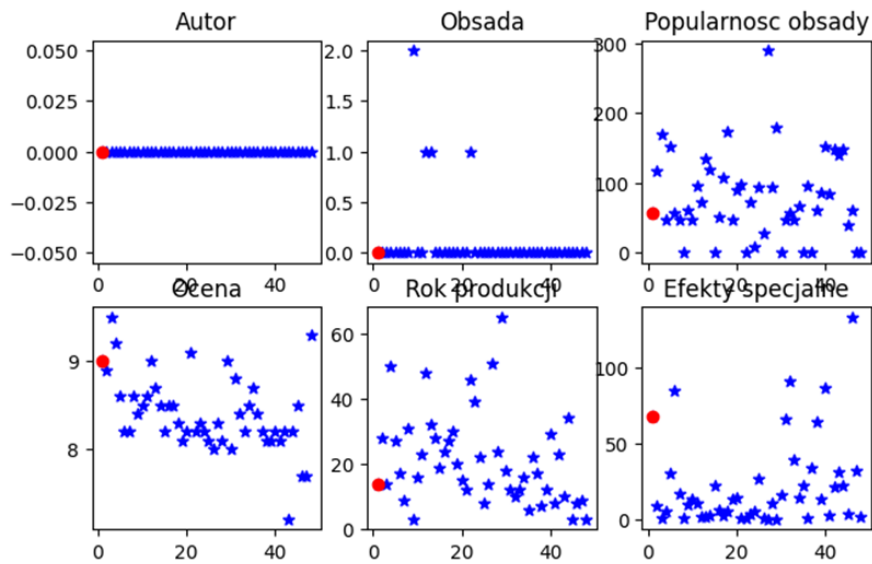
### Algorytm UTA:

Czas wykonania: 5ms

Najlepsze filmy:

1. The Dark Knight
2. Pulp Fiction
3. Breaking Bad

Wykresy:



Rys. 22. Wykresy dla algorytmu UTA

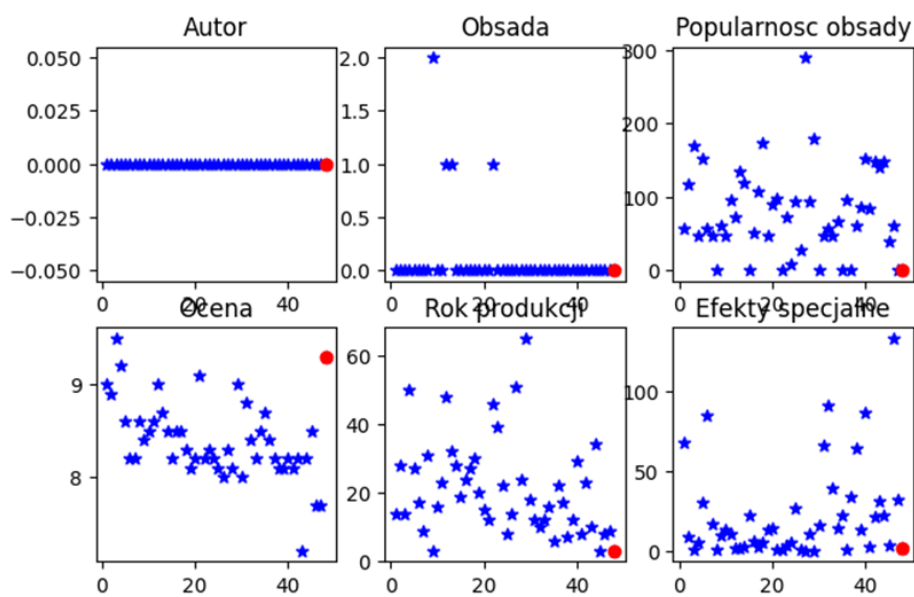
Algorytm RSM:

Czas wykonania: 14ms

Najlepsze filmy:

1. Chernobyl
2. Captain America: The Winter Soldier
3. Parasite

Wykresy:



Rys. 23. Wykresy dla algorytmu RSM

### Siódmy zestaw wag – duża waga efektów specjalnych:

Kryterium	Waga
Autor	0.02
Obsada	0.02
Popularność obsady	0.02
Ocena	0.02
Rok produkcji	0.02
Efekty specjalne	0.9

Rys. 24. Zestaw wag w eksperymencie

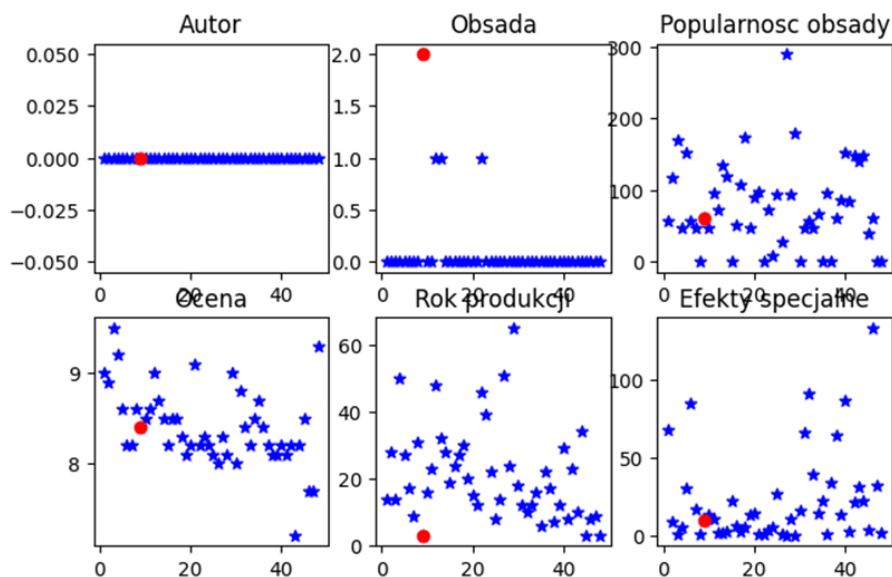
Algorytm TOPSIS:

Czas wykonania: 2ms

Najlepsze filmy:

1. Joker
2. The Dark Knight
3. Pulp Fiction
4. Breaking Bad

Wykresy:



Rys. 25. Wykresy dla algorytmu TOPSIS

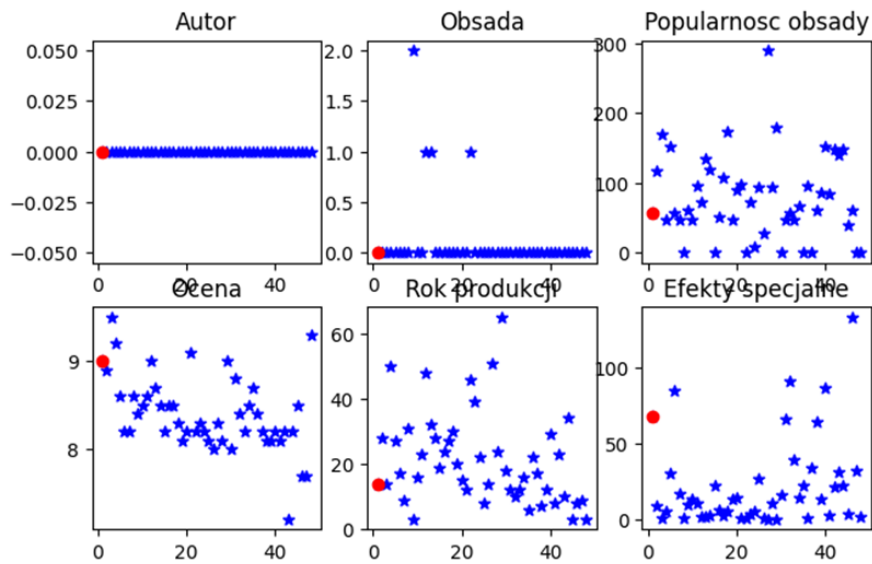
Algorytm UTA:

Czas wykonania: 7ms

Najlepsze filmy:

1. The Dark Knight
2. Pulp Fiction
3. Breaking Bad

Wykresy:



Rys. 26. Wykresy dla algorytmu UTA

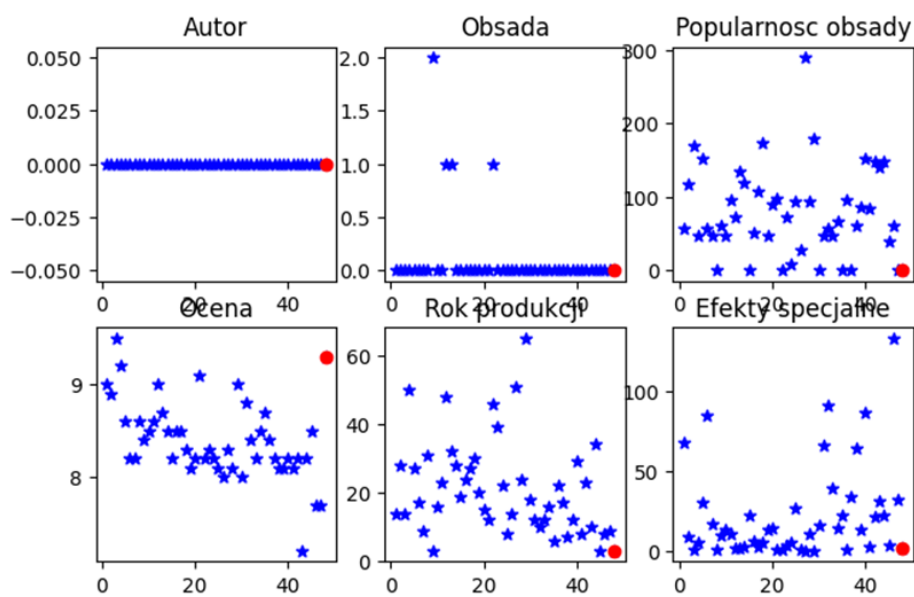
Algorytm RSM:

Czas wykonania: 11ms

Najlepsze filmy:

1. Chernobyl
2. Captain America: The Winter Soldier
3. Parasite

Wykresy:



Rys. 27. Wykresy dla algorytmu RSM

### Podsumowanie serii 1:

Patrząc na powyższe wyniki pierwszą z rzeczy, która rzuca się w oczy jest fakt, że ranking filmów zwrócony dla metod Topsis oraz UTA jest podobny, natomiast ranking dla metody RSM odbiega od tych dwóch pozostałych zawierając inne filmy. Może być to spowodowane odwołaniem do zbiorów referencyjnych, podczas gdy działanie dwóch pierwszych metod jest w miarę zbliżone. Widać również, że rankingi dla poszczególnych zestawów wag kryteriów są bardzo podobne, tak więc wszystkie w wpływają w sposób porównywalny na końcowy ranking dla wszystkich metod. Prawdopodobną przyczyną takiego stanu rzeczy jest w pewnym sensie ograniczony zbiór danych, ponieważ pobierane jest po 30 filmów dla gatunku, tak więc odpowiedni dobór najlepszych filmów nie jest trudny. Uwzględnienie większej liczby filmów jest jednak problematyczne (pomimo, a może właśnie z powodu) bardzo dużej bazy danych. Występuje tutaj dość duża złożoność czasowa podczas wczytywania filmów. Z przedstawionych wykresów widać, że każda metoda zwraca inne punkty, jednak po dokładniejszej analizie można doszukać się pewnych cech wspólnych w dwóch pierwszych metodach, co znajduje potwierdzenie właśnie w podobieństwie końcowych rankingów. Ostatecznym stwierdzeniem, którym można podsumować pierwszą serię eksperymentów jest fakt, że po ręcznym sprawdzeniu, zwrócone listy faktycznie zawierają filmy wysoko oceniane i spełniające większość kryteriów.

## 5.2. Eksperymenty 2

Drugą serię eksperymentów wykonano dla następujących danych:

- gatunki Action, Comedy oraz Romance (pobierane po 20 filmów),
- autor: Patryk Vega,
- rok produkcji: 2015,
- obsada: Piotr Adamczyk, Tomasz Karolak, Agnieszka Dygant.

Dla tej grupy eksperymentów zmieniane były wagi dla poszczególnych kryteriów. Wyniki eksperymentów zostały przedstawione poniżej.

### Pierwszy zestaw wag – konfiguracja bazowa:

Kryterium	Waga
Autor	0.1
Obsada	0.1
Popularność obsady	0.2
Ocena	0.3
Rok produkcji	0.2
Efekty specjalne	0.1

Rys. 28. Zestaw wag w eksperymencie

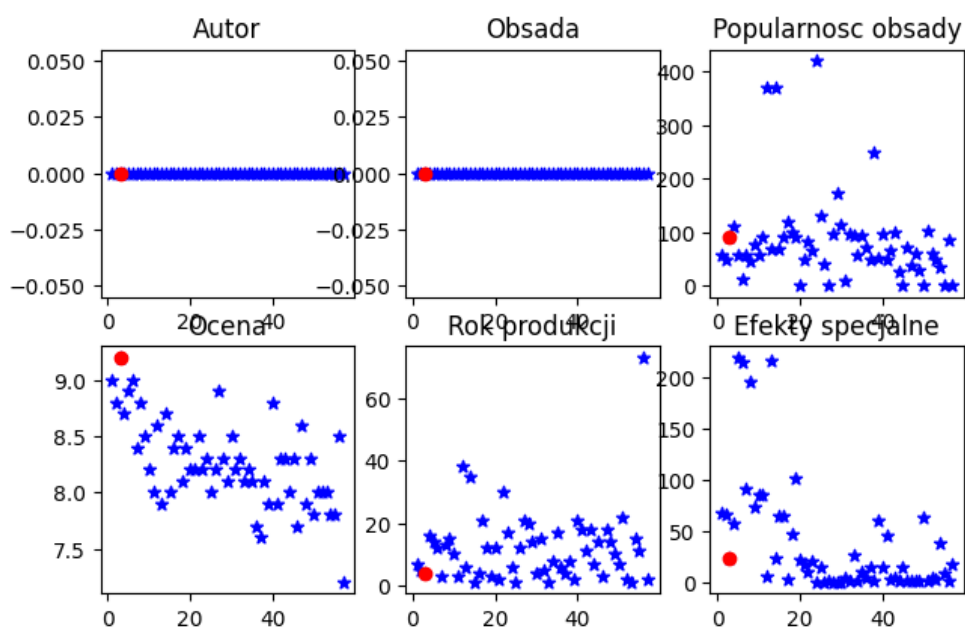
*Algorytm TOPSIS:*

Czas wykonania: 1ms

Najlepsze filmy:

1. Game of Thrones
2. Up
3. The Dark Knight
4. The Matrix
5. Star Wars: Episode V - The Empire Strikes Back

Wykresy:



Rys. 29. Wykresy dla algorytmu TOPSIS

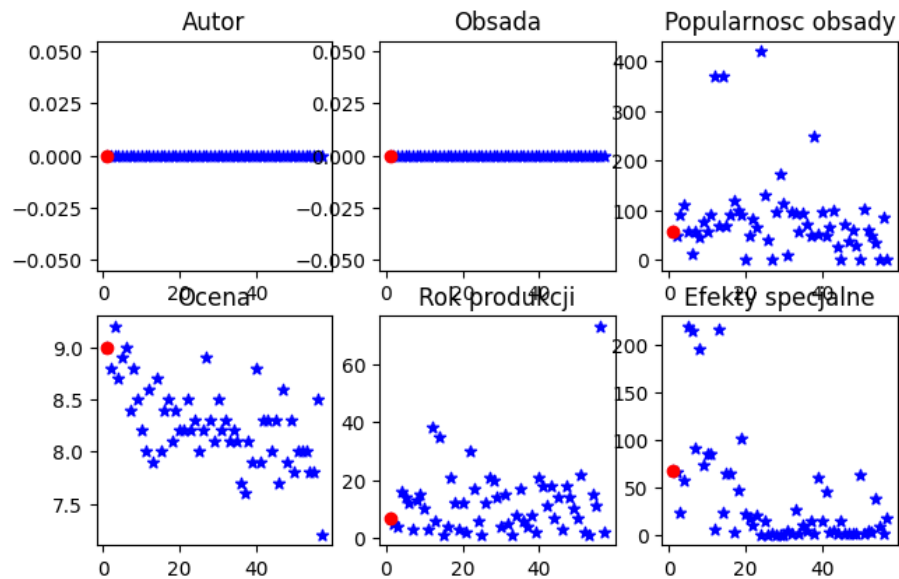
*Algorytm UTA:*

Czas wykonania: 9ms

Najlepsze filmy:

1. The Dark Knight
2. Inception
3. Game of Thrones

Wykresy:



Rys. 30. Wykresy dla algorytmu UTA

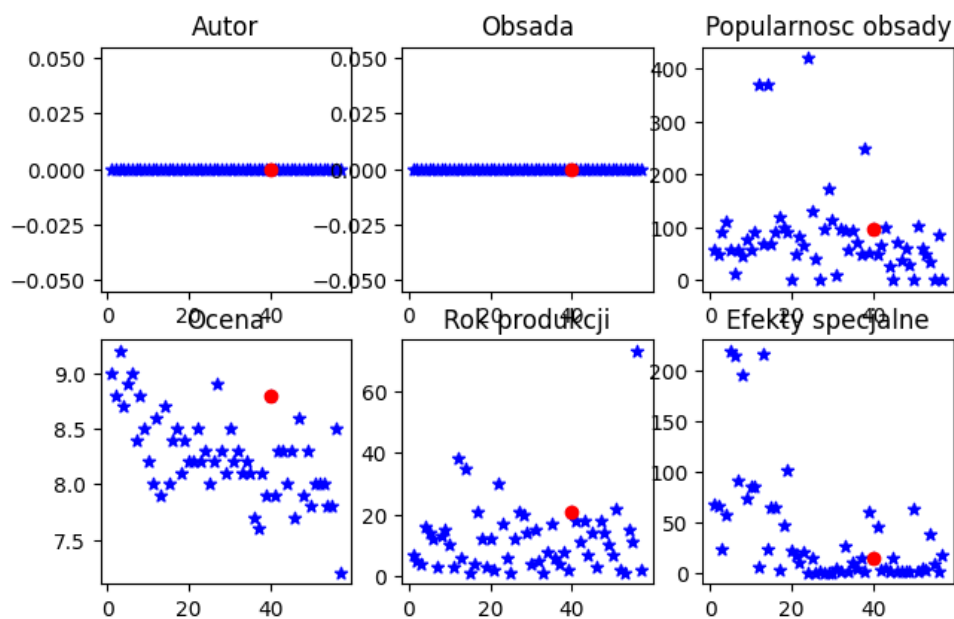
Algorytm RSM:

Czas wykonania: 15ms

Najlepsze filmy:

1. Forrest Gump
2. Ratatouille
3. The Grand Budapest Hotel

Wykresy:



Rys. 31. Wykresy dla algorytmu RSM



## Drugi zestaw wag – duża waga popularności obsady:

Kryterium	Waga
Autor	0.02
Obsada	0.02
Popularność obsady	0.9
Ocena	0.02
Rok produkcji	0.02
Efekty specjalne	0.02

Rys. 32. Zestaw wag w eksperymencie

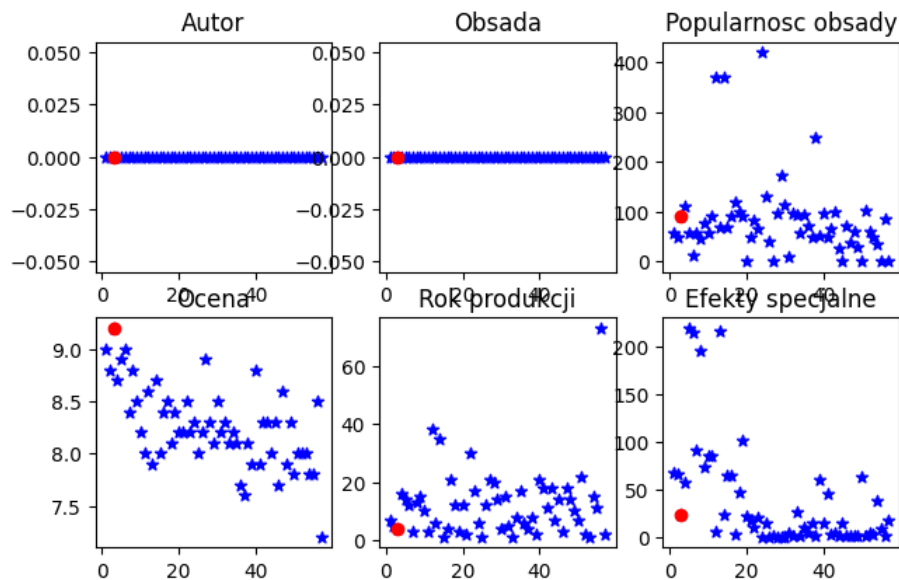
### Algorytm TOPSIS:

Czas wykonania: 1ms

Najlepsze filmy:

1. Game of Thrones
2. Up
3. The Dark Knight
4. The Matrix
5. Star Wars: Episode V - The Empire Strikes Back

Wykresy:



Rys. 33. Wykresy dla algorytmu TOPSIS

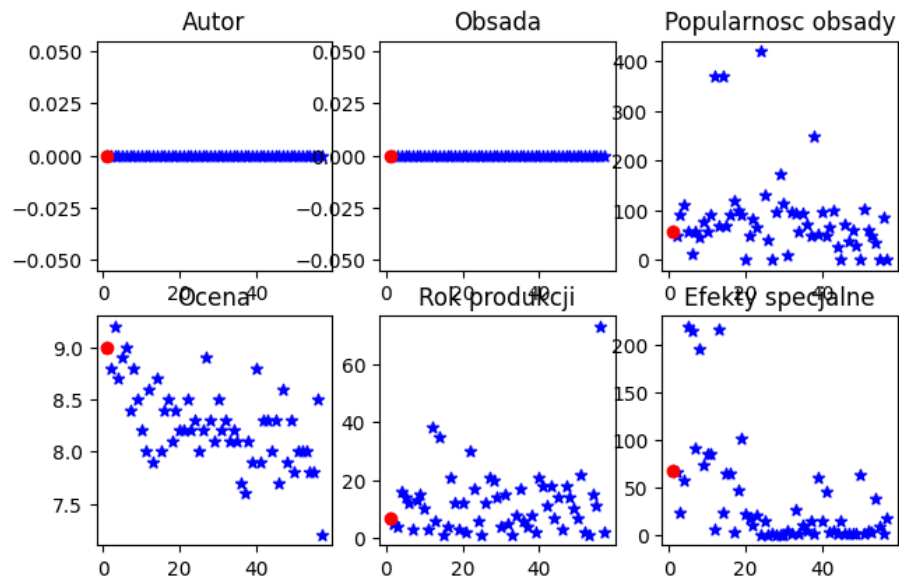
### Algorytm UTA:

Czas wykonania: 9ms

Najlepsze filmy:

1. The Dark Knight
2. Inception
3. Game of Thrones

Wykresy:



Rys. 34. Wykresy dla algorytmu UTA

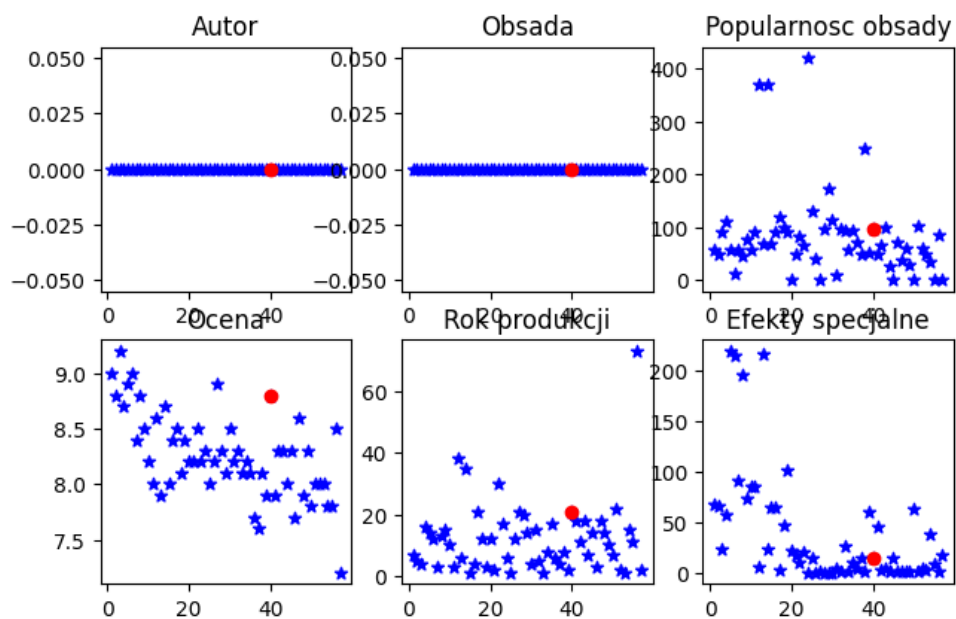
Algorytm RSM:

Czas wykonania: 17ms

Najlepsze filmy:

1. Forrest Gump
2. Ratatouille
3. The Grand Budapest Hotel

Wykresy:



Rys. 35. Wykresy dla algorytmu RSM

### Trzeci zestaw wag – duża waga oceny:

Kryterium	Waga
Autor	0.02
Obsada	0.02
Popularność obsady	0.02
Ocena	0.9
Rok produkcji	0.02
Efekty specjalne	0.02

Rys. 36. Zestaw wag w eksperymencie

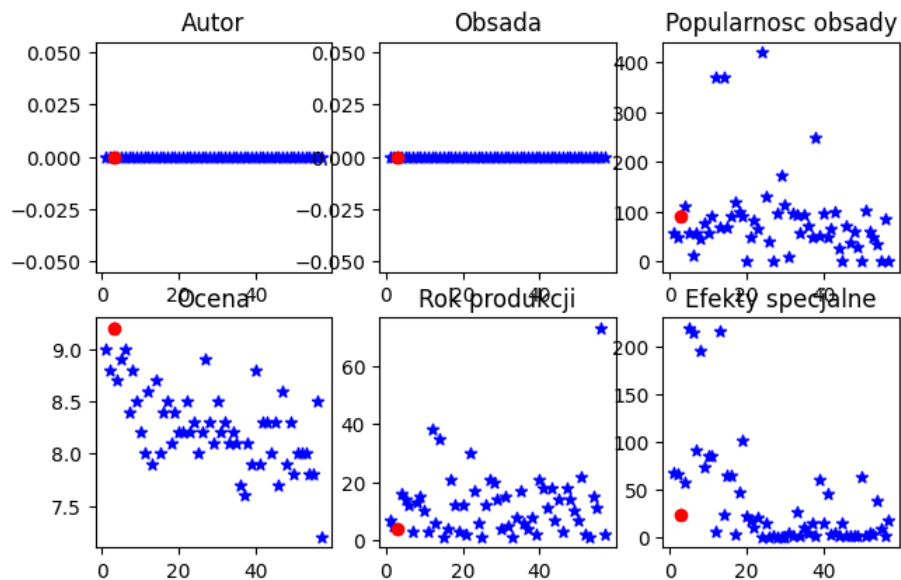
#### Algorytm TOPSIS:

Czas wykonania: 1ms

Najlepsze filmy:

1. Game of Thrones
2. Up
3. The Dark Knight
4. The Matrix
5. Star Wars: Episode V - The Empire Strikes Back

Wykresy:



Rys. 37. Wykresy dla algorytmu TOPSIS

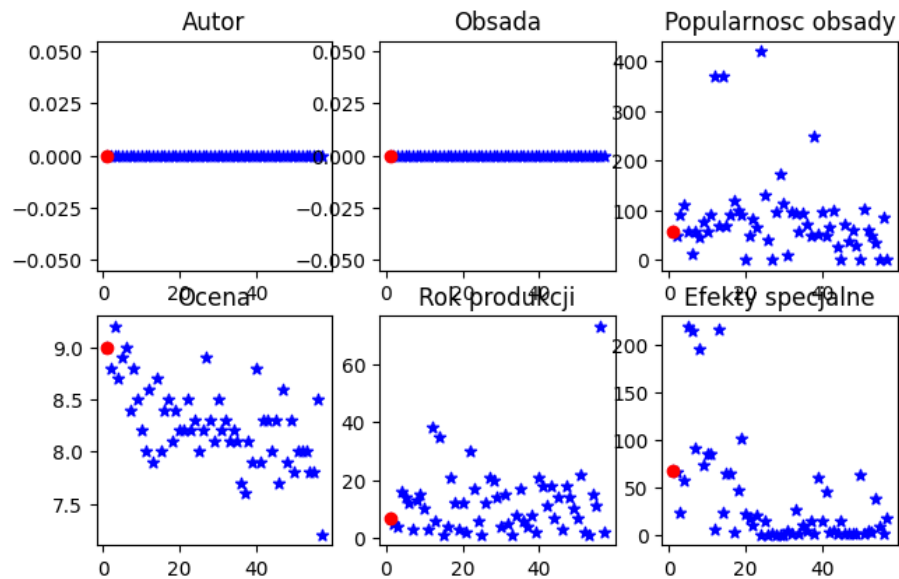
#### Algorytm UTA:

Czas wykonania: 8ms

Najlepsze filmy:

1. The Dark Knight
2. Inception
3. Game of Thrones

Wykresy:



Rys. 38. Wykresy dla algorytmu UTA

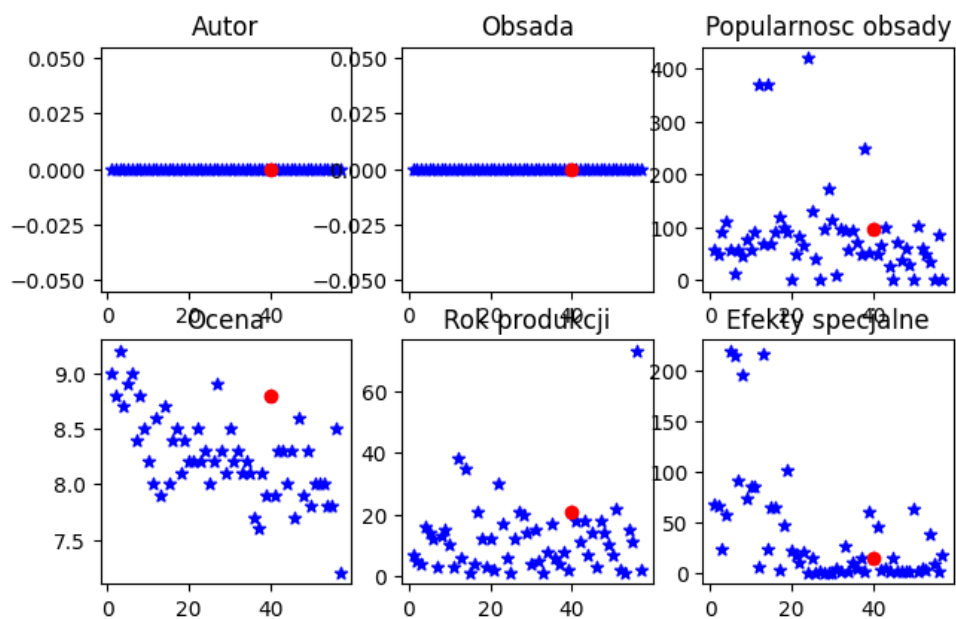
Algorytm RSM:

Czas wykonania: 12ms

Najlepsze filmy:

1. Forrest Gump
2. Ratatouille
3. The Grand Budapest Hotel

Wykresy:



Rys. 39. Wykresy dla algorytmu RSM

### Czwarty zestaw wag – duża waga roku produkcji:

Kryterium	Waga
Autor	0.02
Obsada	0.02
Popularność obsady	0.02
Ocena	0.02
Rok produkcji	0.9
Efekty specjalne	0.02

Rys. 40. Zestaw wag w eksperymencie

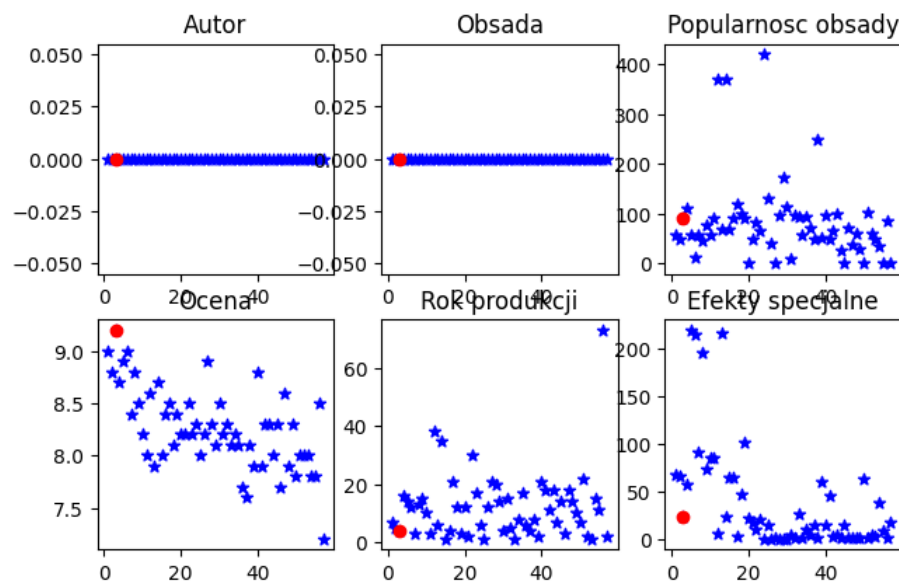
### Algorytm TOPSIS:

Czas wykonania: 1ms

Najlepsze filmy:

1. Game of Thrones
2. Up
3. The Dark Knight
4. The Matrix
5. Star Wars: Episode V - The Empire Strikes Back

Wykresy:



Rys. 41. Wykresy dla algorytmu TOPSIS

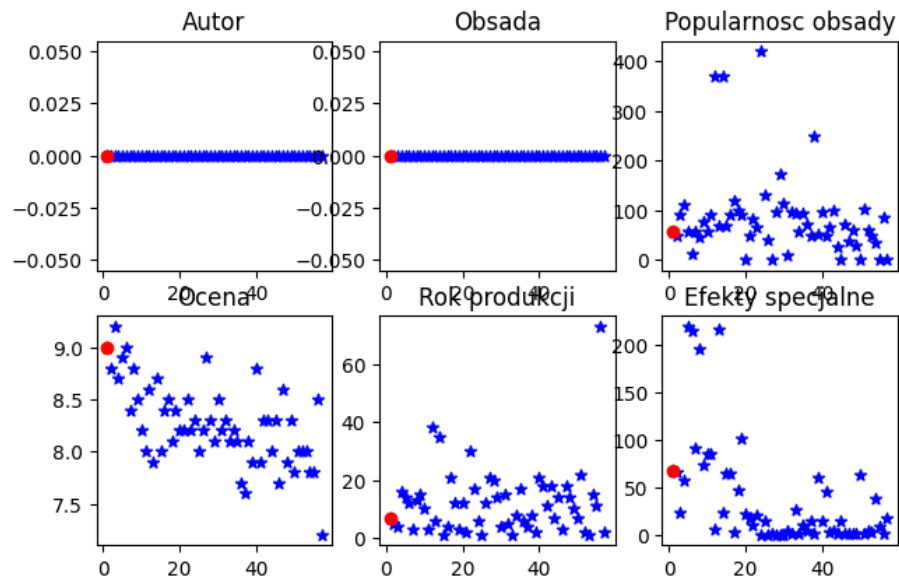
### Algorytm UTA:

Czas wykonania: 11ms

Najlepsze filmy:

1. The Dark Knight
2. Inception
3. Game of Thrones

Wykresy:



Rys. 42. Wykresy dla algorytmu UTA

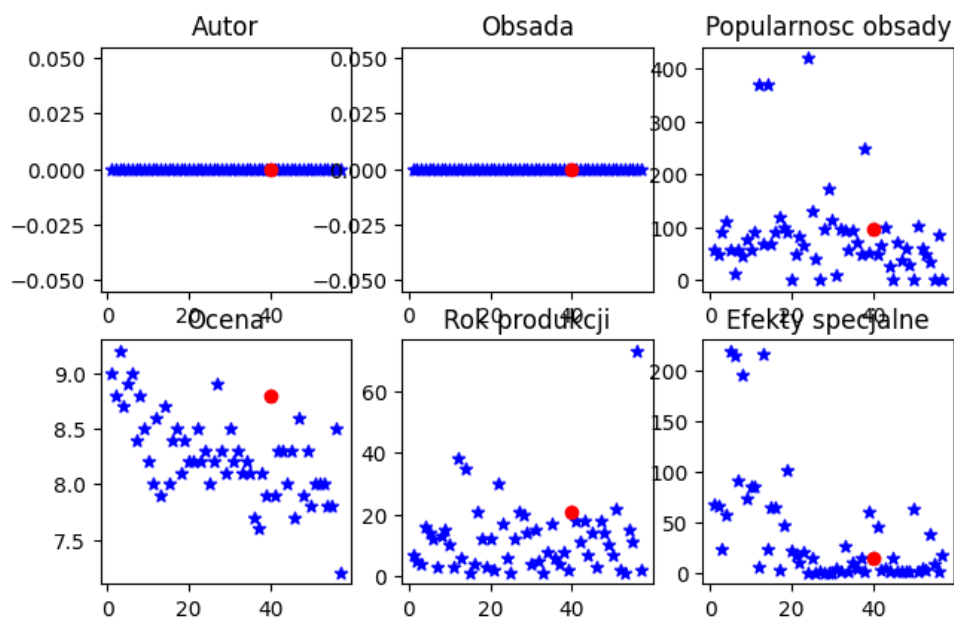
Algorytm RSM:

Czas wykonania: 15ms

Najlepsze filmy:

1. Forrest Gump
2. Ratatouille
3. The Grand Budapest Hotel

Wykresy:



Rys. 43. Wykresy dla algorytmu RSM

### Piąty zestaw wag – duża waga efektów specjalnych:

Kryterium	Waga
Autor	0.02
Obsada	0.02
Popularność obsady	0.02
Ocena	0.02
Rok produkcji	0.02
Efekty specjalne	0.9

Rys. 44. Zestaw wag w eksperymencie

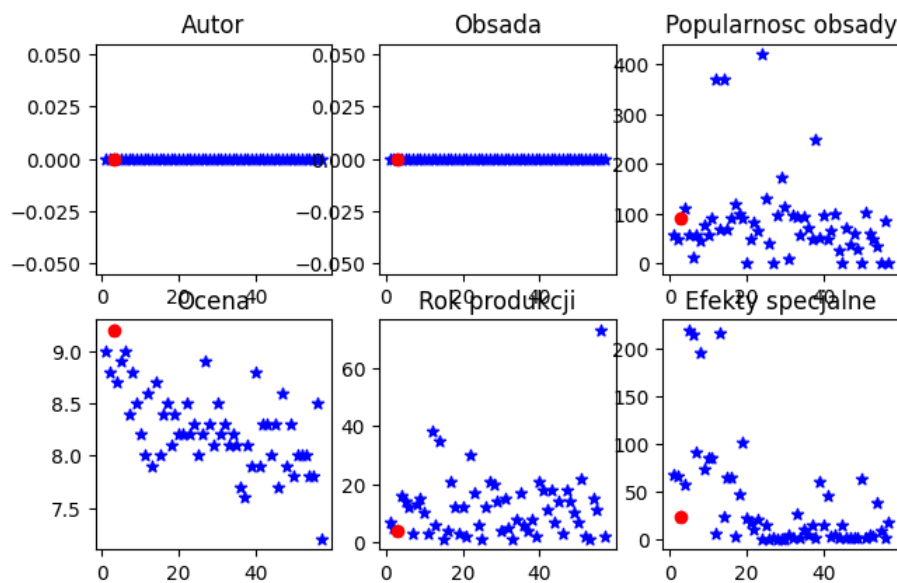
### Algorytm TOPSIS:

Czas wykonania: 2ms

Najlepsze filmy:

1. Game of Thrones
2. The Dark Knight
3. The Matrix
4. Star Wars: Episode V - The Empire Strikes Back
5. Up

Wykresy:



Rys. 45. Wykresy dla algorytmu TOPSIS

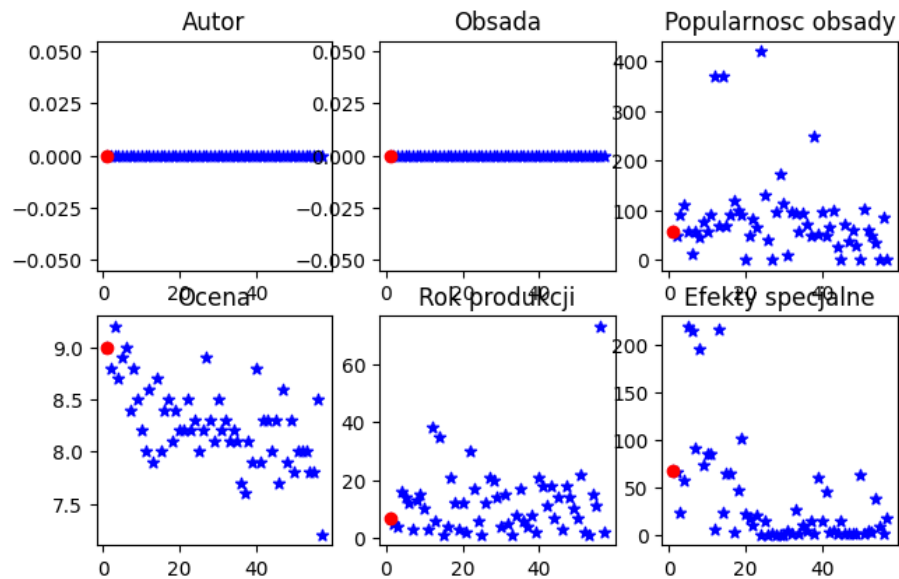
### Algorytm UTA:

Czas wykonania: 7ms

Najlepsze filmy:

1. The Dark Knight
2. Inception
3. Game of Thrones

Wykresy:



Rys. 46. Wykresy dla algorytmu UTA

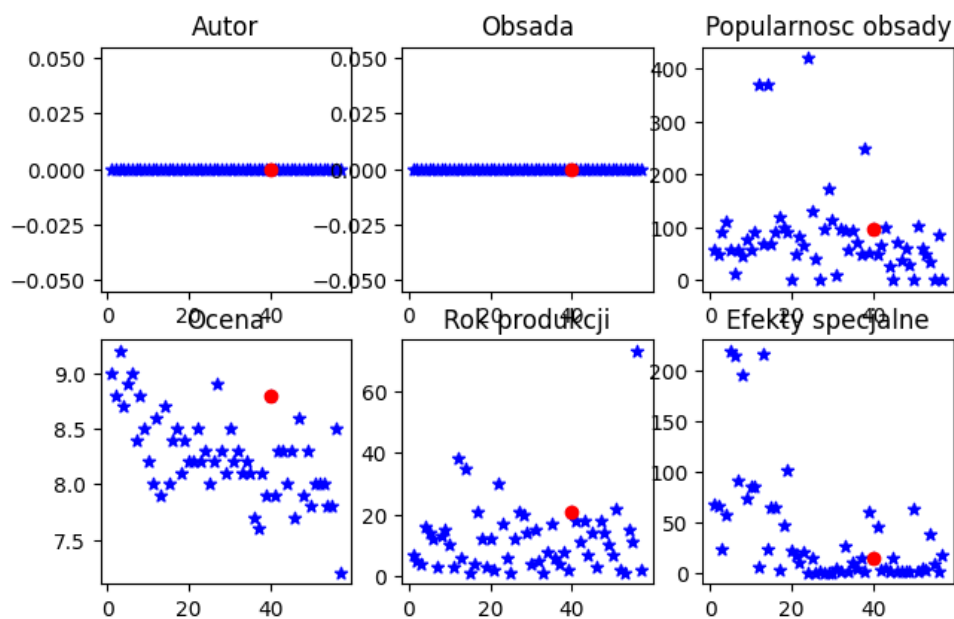
Algorytm RSM:

Czas wykonania: 12ms

Najlepsze filmy:

1. Forrest Gump
2. Ratatouille
3. The Grand Budapest Hotel

Wykresy:



Rys. 47. Wykresy dla algorytmu RSM



## **Podsumowanie serii 2:**

Patrząc na powyższe wyniki pierwszą z rzeczy, która rzuca się w oczy jest fakt, że ranking filmów zwrócony dla metod Topsis oraz UTA jest stosunkowo podobny, natomiast ranking dla metody RSM odbiega od tych dwóch pozostałych zawierając inne filmy. Widać również, że rankingi dla poszczególnych zestawów wag kryteriów są bardzo zbliżone, tak więc wszystkie w podobny sposób wpływają na końcowy ranking dla wszystkich metod. Prawdopodobne przyczyny takiego stanu rzeczy są analogiczne do tych dla eksperymentów dla pierwszej serii danych. Tym razem otrzymane listy filmów nie do końca spełniają wstępne założenia, którymi było otrzymywanie głównie polskich filmów, jednak w żadnej metodzie one nie występują. Przyczyną tej sytuacji może być fakt, że podczas pobierania filmów nie filtrujemy ich po języku lub regionie i zostają wczytane głównie filmy zgraniczne.

## **6. Podsumowanie/wnioski**

### **6.1. Zrealizowane punkty**

W tym projekcie została zrealizowana większość punktów wymienionych w wytycznych. Został opisany problem wielokryterialny oraz jego cel i możliwości wykorzystania go. Zawarta została także informacja o wykorzystanej bazie danych. Ponadto została przeprowadzana akwizycja danych, a więc na przykład niezbędny preprocessing. Zaimplementowane metody optymalizacji parametrów zostały wykorzystane i wdrożone do aplikacji. Jeśli chodzi o wyniki to zostały przedstawione dla dwóch zestawów danych oraz dla kolejnych zmian w wagach kryteriów. Ponadto do każdego zestawu danych zostało przedstawione podsumowanie. Przygotowana aplikacja obsługuje zagadnienie problemowe oraz wizualizuje dane.

### **6.2. Napotkane problemy**

Jeden z pierwszym problemów miał miejsce jeszcze na etapie implementacji metod, głównie chodzi tu o metodę zbiorów odniesienia, gdyż wymagała ona dużego nakładu pracy, czytania i zrozumienia jej idei, co jednak i tak nie było trywialne w przełożeniu na zaimplementowanie tej metody. Kolejnym problemem utrudniającym pracę z aplikacją jest dość długi czas wczytywania filmów z bazy danych, co może wpływać na ich ilość a w dalszej kolejności na jakość otrzymywanego rankingu w odniesieniu do początkowych założeń użytkownika.

### **6.3. Kroki dalszego rozwoju**

Z pewnością jednym z pierwszych kroków dalszego rozwoju byłaby praca nad usprawnieniem i obniżeniem czasu wczytywania filmów, być może mogłoby to być dokonane poprzez pracę z inną niż IMDB bazą danych. Inną możliwością rozwoju byłaby implementacja dodatkowej metody jako metody odniesienia w celu porównania wyników. Kolejnym punktem może być

dodanie większej ilości filtrów podczas wczytywania filmów, tak aby zwiększyć wpływ użytkownika na zwrócone wyniki.

## 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://www.researchgate.net/publication/236153959_Decision_Support_Systems_Based_on_Reference_Sets)

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

## 8. Podział pracy

<b>Funkcjonalność</b>	<b>Artur [wkład w %]</b>	<b>Joanna [wkład w %]</b>	<b>Adrian [wkład w %]</b>
Preprocessing danych	-	50	50
GUI	100	-	-
Implementacja metod	40	30	30
Implementacja wizualizacji wyników	25	25	50
Eksperymenty	-	60	40
Dokumentacja	35	35	30