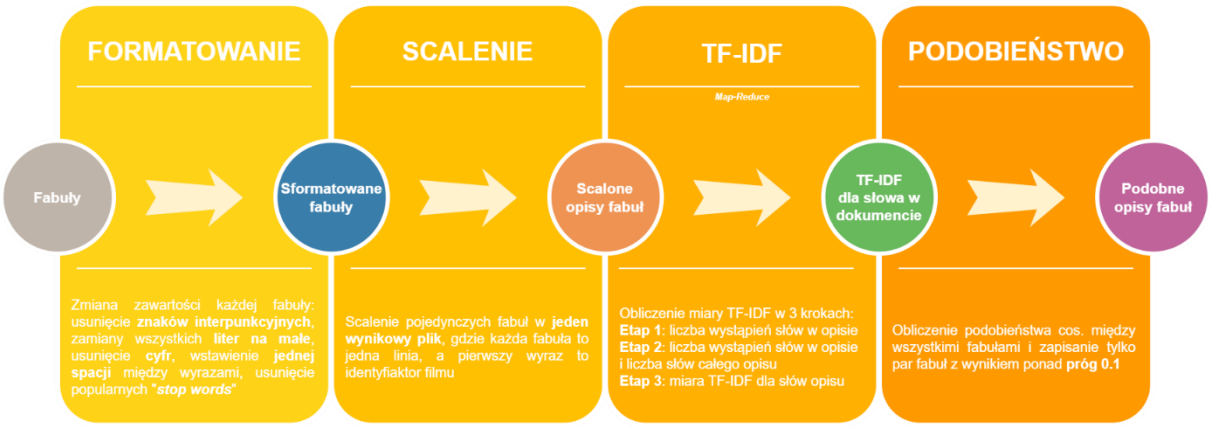
# Silnik rekomendacyjny do filmów

Zespół nr 1

## Ostateczna analiza problemu

**Podobieństwo między fabułami:**

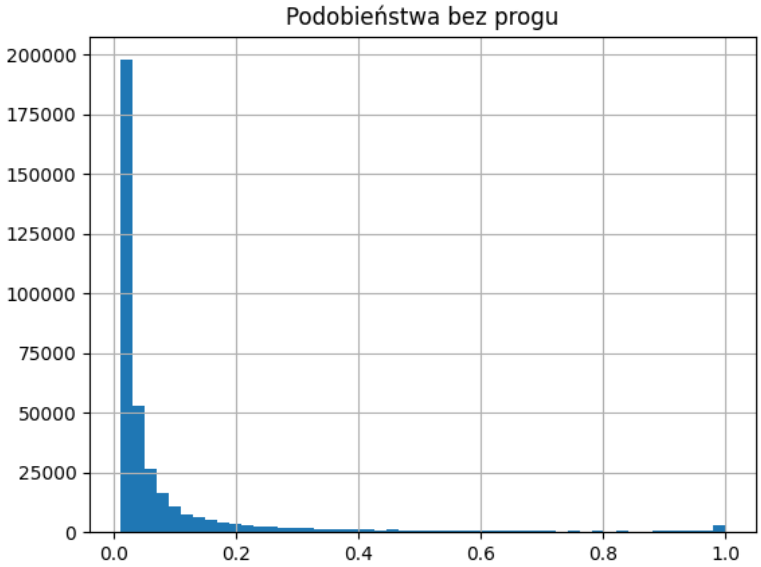
Proces obliczania podobieństwa między fabułami odbywa się w kilku krokach. Najpierw wszystkie pobrane opisy fabuł są formatowane w celu wyodrębnienia wyłącznie wyrazów, które są wartościowe ze względu na dalsze przetwarzanie. Wynikiem takiego działania są ciągi wyrazów zapisanych małymi literami bez znaków interpunkcyjnych i cyfr oddzielonych jedną spacją. Dodatkowo usuwane są wyrazy, które nie wnoszą nowych informacji o opisie, a służą jedynie do budowy zdań (tzw. “stop words”). Opisy przygotowane w ten sposób są następnie scalane do jednego pliku umożliwiającego dalsze przetwarzanie. Następnie za pomocą paradygmatu map-reduce obliczane są parametry słów opisów potrzebne do obliczenia podobieństwa. Ten proces jest podzielony na 3 odrębne etapy, których ostatnim wynikiem jest miara TF-IDF słowa dla każdego opisu. Na podstawie tych danych obliczania jest miara podobieństwa cosinusowego dla wszystkich fabuł i zapisywane są jedynie pary przekraczające ustalony próg.

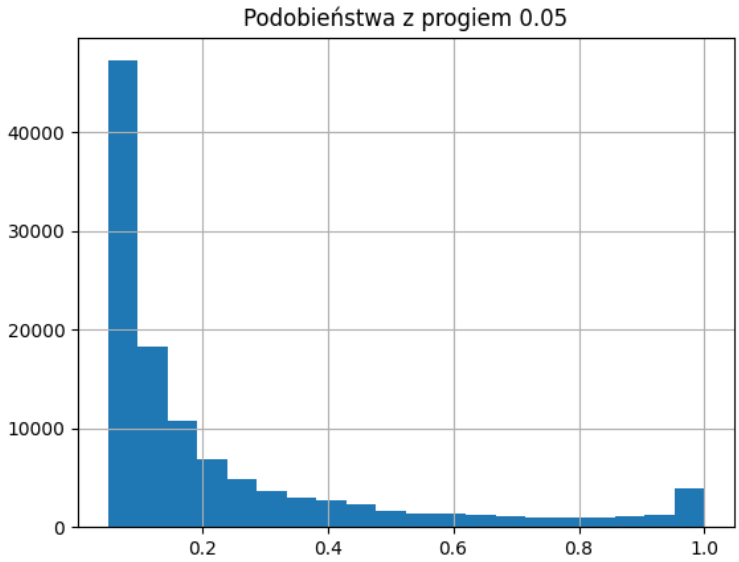



## Osiągnięte rezultaty

**Podobieństwo między fabułami:**

Przejście procesu obliczenia podobieństwa między fabułami dla wszystkich 10000 fabuł było bardzo czasochłonne i wymagało łącznie około 6/7 godzin. Ostatecznie podobieństwa zostały przygotowane z uwzględnieniem progu 0.05. Rozkład uzyskanych wyników pozwala zauważyć, że znaczna większość opisów nie jest do siebie praktycznie wcale podobna, co jest zgodne z przewidywaniami teoretycznymi.




## Interakcja z silnikiem

Silnik nasz wystawia API – tak jak pisaliśmy w poprzednich raportach, zakładamy, że jest on częścią większego serwisu, przyjmuje parametry i zwraca odpowiedzi w formacie JSON. Dla wygody oceny działania wystawiamy też endpoint zwracający niepełne dane, ale w formacie dużo łatwiejszym do odczytania przez człowieka. Wystawiane endpointy i to na co pozwalają:

- `/api/random` – odczytanie z bazy N losowych filmów, skąd klient może wyciągnąć filmy znajdujące się w bazie i na nich bazować rekomendacje
- `/api/info` – zwraca pełne informacje na temat konkretnego filmu
- `/api/recommendations` – zwraca listę ID rekomendowanych filmów, na podstawie listy ID filmów podanych przez klienta
- `/api/rexwithinfo` – ta sama logika biznesowa co wyżej, ale output jest w przystępniejszej dla człowieka formie

## Testy działania silnika

By w pełni pokazać możliwości silnika dodaliśmy do API opcjonalny argument, który sprawia, że silnik przestaje brać podobieństwo fabuł pod uwagę. Dzięki temu można porównać jaki faktycznie wpływ ma nasza tak naprawdę kluczowa funkcjonalność.

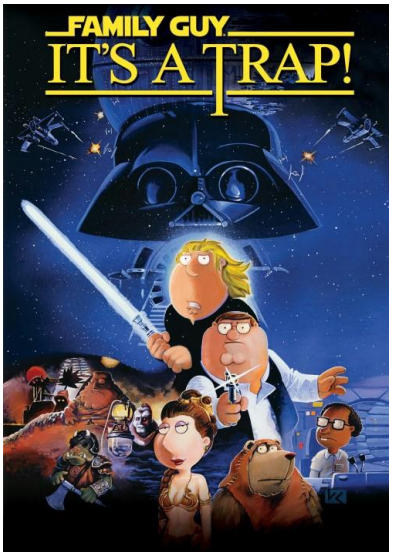
Dla czytelności nie załączyliśmy poniżej zrzutów ekranu. Wystawiamy w końcu API – czytanie JSONów pozostawimy komputerom. Nasza aplikacja jest dostępna publicznie z wykorzystaniem Azure App Service, dlatego przy wszystkich omawianych przykładach dołączyliśmy linki, by zapewnić możliwość weryfikacji opisanych przez nas wyników (ID filmów jednakowe jak w TMDB).

API działa w sposób deterministyczny (najlepsze możliwe rekomendacje, posortowane malejąco po wyniku podobieństwa), więc wyniki nie będą się różnić między odpytaniami.

### Star Wars IV: New Hope

| L.p. | Włączone podobieństwo fabułowe, link | Brak podobieństwa fabułowego, link |
|---|---|---|
| 1 | The Empire Strikes Back | The Empire Strikes Back |
| 2 | Return of the Jedi | Return of the Jedi |
| 3 | Star Wars: The Force Awakens | Star Wars: The Force Awakens |
| 4 | Star Wars: The Rise of Skywalker | Star Wars: The Rise of Skywalker |
| 5 | Star Wars: The Last Jedi | Star Wars: The Last Jedi |
| 6 | Star Wars: Episode I - The Phantom Menace | Ender's Game |
| 7 | **Family Guy Presents: It's a Trap!** | Spider-Man: Into the Spider-Verse |
| 8 | Ender's Game | Inception |
| 9 | Solo: A Star Wars Story | Steven Universe: The Movie |
| 10 | Red Cliff II | Avengers: Endgame |

Star Wars to bardzo popularna seria, w której zagrało wielu ikonicznych aktorów. Nic dziwnego więc, że same podobieństwa gatunkowe i *obsadowe* dobiera inne filmy z tej sagi. Po włączeniu podobieństwa fabułowego jednak rekomendacje są lepiej dopasowane, a na liście znalazła się nawet parodia sagi, w żaden inny sposób z nią nie związana, niż właśnie przez luźne odniesienia w fabule.

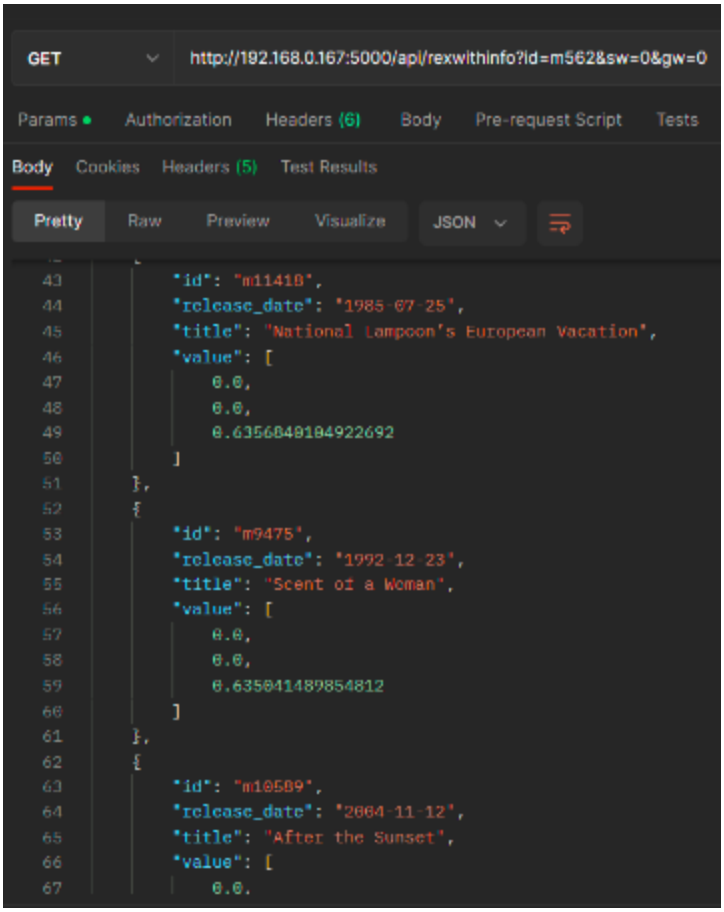



Plakat filmu *Family Guy Presents: It's a Trap!*

**Die Hard**

| L.p. | Włączone podobieństwo fabułowe, link | Brak podobieństwa fabułowego, link | JEDYNIE podobieństwo fabułowe |
|---|---|---|---|
| 1 | Crimson Rivers II: Angels of the Apocalypse | Die Hard 2 | Crimson Rivers II: Angels of the Apocalypse |
| 2 | This Means War | The Fifth Element | This Means War |
| 3 | A Little Chaos | Sin City | A Little Chaos |
| 4 | Airport ’77 | Die Hard: With a Vengeance | Possession |
| 5 | Infinite | Looper | National Lampoon’s European Vacation |
| 6 | Die Hard 2 | Armageddon | Scent of a Woman |
| 7 | After the Sunset | The Last Boy Scout | After the Sunset |
| 8 | Pandora | RED | Airport ’77 |
| 9 | Possession | Planet Terror | Infinite |
| 10 | Blacklight | Live Free or Die Hard | Orbiter 9 |

Przykład ten bardzo dobrze pokazuje, jak duży wpływ na wyniki potrafi mieć liczone przez nas podobieństwo fabuł. Tylko jeden tytuł się pokrywa, a jest to sequel, więc nie jest to szczególnie zadziwiające. Dodatkowa rubryka w tabeli pokazuje wyniki uzyskane przez nasz system w trybie debugowym, przy ustawieniu wag aktorów oraz gatunków na zero, a więc kierowanie się jedynie fabułami. Widzimy, że jest to bardzo zbliżone do ostatecznych wyników, jednak nie odzwierciedla ich w 100%. Oznacza to, że pozostałe aspekty podobieństwa filmów wciąż mają swoją rolę.



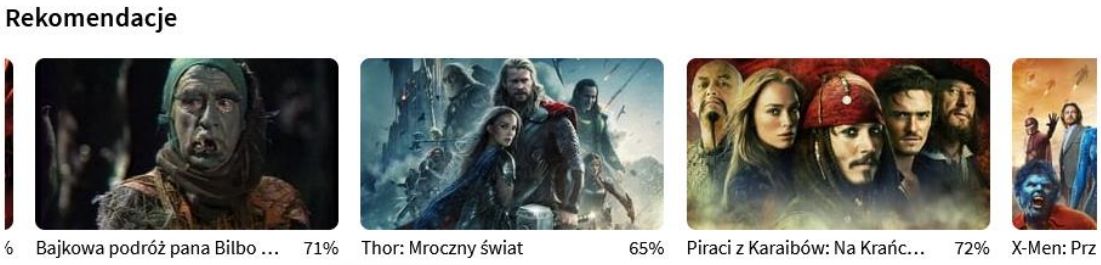

Dodaliśmy tryb debugowy, który zwraca również wartości wszystkich trzech podobieństw. Pozwolił on nam na lepsze zrozumienie z czego wynikają poszczególne polecenia przy procesie fine tuningu. Endpoint ten nie jest publicznie dostępny.

Widzimy tutaj wyniki uzyskane w tym trybie, przy wyzerowaniu wag aktorów oraz gatunków. Obserwujemy wynikowe podobieństwo fabularne przemnożone przez swoją wagę.

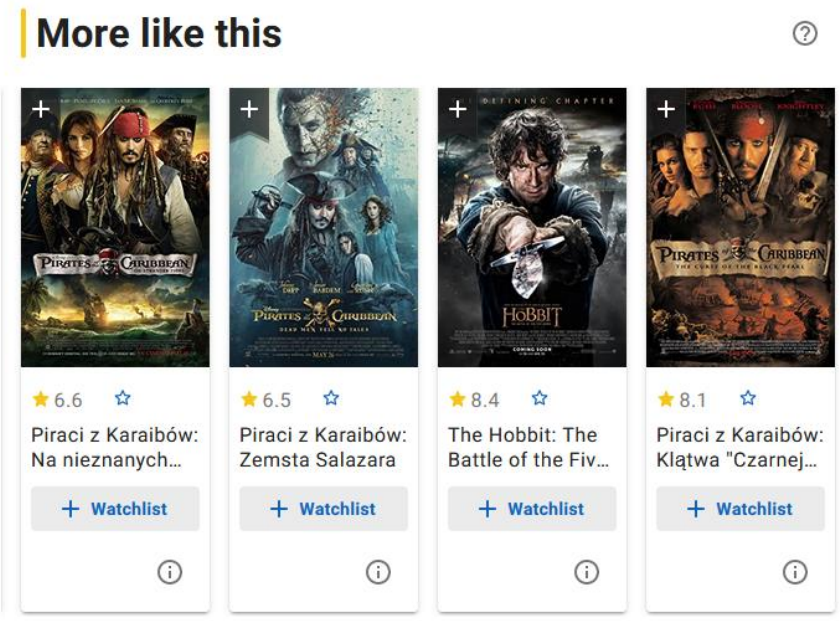
**Hobbit: Pustkowie Smauga**

| L.p. | Włączone podobieństwo fabułowe, link |
|---|---|
| 1 | The Hobbit: The Battle of the Five Armies |
| 2 | The Hobbit: An Unexpected Journey |
| 3 | The Lord of the Rings: The Return of the King |
| 4 | The Lord of the Rings: The Fellowship of the Ring |
| 5 | The Lord of the Rings: The Two Towers |
| 6 | Pirates of the Caribbean: The Curse of the Black Pearl |
| 7 | Doctor Strange |
| 8 | Doctor Strange in the Multiverse of Madness |
| 9 | Pirates of the Caribbean: Dead Man's Chest |
| 10 | Pirates of the Caribbean: At World's End |

Pierwsze 5 rekomendacji nie powinno nikogo dziwić. Kolejne 5 natomiast, mimo że na pierwszy rzut oka trochę losowe, to są również sugerowane przez inne portale filmowe.




Rekomendacje z portalu TMDB, dostępne tutaj




Rekomendacje z portalu IMDB, dostępne tutaj

君の名は **(Kimi no na wa)**

| L.p. | Włączone podobieństwo fabułowe, link |
|---|---|
| 1 | Rascal Does Not Dream of a Dreaming Girl |
| 2 | Josee, the Tiger and the Fish |
| 3 | Violet Evergarden: The Movie |
| 4 | Given |
| 5 | I Want to Eat Your Pancreas |
| 6 | Maquia: When the Promised Flower Blooms |
| 7 | In This Corner of the World |
| 8 | Weathering with You |
| 9 | A Whisker Away |
| 10 | Millennium Actress |

Wyniki są jak najbardziej zadowalające, gdyż duża większość zarekomendowanych filmów jest w zbliżonym klimacie oraz podobna w odbiorze. Widzimy zatem, że nasz silnik potrafi udzielić dobrych rekomendacji nie tylko dla globalnie popularnych klasyków.

## Co zostało zmienione w ramach ostatnich szlifów

- Liczne testy, dopracowanie wag podobieństw
- Kolejność wyświetlanych wyników jest teraz wedle stopnia podobieństwa (malejąco)
- Dodano parametr pozwalający na wyłączenie wpływu podobieństwa fabularnego
- Dodano parametr określający liczbę zwracanych wyników

## Wnioski

1) Przetwarzanie dużej liczby danych w wielu etapach jest procesem bardzo czasochłonnym i podatnym na utratę danych. Z tego powodu niezbędne jest ciągłe zapisywanie kopii zapasowych poszczególnych wyników każdego etapu i maksymalna automatyzacja procesu.
2) Oprogramowanie Hadoop HDFS i Hadoop MapReduce jest narzędziem trudnym w konfiguracji ze względu na skomplikowany proces raportowania przyczyn błędów. Niemniej jednak sama obsługa działającego poprawnie programu jest intuicyjna i daje możliwość bezawaryjnego i wydajnego przetwarzania danych.
3) Stworzenie prostej aplikacji we Flasku okazało się bardzo problematyczne. Błędy pojawiały się prawie na każdym kroku podjętej próby. Próba wykorzystania Pythona 3.8, 3.9 oraz 3.11 zakończyły się niepowodzeniem z powodu odmiennych niekompatybilności. Całość wynikała z ubogiej implementacji Gremlina na platformie Azure, co wymusiło ostatecznie wykorzystanie niewspieranych już bibliotek.

## Dalsze możliwości rozwoju

1) Proces obliczania podobieństwa między fabułami w omawianym projekcie posiada duży potencjał zwiększenia wydajności i jakości otrzymywanych wyników, a cały proces mógłby zostać uproszczony. Formatowanie i scalanie poszczególnych opisów mogłoby się odbywać już na poziomie pobierania fabuł. Takie podejście pozwoliłoby na zaoszczędzenie przestrzeni dyskowej i uproszczenie samego procesu. Dodatkowo proces obliczania samego podobieństwa również mógłby zostać napisany w oparciu o paradygmat map-reduce, co zwiększyłoby wydajność i zapewniło większą bezawaryjność. Warto byłoby również

zastosować inne miary podobieństwa między opisami fabuł, co z pewnością zwiększyłoby wiarygodność wyników.

2) Elementem, który warto byłoby w projekcie rozwinąć, jest strumieniowe dodawanie nowych filmów do bazy w miarę ich powstawania. Chcielibyśmy przecież, by najnowsze produkcje również mogły zostać polecone użytkownikom.
3) Również wartym uwagi kierunkiem rozwoju byłoby zintegorwanie naszego systemu opartego na *Content-based filtering* z innym, opartym na *Collaborative filtering*, tworząc w ten sposób system hybrydowy. Pozwoliłoby to na znajdowanie filmów nie tylko podobnych do tego co nam przypadło do gustu, ale w dodatku takich, które są pozytywnie odbierane przez podobnych nam odbiorców. Samo zastosowanie *Content-based filtering* tego bowiem nie zapewnia. Pozwoliłoby to również na zwiększenie ogólnej skuteczności odnajdywania filmów podobnych, bo nie opieralibyśmy się jedynie na opisach, które nie zawsze muszą być wystarczająco trafne czy szczegółowe, ale również na odczuciach i spostrzeżeniach społeczności.
4) Kolejnym elementem zwiększającym atrakcyjność projektu byłoby dodanie interfejsu użytkownika w postaci aplikacji internetowej. Ułatwiłoby to korzystanie z narzędzia i uzyskiwanie rekomendacji osobom nieuczestniczącym w projekcie, nie rozumiejącym metod w nim wykorzystywanych.
5) Jako że obecnie przy zapytaniu o rekomendację do kilku filmów jednocześnie zwracana jest suma rekomendacji dla każdego z filmów z osobna, warto byłoby wprowadzić miarę podobieństwa do kilku filmów jednocześnie. Można by spróbować zaimplementować to za pomocą algorytmu k najbliższych sąsiadów.

## Skład zespołu


- Michał Banaszczak
- Patryk Chojnicki
- Weronika Skiba
- Daniel Ślusarczyk


Repozytorium: [movie-recs-engine @ GitHub]