



# REKOMENDER GIER WIDEO

Projekt – Usługi Sieciowe

**Autorzy:**

Karolina Alisz (181227)  
Magdalena Pietrzak (182146)  
Krystian Wachowski (190017)

[https://github.com/Stiwenoo/US\\_Projekt/tree/main](https://github.com/Stiwenoo/US_Projekt/tree/main)

## Spis treści

Wstęp do problematyki .....	2
Rozwinięcie problemu .....	3
Złożoność wyboru i saturacja rynku .....	3
Indywidualne preferencje i subiektywne oczekiwania .....	3
Personalizacja rekomendacji i techniki uczenia maszynowego .....	3
Wyzwania związane z jakością danych i integracją źródeł .....	3
Ewaluacja skuteczności i zaufanie użytkowników .....	4
Metodologia .....	5
Baza danych .....	5
Opis Działania Aplikacji .....	6
Wyniki .....	7
Know-How .....	7
API .....	7
Interfejs użytkownika (UI) .....	8
Wyniki rekomendacji .....	9
Pliki .....	9
Kroki wdrożeniowe .....	10
Dyskusja i przyszłe działania .....	12
Przyszłe działania .....	12
Podsumowanie .....	12

## Wstęp do problematyki

Współczesna przemysł rozrywki elektronicznej, w szczególności gry wideo, rozwija się w zawrotnym tempie, oferując milionom użytkowników na całym świecie szeroki wybór tytułów do wyboru. Jednakże, z tak ogromną ilością gier dostępnych na rynku, znalezienie tych, które najlepiej odpowiadają indywidualnym preferencjom graczy, staje się wyzwaniem.

Rekomender gier wideo można postrzegać jako narzędzie wspomagające decyzje użytkowników w procesie selekcji gier, które są najbardziej zgodne z ich gustami i oczekiwaniami. Takie systemy rekomendacji wykorzystują zaawansowane techniki analizy danych i uczenia maszynowego, aby sugerować gry na podstawie wcześniejszych preferencji użytkowników oraz charakterystyk samych tytułów.

Celem tego projektu jest eksploracja i implementacja rekomendera gier wideo opartego na danych. Projekt ten czerpie inspirację z istniejących rozwiązań oraz korzysta z publicznie dostępnych zbiorów danych dotyczących gier wideo. Analiza takich danych pozwoli na identyfikację wzorców i preferencji wśród użytkowników oraz umożliwi budowę modeli predykcyjnych, które mogą dokładnie przewidywać, jakie gry mogą się spodobać konkretnemu użytkownikowi.

W niniejszym raporcie przedstawimy proces tworzenia oraz ocenę skuteczności stworzonego rekomendera, zaczynając od zbierania danych i ich przetwarzania, przez wybór odpowiednich algorytmów i technik uczenia maszynowego, aż po testowanie i ocenę końcowych wyników. Ostatecznie, naszym celem jest stworzenie narzędzia, które nie tylko pomoże w lepszym odkrywaniu gier wideo.

## Rozwinięcie problemu

Rynek gier wideo, będący dynamicznie rozwijającym się sektorem przemysłu rozrywkowego, charakteryzuje się ogromną różnorodnością zarówno pod względem liczby dostępnych tytułów, jak i ich specyficznych cech, które determinują odbiór przez użytkowników. Zjawisko to, znane jako "problem nadmiaru wyboru", stawia przed graczami wyzwanie w doborze gier, które najlepiej odpowiadają ich indywidualnym preferencjom i oczekiwaniom rozrywkowym.

## Złożoność wyboru i saturacja rynku

Wzrost liczby gier dostępnych na rynku, obejmujący zarówno produkcje o wysokim budżecie, jak i niezależne tytuły, prowadzi do sytuacji, gdzie wybór odpowiedniej gry staje się kwestią niezwykle złożoną. Dla nowych graczy oraz dla doświadczonych entuzjastów, proces selekcji staje się wyzwaniem ze względu na nadmierną ilość opcji, co wymaga efektywnych narzędzi wspierających decyzje zakupowe i wybór gier.

## Indywidualne preferencje i subiektywne oczekiwania

Preferencje graczy są kształtowane przez różnorodne czynniki, takie jak gatunek gry (np. RPG, strzelanka, symulacja), mechanika rozgrywki (np. akcja, strategia, rozwój postaci), fabuła, estetyka graficzna oraz aspekty społeczne (np. współpraca czy rywalizacja). Złożoność tych preferencji sprawia, że jedna metoda oceny i rekomendacji gier nie jest wystarczająca, a konieczne stają się systemy personalizacji rekomendacji.

## Personalizacja rekomendacji i techniki uczenia maszynowego

Tradycyjne metody rekomendacyjne, takie jak filtrowanie kolaboracyjne czy systemy oparte na treści, coraz częściej uzupełniane są przez zaawansowane techniki uczenia maszynowego. Takie podejścia analizują szeroki zakres danych behawioralnych i metadanych gier, aby dostosować rekomendacje do indywidualnych potrzeb użytkowników. Przykłady takich danych obejmują oceny użytkowników, historię gier zakupionych lub odtworzonych, czas spędzony na poszczególnych tytułach oraz interakcje społeczne w grach online.

## Wyzwania związane z jakością danych i integracją źródeł

Efektywne działanie systemów rekomendacyjnych wymaga nie tylko zaawansowanych algorytmów uczenia maszynowego, ale także dostępu do wysokiej jakości danych. Kluczowe jest integrowanie różnych źródeł danych, zapewniając jednocześnie ich dokładność i spójność. Wykorzystanie danych pochodzących z różnych platform sprzedaży gier, recenzji, społeczności graczy oraz danych behawioralnych staje się niezbędnym do budowy wiarygodnych modeli rekomendacyjnych.

## Ewaluacja skuteczności i zaufanie użytkowników

Istotnym aspektem projektowania rekomenderów gier wideo jest ich ocena pod kątem skuteczności oraz akceptacji przez użytkowników. Skuteczność mierzona jest przez wskaźniki takie jak trafność (precision) i pokrycie (recall) rekomendacji, które powinny być adekwatne do różnych kontekstów i preferencji. Ponadto, kluczowe jest budowanie zaufania użytkowników poprzez transparentność działania systemu oraz odpowiednie zarządzanie ich oczekiwaniami w kontekście rekomendacji.

W kontekście realizacji projektu rekomendacji gier wideo opartego na danych, głównym celem jest nie tylko opracowanie skutecznego narzędzia wspierającego decyzje graczy, ale również wkład w rozwój technologii analizy danych i uczenia maszynowego w kontekście personalizowanych rekomendacji produktów w branży rozrywki elektronicznej. Poprzez integrację zaawansowanych technik analitycznych z głębszym zrozumieniem indywidualnych preferencji użytkowników, możliwe jest stworzenie bardziej satysfakcjonujących i zindywidualizowanych doświadczeń z grami wideo.

# Metodologia

## Baza danych

Do realizacji projektu wykorzystano dane udostępnione na platformie kaggle.com<sup>1</sup>. Zestaw danych oparty jest na grach dostępnych w sklepie Steam. Zawiera informacje na temat różnych aspektów gier dostępnych w sklepie, takich jak gatunek gry oraz szacowana liczba jej posiadaczy. Zebrano wszystkie dane do maja 2019 roku, zawierają większość gier dostępnych w sklepie wydanych przed tą datą.

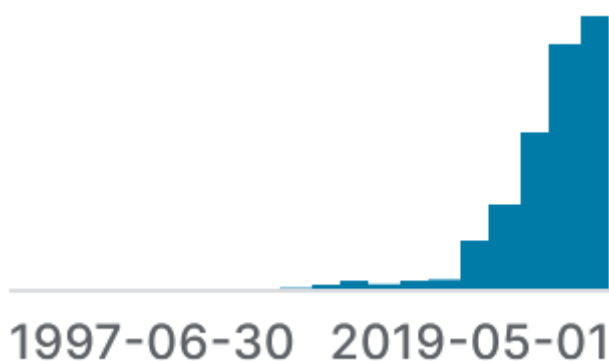
Z datasetu wybrano główny plik o nazwie steam.csv. Każdy rekord zawiera unikalne AppID, które stanowi osobą publikację gry.

Z pliku wybrano następujące zmienne:

1. Name – tytuł gry
2. Release\_date – data wydania
3. Categories – kategorie
4. Developer – nazwa dewelopera
5. Publisher – nazwa wydawcy
6. Genres – gatunki
7. steamspy\_tags – gatunki wybrane przez graczy

Plik zawiera łącznie 27033 unikatowych tytułów, 17113 różnych deweloperów i 14354 wydawców. Kategorie gier dzielą się m.in. na gry typu „singleplayer” i „multiplayer”. Udział gier jednoosobowych w całym zbiorze to 23%. Najpopularniejszymi grami są produkcje niszowe (Indie) oraz gry akcji. Steamspy\_tags to tworzone na podstawie rekomendacji graczy, trzy najbardziej dopasowane do danej gry tagi.

Produkcja gier wideo w latach 1997-2019 przedstawia się następująco.



Rysunek 1. Ilość gier na platformie steam wg daty publikacji

<sup>1</sup> <https://www.kaggle.com/datasets/nikdavis/steam-store-games/data?select=steam.csv>

Źródło: <https://www.kaggle.com/datasets/nikdavis/steam-store-games/data?select=steam.csv>

Na rysunku nr 1 można zauważyć wzrostową tendencję ilości gier wideo od roku 1997.

## Opis Działania Aplikacji

Do celów projektu stworzono rekomender gier wideo oparty na filtrowaniu zawartości (content-based filtering). To aplikacja, która analizuje zawartość różnych gier, aby sugerować użytkownikowi inne gry, które mogą mu się spodobać. Aplikacja działa na podstawie danych zawierających szczegółowe informacje o grach, takie jak kategorie, gatunki, deweloperzy, wydawcy i tagi. Używamy modelu podobieństwa kosinusowego do porównywania gier i generowania rekomendacji.

# Wyniki

## Know-How

### 1. Przygotowanie danych:

- Dane dotyczące gier są wczytywane z plików CSV i przetwarzane, aby usunąć niepotrzebne kolumny i oczyścić dane z niepotrzebnych znaków i tagów HTML.
- Dane są następnie łączone w jedną tabelę, w której każdy wiersz reprezentuje jedną grę wraz z jej cechami.

### 2. Stworzenie modelu rekomendacji:

- Tworzymy nową kolumnę combined, która łączy różne cechy gry w jeden tekstowy ciąg znaków.
- Używamy CountVectorizer do przekształcenia tej kolumny w macierz liczb, a następnie obliczamy podobieństwo kosinusowe między wszystkimi grami.
- Model podobieństwa jest zapisywany do pliku za pomocą pickle.

### 3. API:

- Aplikacja Flask udostępnia REST API, które umożliwia użytkownikowi uzyskanie rekomendacji na podstawie podanego tytułu gry.
- API jest uruchamiane na serwerze lokalnym i umożliwia wysyłanie żądań HTTP GET, które zwracają listę rekomendacji.

## API

Endpoint: /recommend

Metoda: GET

Parametry:

title (string): Tytuł gry, dla której chcemy uzyskać rekomendacje.

Opis: Zwraca listę rekomendowanych gier na podstawie podanego tytułu gry. Jeśli tytuł nie jest znaleziony, API zwraca podobne tytuły gier, które mogły być błędnie wpisane.

Przykład żądania: `http://127.0.0.1:5000/recommend?title=Tropico%204`

Przykład odpowiedzi



```
{
  "recommendations": [
    "Tropico 4",
    "Tropico 5",
    "SimCity",
    "Cities: Skylines",
    "Anno 2070",
    "Civilization V",
    "Banished",
    "Planet Coaster",
    "Prison Architect",
    "Age of Empires II"
  ]
}
```

Przykład odpowiedzi, gdy tytuł jest błędny:

```
{
  "error": "Game title not found. Did you mean: ['tropico 4', 'tropico 5']"
}
```

## Interfejs użytkownika (UI)

UI aplikacji jest prosty i opiera się na interakcji za pomocą zapytań HTTP. Można jednak łatwo rozszerzyć aplikację o interfejs graficzny, np. przy użyciu HTML, CSS, i JavaScript, lub frameworków frontendowych takich jak React, Vue.js, czy Angular.

Przykładowa implementacja może wyglądać następująco:

- Strona główna:
  - Prosty formularz z polem tekstowym, gdzie użytkownik może wpisać tytuł gry.
  - Przycisk "Rekomenduj", który wysyła zapytanie do API.
- Wyświetlanie wyników:
  - Sekcja wyników, która wyświetla rekomendowane gry w formie listy.

# Game Recommender

Enter game title:

## Recommendations:

- Cities: Skylines
- Prison Architect
- Valhalla Hills
- Rebuild 3: Gangs of Deadsville
- Planetbase
- Tropico 5
- Anno 2205™
- Space Colony: Steam Edition
- Parkitect
- Terraform
- Car Mechanic Simulator 2018

*Rysunek 2. UI Aplikacji*

Źródło: Opracowanie własne.

## Wyniki rekomendacji

Aplikacja została przetestowana na zestawie danych zawierającym szczegółowe informacje o wielu grach wideo. Po przetworzeniu danych i utworzeniu modelu podobieństwa, API było w stanie generować trafne rekomendacje gier na podstawie podanego tytułu. Funkcja znajdowania podobnych tytułów poprawia doświadczenie użytkownika, sugerując poprawne tytuły w przypadku literówek lub błędów.

## Pliki

app.py

- Główna aplikacja Flask uruchamiająca serwer i obsługująca żądania API.

exp\_cbr.py

- Zawiera funkcje do generowania modelu rekomendacji i znajdowania podobnych gier.

data\_manipulation.py

- Skrypty do wstępnego przetwarzania danych i tworzenia ostatecznego zestawu danych do modelowania.

## Kroki wdrożeniowe

Przygotowanie środowiska:

- Upewnij się, że masz zainstalowane wszystkie wymagane biblioteki: Flask, pandas, sklearn, nltk.
- Struktura katalogów:
  - app/ - zawiera app.py
  - data/ - zawiera pliki cbr\_model.pkl, rec\_games\_more.pkl oraz pliki CSV z danymi
  - experiments/ - zawiera data\_manipulation.py i exp\_cbr.py
- Stworzenie modelu (krok opcjonalny)
  - Wywołanie pliku generate\_cbr.py z folderu experiments
  - Jeśli nie wygenerujesz modelu to podczas uruchamiania, model automatycznie się wygeneruje, co może spowodować dłuższą inicjalizację serwera.
- Uruchomienie aplikacji:
  - Przejdź do katalogu app/ i uruchom app.py:  
~ python app.py

```
PS C:\Users\karol\Repozytoria\US_Projekt> python app/app.py
exp_cbr: Importing necessary libraries...
exp_cbr: Functions defined.
app: Importing recommend function...
app: Recommend function imported successfully.
app: Loading model...
app: Model file not found. Generating model...
exp_cbr: Starting model generation...
app: Loading data...
app: Data file opened.
* Serving Flask app 'app'
* Debug mode: on
app: Data loaded.
exp_cbr: Data loaded, creating CountVectorizer...
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
exp_cbr: Calculating cosine similarity...
exp_cbr: Importing necessary libraries...
exp_cbr: Functions defined.
app: Importing recommend function...
app: Recommend function imported successfully.
app: Loading model...
app: Model file not found. Generating model...
exp_cbr: Starting model generation...
app: Loading data...
app: Data file opened.
app: Data loaded.
exp_cbr: Data loaded, creating CountVectorizer...
* Debugger is active!
* Debugger PIN: 162-127-527
exp_cbr: Calculating cosine similarity...
exp_cbr: Model saved.
app: Loading model...
app: Model file opened.
app: Model loaded.
```

Rysunek 3. Uruchamianie aplikacji (printy z konsoli)

Źródło: Opracowanie własne.

- Aplikacja będzie dostępna pod adresem <http://127.0.0.1:5000>.
- Rekomender będzie działać dopiero po załadowaniu modelu „App: model loaded”

```
app: Loading model...  
app: Model file opened.  
app: Model loaded.
```

*Rysunek 4. Status informujący o załadowaniu modelu*

Źródło: Opracowanie własne.

# Dyskusja i przyszłe działania

## Przyszłe działania

1. Model podobieństwa może być dalej udoskonalany poprzez uwzględnienie innych cech gier lub zastosowanie bardziej zaawansowanych technik przetwarzania języka naturalnego.
2. Dodanie funkcji autouzupełniania w trakcie wpisywania tekstu.
3. Poprawa pod kątem wizualnym interfejsu UI.

## Podsumowanie

Podczas realizacji projektu rekomendacji gier wideo opartego na danych, kluczowym aspektem było zastosowanie filtrowania zawartości (content-based filtering) do generowania rekomendacji. Przy użyciu zaawansowanych technik przetwarzania danych i uczenia maszynowego, udało się skutecznie zbudować model rekomendacyjny, który analizuje cechy różnych gier i sugeruje użytkownikowi podobne tytuły.

Metodologia projektu obejmowała kilka kluczowych etapów, począwszy od przetwarzania danych, poprzez tworzenie modelu rekomendacji opartego na podobieństwie kosinusowym, aż do implementacji interfejsu API przy użyciu frameworka Flask. Interfejs użytkownika został zaprojektowany w sposób prosty, umożliwiając użytkownikowi łatwe wprowadzanie tytułów gier i otrzymywanie rekomendacji w formie listy.

Wyniki aplikacji zostały przetestowane na rzeczywistych danych dotyczących gier wideo z platformy Steam, co potwierdziło skuteczność modelu w generowaniu trafnych rekomendacji. Funkcja znajdowania podobnych tytułów okazała się szczególnie użyteczna, zapewniając użytkownikom poprawne sugestie nawet w przypadku błędów w wprowadzanych tytułach.