Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Instytut Automatyki i Informatyki Stosowanej

Praca dyplomowa magisterska

na kierunku Informatyka
w specjalności Inteligentne systemy

Analiza danych telemetrycznych z zastosowaniem systemu rekomendacji audycji telewizyjnych

Michał Wojciech Wiśniewski
Numer albumu 276923

promotor
dr hab. inż. Mariusz Kamola

WARSZAWA 2023

Analiza danych telemetrycznych z zastosowaniem systemu rekomendacji audycji telewizyjnych

Streszczenie. Wraz z rozwojem technologii natłok informacji, który nas otacza stale się zwiększa. W ciągu ostatnich lat coraz więcej pozycji pojawia się w ofercie sklepów internetowych, serwisów internetowych, czy też serwisów VOD. Podobny problem spotyka odbiorców telewizji, którzy z uwagi na liczbę oraz różnorodność emitowanych audycji mogą mieć problem z wyborem interesujących ich pozycji. Na potrzeby opisywanej pracy stworzono system rekomendacji audycji telewizyjnych, który potencjalnie pomógłby użytkownikowi w podjęciu decyzji o tym, co mółby obejrzeć. System bazuje na rzeczywistych danych telemetrycznych z zamontowanych w gospodarstwach domowych urządzeń abonenckich przeznaczonych do odbioru telewizji. Stworzony system rekomendacji opiera się na metodzie *collaborative filtering* oraz technice faktoryzacji macierzy. Opisane w pracy badania starają się zweryfikować działanie oraz użyteczność takiego mechanizmu rekomendacji.

Słowa kluczowe: dane telemetryczne, systemy rekomendacji, metoda *collaborative filtering*, faktoryzacja macierzy, algorytm naprzemiennych najmniejszych kwadratów, oceny niejawne

Analysis of telemtric data using a TV program recommender system

Abstract. With the development of technology, the amount of information that surrounds us is constantly increasing. Over the last few years, more and more items appear in the offer of online stores, online services, or VOD services. A similar problem is encountered by television viewers who, due to the number and diversity of broadcasts, may have a problem with choosing the items they are interested in. For the purpose of the described work, a TV program recommender system, which would potentially help the user in making a decision about what to watch, was created. The system is based on real telemetric data from subscription devices installed in the households intended for television reception. The created recommendation system is based on the collaborative filtering method and the matrix factorization technique. The research described in the paper tries to verify the mechanism and usefulness of such a recommendation system.

Keywords: telemetric data, recommender systems, collaborative filtering, matrix factorization, alternating least squares, implicit ratings

Spis treści

1. Wstęp

1.1. Systemy rekomendacji - Wprowadzenie

W ciągu ostatnich lat wraz z rozwojem technologii natłok informacji, który nas otacza stale się zwiększa. Z roku na rok coraz więcej pozycji pojawia się w ofercie sklepów internetowych, serwisów internetowych, czy też serwisów VOD. Z tego względu użytkownicy takich systemów muszą zmagać się z nadmiarem możliwości wyboru i dlatego coraz ciężej jest im odnaleźć najbardziej trafiające w ich gusta treści. Na przeciw temu wychodzi rozwój domeny *systemów rekomendacji* (ang *recommender systems*). Systemy te prezentują użytkownikom sugestie tych pozycji, które mogą ich potencjalnie zainteresować. Przykłady wykorzystania systemów rekomendacji możemy zaobserwować tak naprawdę już na każdym kroku, na przykład w postaci proponowanych produktów z oferty w sklepach internetowych, czy sugestii następnych filmów lub seriali do obejrzenia w serwisach VOD.

Działanie systemów rekomendacji bazuje na analizie historycznych danych o preferencjach użytkowników do danych produktów. Preferencje mogą być wyrażone na różne sposoby. Pierwszym, a zarazem najprostszym z nich, są *oceny jawne* (ang. *explicit ratings*), które bazują na odpowiedzi zwrotnej użytkownika (na przykład w postaci oceny danego produktu w skali numerycznej). Jednak często zebranie takich odpowiedzi nie jest możliwe (przykładowo przez pewne limitacje systemowe, koszt wdrożenia funkcjonalności zbierania takich odpowiedzi, czy brak chęci użytkowników do zostawiania takich ocen). Z tego powodu wykorzystuje się również tzw. *oceny niejawne* (ang. *implicit ratings*), które niebezpośrednio opisują opinie użytkownika o produkcie poprzez analizę jego zachowań (historię zakupów, historię wyszukiwań, czas spędzony na stronie).

Wśród systemów rekomendacji istnieje podział na różne metody w zależności od sposobu ich działania. Wiodącymi metodami jest podejście bazujące na *analizie cech produktów* (ang. *content-based filtering systems*) oraz podejście bazujące na *powiązaniach pomiędzy użytkownikami* (ang. *collaborative filtering systems*).

Systemy *content-based filtering* tworzą sugestie produktów podobnych do tych, które użytkownik pozytywnie ocenił w przeszłości. Podobieństwo obliczane jest na podstawie pewnego zbioru cech produktów. Na przykład, gdy użytkownik pozytywnie ocenił film należący do gatunku komedii, na podstawie tego system będzie sugerował mu filmy należące do tej kategorii.

Natomiast systemy *collaborative filtering* działają na podstawie analizy zależności pomiędzy użytkownikami (alternatywnie pomiędzy produktami). Systemy te identyfikują użytkowników o podobnych gustach, którzy podobnie ocenili konkretne produkty (alternatywnie identyfikują produkty, które zostały podobnie ocenione) i na tej podstawie tworzą rekomendacje. Przykładowo, jeśli dwóch użytkowników podobnie oceniło pewien zbiór produktów, na tej podstawie możemy zidentyfikować pewne podobieństwo pomiędzy

nimi. Następnie, jeśli jeden produkt został oceniony tylko przez pierwszego użytkownika, istnieje duże prawdopodobieństwo, że zostanie podobnie oceniony przez drugiego z nich.

1.2. Cel i założenia badań

Elektroniczny przewodnik po programach (ang. *electronic program guide*, EPG) jest obecnie głównym narzędziem za pomocą, którego użytkownicy telewizji mogą wybrać interesujące ich audycje. Jednak liczba programów telewizyjnych oraz różnorodność udostępnionych przez nie audycji może sprawić, że proces ten będzie uciążliwy.

W pracy przedstawiony został system rekomendacji audycji telewizyjnych opierający się na metodzie *collaborative filtering*, który potencjalnie pomógłby użytkownikowi w podjęciu decyzji o tym, co mógłby obejrzeć. W rezultacie przekładałoby się to na zwiększenie jego satysfakcji z korzystania usług telewizyjnych.

System bazuje na rzeczywistych *danych telemetrycznych* z zamontowanych w gospodarstwach domowych urządzeń abonenckich przeznaczonych do odbioru telewizji. Zgromadzone dane telemetryczne opisują zarejestrowane informacje o czasie włączenia i wyłączenia telewizora oraz identyfikatorze aktualnie odbieranego programu telewizyjnego. Na podstawie tych informacji opisywane są preferencje abonentów do poszczególnych audycji w postaci przytaczanych wcześniej ocen niejawnych. Oceny te służą do stworzenia modelu rekomendacji, który będzie w stanie zaprezentować użytkownikom sugestie audycji, którymi mogą być potencjalnie zainteresowani.

Opisywane w tej pracy badania starają się zweryfikować działanie oraz użyteczność takiego systemu. Badania opisują również sposób w jaki zmierzono się ze związanymi z tym wyzwaniami w postaci:

- pierwszym problemem, który trzeba zaadresować jest pierwotna postać oraz pochodzenie danych wejściowych. Dane w czystej postaci zawierają jedynie informacje o oglądanych programach (jednakże w celu stworzenia rekomendacji audycji potrzebne są informacje o audycjach emitowanych na tych programach). Dodatkowo fakt, że pozyskano je z rzeczywistych urządzeń sprawia, że znajduje się w nich wiele szumu, który w celu uniknięcia potencjalnych negatywnych efektów trzeba zniwelować,
- następnym wyzwaniem jest sposób wyznaczania preferencji użytkowników do konkretnych audycji. W przedstawionych badaniach zaprezentowano oraz porównano trzy alternatywne sposoby obliczania ocen niejawnych, które na podstawie analizy zachowań starają się opisać zainteresowanie daną audycją,
- bardzo ważny jest również wybór metody, która pomoże w ocenie skuteczności działania systemu rekomendacji. W pracy zaproponowano i opisano dwie miary oceny jakości rekomendacji.

Praca została zorganizowana w następujący sposób: w rozdziale 2. przedstawiono ideę oraz koncepcję działania systemów rekomendacji. W rozdziale 3. opisano wykorzystane na

potrzeby pracy biblioteki programistyczne. W rozdziale 4. przedłożono strukturę danych wejściowych wykorzystanych w stworzeniu oraz oceny jakości modelu systemu rekomendacji. Natomiast w rozdziale 5. omówiono jego architekturę oraz metodę sprawdzenia jego skuteczności. Rozdział 6. prezentuje rezultaty badań. Praca została podsumowana w rozdziale 7.

2. Systemy rekomendacji

2.1. Idea systemów rekomendacji

Systemy rekomendacji (ang. *recommender systems*) są to narzędzia, które mają na celu prezentowanie treści dostosowanych do zainteresowań i potrzeb użytkowników. W zależności od środowiska w jakim działa system rekomendacji prezentowanymi treściami mogą być na przykład:

- w przypadku serwisów VOD - filmy lub seriale,
- w przypadku sklepów internetowych - konkretne produkty z oferty,
- w przypadku serwisów muzycznych - utwory muzyczne.

Istnieje wiele powodów, z których dostawcy treści decydują się na wdrożenie technologii systemów rekomendacji. Między innymi mogą nimi być [1]:

- *zwiększenie liczby skonsumowanych treści* (np. liczby zakupionych produktów w sklepie),
- *zwiększenie dywersyfikacji konsumowanych treści* (sugerowanie użytkownikom mniej popularnych treści, które byłyby trudne do odnalezienie bez konkretnej rekomendacji),
- *zwiększenie satysfakcji użytkownika* (poprawa doświadczeń z użytkowania danego sklepu, czy też serwisu internetowego),
- *zwiększenie przywiązania użytkownika do systemu* (użytkownik będzie bardziej przywiązany do systemu, który proponuje mu spersonalizowane treści, które przypadają w jego gusta).

2.2. Metody zbierania preferencji użytkowników

W związku z głównym zadaniem omawianych systemów, czyli prezentowania użytkownikom pewnych rekomendacji, muszą one zbierać informacje na temat historii zachowań i preferencji tych użytkowników. Informacje te możemy interpretować jako ich *oceny* poszczególnych treści.

Jednym typem ocen są tzw. *oceny jawne* (ang. *explicit ratings*). W tym przypadku użytkownik bezpośrednio proszony jest o podzielenie się opinią na temat danej treści w postaci wartości w pewnej skali. Skale takich ocen mogą przyjmować różne formy [2]:

- *skala binarna*, na przykład ocena danego filmu w serwisie YouTube w postaci opcji "To mi się podoba"oraz "To mi się nie podoba",
- *skala numeryczna*, na przykład skala 1-5 gwiazdek w sklepie Amazon,
- *skala porządkowa*, na przykład miara zadowolenia klienta w skali "bardzo nieusatysfakcjonowany, niezadowolony, neutralny, zadowolony, bardzo zadowolony".

Jednak oceny jawne nie zawsze są łatwe lub możliwe do pozyskania (na przykład przez limitacje systemowe). W takich sytuacjach preferencje mogą zostać wyrażone za pomocą

ocen niejawnych (ang. *implicit ratings*). Ich założeniem jest to, że opinia użytkownika na temat poszczególnych treści wnioskowana jest na podstawie historycznych działań i interakcji z systemem [3][4]. Na przykład:

- zakup może oznaczać pozytywne nastawienie do danego produktu,
- czas spędzony na stronie pokazuje zainteresowanie danym artykułem,
- lub w całości obejrzany odcinek serialu wskazuje, że użytkownik lubi daną produkcję.

2.3. Definicja problemu rekomendacji

Problem, który systemy rekomendacji próbują rozwiązać można opisać na różne sposoby. Dwie wiodące definicje problemu rekomendacji to [5][6]:

- *problem predykcji ocen* (ang. *rating prediction problem*), który polega na predykcji oceny, którą użytkownik u przyzna nieocenionej jeszcze treści i. Problem zakłada, że istnieje pewien zbiór *danych treningowych*, który zawiera informacje o preferencjach użytkownika do poszczególnych produktów. Zbiór ten można zdefiniować jako macierz $m \times n$, gdzie m to liczba użytkowników, a n to liczba produktów, która uzupełniona jest zaobserwowanymi wartościami ocen dla par użytkownik-produkt. Wartości zaobserwowane służą do estymacji wartości brakujących (niezaobserwowanych) w tej macierzy,
- *problem rekomendacji K najważniejszych treści* (ang. *top-K recommendation problem*), którego rozwiązanie jest wykorzystywane w przypadkach, gdy estymacja konkretnej wartości oceny nie jest potrzebna (lub jest niemożliwa do predykcji). W takim przypadku problem rekomendacji można zdefiniować jako określenie listy K treści, które najbardziej mogą trafić w gusta użytkownika u.

2.4. Typy systemów rekomendacji

Głównym podziałem systemów rekomendacji jest podział na *systemy spersonalizowane* (ang. *personalized*) - takie, które bazują na analizie historycznych danych o zachowaniach i interakcjach użytkownika oraz *systemy niespersonalizowane* (ang. *non-personalized*) - takie, które nie potrzebują wiedzy na temat preferencji użytkownika, a działają na podstawie ogólnych statystyk popularności danych produktów w danym serwisie.

Wśród spersonalizowanych systemów rekomendacji wiodącymi podejściami są metody bazujące na:

- analizie cech produktów (ang. *content-based filtering*),
- powiązaniach pomiędzy użytkownikami lub produktami (ang. *collaborative filtering*).

Wybór metody jaka zostanie wykorzystana różni się w zależności od zastosowań i od tego, jakie dane są dostępne w danym środowisku. Dodatkowo oba podejścia posiadają swoje wady i zalety. Z tego powodu w praktycznych zastosowaniach często wykorzystuje się tzw.

hybrydowe systemy rekomendacji (ang. *hybrid recommender systems*), czyli takie, które agregują rezultaty z kilku metod.

W opisywanych w tej pracy badania bazują na wykorzystywaniu metody *collaborative filtering*. Jednakże w następnych sekcjach przedstawiono przekrojowy sposób działania obu tych metod.

2.4.1. Metoda *content-based filtering*

Systemy rekomendacji oparte na metodzie *content-based filtering* rozwiązują problem rekomendacji poprzez sugerowanie użytkownikom treści na podstawie podobieństwa do treści pozytywnie ocenionych przez nich w przeszłości. Samo podobieństwo treści bazuje na pewnym zbiorze ich atrybutów. Atrybuty te pozyskiwane są zazwyczaj poprzez metadane lub opis tekstowy danego produktu i są specyficzne dla domeny, w której taki system rekomendacji byłby wprowadzony. Na przykład w domenie serwisów VOD, system rekomendacji może bazować na atrybutach filmu w postaci jego gatunku, reżysera i głównego aktora. Innym przykładem mogą być serwisy informacyjne, które oceniałyby podobieństwo artykułów na podstawie występujących w nich słów kluczowych.

Sam proces rekomendacji systemu *content-based filtering* można uogólnić i zaprezentować w postaci trzech kroków [7][8]:

1. *ekstrakcja atrybutów produktu* - jest to etap, w którym z informacji na temat danego produktu (informacje te są specyficzne dla konkretnej domeny) wyodrębniony zostaje zestaw jego cech (najczęściej w postaci wektorowej),
2. *stworzenie profilu użytkownika* - następnym krokiem jest stworzenie modelu (unikalnego dla jednego użytkownika, nazywanego również profilem użytkownika), który będzie zdolny do predykcji preferencji użytkownika do konkretnych pozycji. Model budowany jest na podstawie *zbioru danych treningowych* składającego się z połączenia informacji o historycznych preferencjach (ocenach) użytkownika z atrybutami produktów wyodrębnionymi w poprzednim kroku. Sam problem stworzenia takiego modelu można interpretować jako problem uczenia maszynowego, na przykład jako problem klasyfikacji, czy też regresji liniowej,
3. *prezentowanie rekomendacji* - finalnie, z wykorzystaniem stworzonego profilu użytkownika możliwe jest stworzenie rekomendacji zbioru produktów (o atrybutach korelujących z profilem użytkownika, które z uwagi na to potencjalnie zainteresują użytkownika).

Zaletą systemów rekomendacji bazujących na analizie cech produktów jest to, że są w stanie rekomendować produkty, które nie zostały jeszcze przez nikogo ocenione. Ważnym aspektem jest również to, że z wykorzystaniem wyodrębnionych cech można w łatwy sposób wytłumaczyć użytkownikowi pojawienie się konkretnej rekomendacji (na przykład rekomendowany film jest tego samego gatunku co inny wcześniej oceniony film).

Natomiast wadą takich systemów jest fakt, że w przypadku pojawienia się nowego

użytkownika nie jest możliwa analiza jego preferencji (nie jest możliwe stworzenie profilu użytkownika), a co za tym idzie system nie będzie w stanie prezentować wiarygodnych rekomendacji. Taka sytuacja nazywana jest również *problemem zimnego startu* (ang. *cold-start problem*) dla użytkowników. Drugą wadą jest różnorodność samych rekomendacji prezentowanych użytkownikom. Z uwagi na fakt, że systemy oparte na metodzie *content-based filtering* bazują na podobieństwie produktów, często rekomendują produkty bardzo do siebie zbliżone.

2.4.2. Metoda *collaborative filtering*

Systemy rekomendacji bazujące na powiązaniach pomiędzy użytkownikami w celu stworzenia rekomendacji dla użytkownika wykorzystują informacje o preferencjach innych użytkowników w systemie. Ideą ich działania jest to, że zainteresowanie użytkownika danym produktem może zostać wywnioskowane na podstawie innego użytkownika, który podobnie oceniał inne produkty. Istnieją dwa typy metod wykorzystywanych w systemach *collaborative filtering*. Są nimi metody *oparte o pamięć* oraz metody *oparte o model* [1]:

1. *metody oparte o pamięć* (ang. *memory-based*) ze względu na sposób działania nazywane są również *metodami opartymi na sąsiedztwie* (ang. *neighborhood-based*). W tym podejściu estymacja ocen dla par użytkownik-produkt obliczana jest na podstawie sąsiedztwa. Same sąsiedztwo może zostać zdefiniowane na dwa sposoby [9][10]:
   - *user-based collaborative filtering* definiuje sąsiedztwo na podstawie podobieństwa użytkowników. W celu znalezienia sąsiedztwa użytkownika u, obliczone zostaje jego podobieństwo do pozostałych użytkowników w systemie. Podobieństwo między użytkownikami u i v może zostać wyrażone np. za pomocą *podobieństwa cosinusowego* (ang. *cosine similarity*)[1]:

     $$sim(u,v) = \frac{\sum_{k \in I_u \cap I_v} r_{uk} \cdot r_{vk}}{\sqrt{\sum_{k \in I_u \cap I_v} r_{uk}^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} r_{vk}^2}} \quad ,$$

     gdzie I_u to zbiór produktów ocenionych przez użytkownika u, a r_{ui} to ocena, którą wystawił produktowi i. *k najbliższych sąsiadów* (ang. *k nearest neighbours*, kNN), pod względem podobieństwa do użytkownika u, służy do estymacji oceny produktów jeszcze przez niego nieocenionych. Predykcja wyznaczona jest jako średnia ważona ocen tego produktu w najbliższym sąsiedztwie,
   - *item-based collaborative filtering* działa na podobnej zasadzie co poprzednik. Z tym, że definiuje sąsiedztwo na podstawie na podstawie podobieństwa produktów. Predykcja oceny użytkownika u dla produktu i wyznaczona jest jako średnia ważona ocen, wystawionych przez u, produktów, które są w najbliższym sąsiedztwie produktu i.

[1] Powrzechnie wykorzystuje się również takie miary jak *współczynnik korelacji Pearsona* (ang. *Pearson correlation coefficient*) oraz *skorygowane podobieństwo consinusowe* (ang. *adjusted cosine similarity*).

2. *metody oparte o model* (ang. *model-based*) polegają na stworzeniu modelu, który będzie w stanie estymować preferencje użytkownika do danego produktu. Przykładami metod wykorzystywanych w tym celu są: *algorytm drzew decyzyjnych* (ang. *decision trees*), *reguły asocjacyjne, klasyfikatory bayesowskie* (ang. *Bayes classifiers*) oraz *modele czynników ukrytych* (ang. *latent factor models*). [11]

Zaletą systemów *collaborative filtering*, w porównaniu do systemów bazujących na cechach produktu, jest to, że rekomendacje są zdywersyfikowane. Ze względu na to, że metoda bazuje również na gustach innych użytkowników, rekomendacje mogą znacząco się różnić od produktów, które użytkownik ocenił dotychczas.

Problemem tych systemów jest jednak fakt, że w sytuacji gdy pojawi się nowy użytkownik nie są znane jego preferencje oraz historia zachowań, przez co system nie jest w stanie spersonalizować jego rekomendacji (*problem zimnego startu* dla użytkowników). Podobna sytuacja występuje dla nowych produktów, tzn. dopóki nowy produkt nie zostanie oceniony, nie będzie mógł zostać dla nikogo zarekomendowany (*problem zimnego startu* dla produktów).

2.5. Modele czynników ukrytych

Modele czynników ukrytych (ang. *latent factor models*) są metodami wykorzystywanymi w systemach *collaborative filtering* opartych o model. Ich głównym założeniem jest to, że użytkownicy oraz produkty zostają scharakteryzowani za pomocą skończonej liczby cech (czynników ukrytych) automatycznie wywnioskowanych ze zbioru preferencji użytkowników (zbioru ocen). Wnioskowanie owych cech opiera się na odkryciu pewnych związków pomiędzy zaobserwowanymi wartościami preferencji. Przykładowo, jeśli jako produkty opisane weźmiemy produkcje telewizyjne, odkrytymi czynnikami mogą być: przynależność do danego gatunku (dramat, komedia), preferowany wiek odbiorcy danej produkcji lub też inne cechy, których nie jesteśmy w stanie w prosty sposób zinterpretować.[12][13]

Jedną z najbardziej popularnych metod opartych o modele czynników ukrytych jest *faktoryzacja macierzy* (ang. *matrix factorization*), która jest chętnie wykorzystywana z uwagi na dobre rezultaty oraz skalowalność. Metoda faktoryzacji macierzy zakłada, że zbiór preferencji użytkowników można zdefiniować jako macierz R o rozmiarach $m \times n$. Wartość r_{ui} oznacza ocenę użytkownika u dla produktu i, która znajduję się na przecięciu u-tego wiersza i i-tej kolumny tej macierzy. Macierz R nie jest uzupełniona, tzn. istnieje wiele brakujących wartości preferencji z uwagi na to, że z reguły użytkownicy nie zostawiają ocen dla wszystkich dostępnych produktów.[12][14][4]

Głównym założeniem metody faktoryzacji macierzy w przypadku systemów rekomendacji jest to, że macierz R (o wymiarach $m \times n$) można przybliżyć poprzez iloczyn macierzy

P (o wymiarach $m \times k$) oraz Q (o wymiarach $n \times k$) [13][12]:

$$R_{m \times n} \approx P_{m \times k} Q^T_{n \times k} \tag{1}$$

, gdzie k odnosi się do liczby czynników ukrytych, za pomocą których zostaną opisani użytkownicy oraz produkty. Każdy użytkownik u odpowiada u-temu wierszowi ($p_u \in \mathbb{R}^k$) macierzy P. Natomiast każdy produkt i odpowiada i-temu wierszowi ($q_i \in \mathbb{R}^k$) macierzy Q. Wektor q_i opisuje jak otrzymane k cech ukrytych charakteryzuje produkt i (jakie cechy posiada dany produkt). Odpowiednio p_u opisuje jak te k cech charakteryzuje użytkownika u (jakimi cechami produktu użytkownik jest zainteresowany).

Następnie na podstawie uzyskanych macierzy P i Q mogą zostać wyestymowane brakujące wartości macierzy wejściowej R - wartości r_{ui} (brakująca preferencja użytkownika u dla produktu i). Przybliżenie wartości r_{ui} może zostać obliczone na podstawie iloczynu skalarnego wektorów p_u oraz q_i [12]:

$$\hat{r_{ui}} = q_i^T p_u \tag{2}$$

W celu odnalezienia macierzy P oraz Q spełniających równanie (1) wykorzystuje się metody oparte na *rozkładzie według wartości osobliwych* (ang. *singular value decomposition*, SVD). Jednak w przypadku systemów rekomendacji *collaborative filtering*, gdy w macierzy preferencji R istnieją brakujące wartości, zastosowanie klasycznych metod SVD nie jest możliwe [12]. Z tego względu znalezienie P i Q powinno bazować jedynie na dostępnych (zaobserwowanych) wartościach w macierzy R. Można to osiągnąć poprzez sformułowanie problemu optymalizacji bazującego jedynie na dostępnych wartościach r_{ui} [12][11]:

$$(P^*, Q^*) := \arg\min \sum_{(u,i) \in S} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{3}$$

, gdzie S to zbiór wszystkich par (u, i), dla których r_{ui} występuje w wejściowej macierzy R. Natomiast parametr λ wykorzystywany jest do regularyzacji, która próbuje rozwiązać problem nadmiernego dopasowania (ang. *overfitting*).

Do rozwiązania problemu minimalizacji z równania (3) często wykorzystuje się metodę *stochastycznego spadku wzdłuż gradientu* (ang. *stochastic gradient descent*, SGD). Drugim podejściem, które zostało użyte w opisywanych w tej pracy badaniach, jest algorytm *naprzemiennych najmniejszych kwadratów* (ang. *alternating least squares*, ALS).

2.5.1. Algorytm naprzemiennych najmniejszych kwadratów (ang. *alternating least squares*, ALS)

Algorytm *naprzemiennych najmniejszych kwadratów* (ang. *alternating least squares*, ALS)[15][16] jest metodą faktoryzacji macierzy, która jest wykorzystana do rozwiązania problemu optymalizacji opisanego w równaniu (3). Algorytm ALS podchodzi do tego w

sposób iteracyjny, dzieląc omawiany problem na mniejsze problemy o niższej złożoności. Przebieg algorytmu można opisać jako naprzemienne wywołanie dwóch poniższych kroków [11]:

1. Wartości macierzy P (wartości p_u) zostają uznane jako wartości stałe. Następnie dla każdego z n wierszy macierzy Q (każdego wektora q_i) rozwiązany zostaje problem optymalizacji:

$$Q^* := \arg\min \sum_{i:(u,i)\in S} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{4}$$

   , gdzie S to zbiór wszystkich par (u,i), dla których r_{ui} występuje w wejściowej macierzy R. Natomiast p_u traktowane są jako wartości stałe.
2. Wartości macierzy Q (wartości q_i) zostają uznane jako wartości stałe. Następnie dla każdego z m wierszy macierzy P (każdego wektora p_u) rozwiązany zostaje problem optymalizacji:

$$P^* := \arg\min \sum_{u:(u,i)\in S} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{5}$$

   , gdzie S to zbiór wszystkich par (u,i), dla których r_{ui} występuje w wejściowej macierzy R. Natomiast q_i traktowane są jako wartości stałe.

Powyższe kroki wykonywane są iteracyjne do osiągnięcia zbieżności. Poniżej przedstawiono omawiany algorytm również w postaci pseudokodu [17]:

Algorytm 1 Alternating Least Squares (ALS)

Inicjalizacja $q_i \leftarrow 0$
Inicjalizacja p_u *wartościami losowymi*
powtarzaj
  Ustaw p_u *jako wartości stałe, oblicz* q_i *przez minimalizację funkcji celu (4)*
  Ustaw q_i *jako wartości stałe, oblicz* p_u *przez minimalizację funkcji celu (5)*
dopóki *nie osiągnięto maksymalnej liczby iteracji*

Algorytm ALS zyskał dużą popularność i jest często stosowany w zastosowaniach systemów rekomendacji *collaborative filtering* z uwagi na to, że problemy optymalizacji dla każdego p_u (równanie 5) oraz dla każdego q_i (równanie 4) są niezależne. Z tego względu algorytm może zostać w prosty sposób zrównoleglony i przyspieszony poprzez obliczenia rozproszone [15].

2.5.2. Metoda faktoryzacji macierzy z wykorzystaniem ocen niejawnych

Tak jak opisano w rozdziale 2.2 systemy rekomendacji mogą działać na podstawie różnych typów ocen użytkowników. Pierwszym z nich są *oceny jawne*, tzn. takie w których użytkownik bezpośrednio wyraża opinię na temat danego produktu. Drugim typem są

oceny niejawne, w których preferencje użytkowników wnioskowane są na podstawie historycznych działań i interakcji z systemem.

Standardowe podejście do metody faktoryzacji macierzy w systemach rekomendacji, które zostało opisane w rozdziale 2.5, traktuje wartości macierzy preferencji jako oceny jawne. Jednak w wielu przypadkach (również w przypadku omawianej pracy) możliwe jest wyłącznie zastosowanie ocen niejawnych. Z tego względu autorzy [4] zaproponowali modyfikację metody faktoryzacji macierzy uwzględniającą typ ocen niejawnych.

W przypadku ocen niejawnych same oceny nie definiują bezpośrednich preferencji użytkowników do danych produktów, a opisują siłę (pewność) obserwacji wywnioskowanych na podstawie historycznych działań użytkowników (np. sumaryczny czas spędzony na oglądaniu danej serii programu telewizyjnego, serialu telewizyjnego). W związku z tym niskie wartości r_{ui} wyrażają małą pewność, że użytkownik u jest zainteresowany produktem i (użytkownik mógł obejrzeć dany program telewizyjny tylko ze względu na włączony w tle odbiornik). Natomiast wysokie wartości r_{ui} oznaczają dużą pewność obserwacji (cykliczne oglądanie danej serii audycji oznacza rzeczywiste zainteresowanie daną produkcją).

Autorzy [4] wprowadzają dwie dodatkowe wartości. Pierwszą z nich jest wartość d_{ui}:

$$d_{ui} = \begin{cases} 1 & \text{dla } r_{ui} \geq 0 \\ 0 & \text{dla } r_{ui} = 0 \end{cases} \tag{6}$$

Wartość binarna d_{ui} definiuje odnotowanie jakiejkolwiek interakcji pomiędzy użytkownikiem u, a produktem i. Drugą wartością jest c_{ui}, która opisuję siłę obserwacji d_{ui}:

$$c_{ui} = 1 + \alpha r_{ui} \tag{7}$$

Im wyższa jest wartość r_{ui}, tym szybszy jest wzrost siły obserwacji c_{ui}. Tempo tego wzrostu kontrolowane jest z wykorzystaniem wartości stałej α.

Następnie autorzy przedstawiają modyfikację problemu optymalizacji opisanego wzorem (3), który stosowany jest w standardowej metodzie faktoryzacji macierzy. Modyfikacja uwzględnia zmieniające się wartości siły obserwacji. Dodatkowo w przeciwieństwie do klasycznego przypadku operuje ona na wszystkich możliwych parach (u,i). Poniżej przedstawiono zmodyfikowany problem optymalizacji, który może służyć do faktoryzacji macierzy ocen R w przypadku, gdy wartości tej macierzy opisują oceny niejawne użytkowników [4]:

$$(P^*, Q^*) := \arg\min \sum_{u,i} c_{ui}(d_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{8}$$

Sformułowana w powyższy sposób metoda faktoryzacji macierzy (a w szczególności jej implementacja w silniku *Apache Spark* opisanego w rozdziale 3.2) została wykorzystana

do stworzenia analizowanego w opisanych w tej pracy badaniach systemu rekomendacji audycji telewizyjnych.

3. Wykorzystane biblioteki oraz narzędzia programistyczne

Badania zostały zrealizowane w języku *Python* w wersji *3.9*, który wykorzystano do przetworzenia danych wejściowych (opisanego w rozdziale 4.), stworzenia systemu rekomendacji (opisanego w rozdziale 5.) oraz oceny jego jakości. Dodatkowo poniżej opisano kluczowe biblioteki oraz narzędzia programistyczne wykorzystane w tej pracy.

3.1. PostgreSQL

PostgreSQL [18] jest jednym z najpopularniejszych systemów zarządzania relacyjnymi bazami danych. System został wykorzystany w badaniach ze względu na możliwość wykorzystania indeksu *GiST* opartego na algorytmie *generalized search tree*. Jest to rodzaj indeksu, który jest szczególnie przydatny do wyszukiwania w bardzo dużych bazach danych. W pracy wykorzystano jego zastosowanie do tworzenia optymalnych zapytań dotyczących przedziałów czasowych [19], co było kluczowym mechanizmem w procesie przetwarzania danych wejściowych (opisanych w rozdziale 4.).

3.2. Apache Spark

Apache Spark [20] jest to silnik analityczny wykorzystywany do przetwarzania dużej ilości danych. Jedną z głównych zalet tego rozwiązania jest możliwość równoległego wykonywania obliczeń.

Na potrzeby opisywanej pracy wykorzystano rozszerzenie silnika *Spark* w postaci biblioteki *MLlib* [21], która umożliwia zastosowanie go do celów rozwiązywania problemów uczenia maszynowego, a w szczególności dla tych badań - rozwiązania problemu rekomendacji z wykorzystaniem metody *collaborative filtering* opartej o model. Do tego celu biblioteka *MLlib* udostępnia implementację opisywanego w rozdziale 2.5.1 algorytmu *alternating least squares* (ALS) [22].

Na rysunku 3.1 przedstawiono przykład wykorzystania zawartej w bibliotece *MLlib* implementacji algorytmu ALS. Przykład opiera się na środowisku *PySpark* - interfejsie programistycznym silnika *Apache Spark* dla języka *Python*.

```
from pyspark.ml.recommendation import ALS

als = ALS(rank = 100
          maxIter = 5,
          regParam = 0.01,
          implicitPrefs = True,
          alpha = 10,
          userCol = "userId",
          itemCol = "movieId",
          ratingCol = "rating")
```

```
11
12 model = als.fit(training_df)
13
14 predictions = model.transform(test_df)
```

Rysunek 3.1. Przykład wykorzystania algorytmu ALS w środowisku *PySpark*.

Zmienne *training_df* oraz *test_df* definiują odpowiednio zbiór danych treningowych oraz zbiór danych testowych, które opisują dane w sposób tabelaryczny. Składają się one z krotek w postaci "*(userId, itemId, rating)*"dotyczących identyfikatora użytkownika, identyfikatora produktu oraz wartości oceny (w postaci jawnej lub niejawnej). W linii trzeciej zdefiniowane zostają parametry algorytmu ALS:

- parametr *rank* opisuje liczbę czynników ukrytych wykorzystanych przy faktoryzacji macierzy,
- parametr *maxIter* opisuje maksymalną liczbę iteracji algorytmu,
- parametr *regParam* opisuje wartość λ wykorzystywaną do regularyzacji ze wzorów (3) oraz (8),
- parametr *implicitPrefs* jest flagą, która służy do wyboru odpowiedniej wersji algorytmu *alternating least squares*. Wartość *False* oznacza wybór algorytmu z wykorzystaniem ocen jawnych (opisanego w rozdziale 2.5). Natomiast wartość *True* oznacza zastosowanie wersji algorytmu dla przypadku wykorzystania ocen niejawnych (opisanego w rozdziale 2.5.2),
- parametr *alpha* opisuje wartość stałą α wykorzystaną we wzorze (7), która określa tempo wzrostu siły obserwacji w przypadku ocen niejawnych,
- parametry *userCol*, *itemCol*, *ratingCol* oznaczają odpowiednio nazwy kolumn dla kolumn identyfikatorów użytkowników, identyfikatorów produktów, wartości ocen w zbiorach danych treningowych oraz testowych (*training_df* oraz *test_df*).

W linii dwunastej utworzony zostaje model systemu rekomendacji poprzez wywołanie algorytmu ALS na zbiorze danych treningowych. Następnie na podstawie uzyskanego modelu wyznaczone zostają przewidywane wartości ocen dla par *(userId, itemId)* ze zbioru danych testowych (linia czternasta).

4. Postać danych wejściowych

4.1. Pochodzenie danych

Dane użyte na potrzeby systemu rekomendacji opisywanego w tej pracy zostały przekazane do celów naukowych przez instytut badawczy NASK (Naukowa i Akademicka Sieć Komputerowa). Zbiór pochodzi od jednego z operatorów telewizji kablowej i dotyczy oglądalności polskich programów telewizyjnych przez abonentów. Dane zebrane zostały poprzez zagregowanie (w celu analizy danych potrzebnej do realizacji założonych celów bizensowych) zanonimizowanych informacji z dekoderów STB (ang. *set-top box*), tzn. elektronicznych urządzeń abonenckich, których zadaniem jest zamiana zewnętrznego źródła sygnału w możliwy do wyświetlenia na telewizorze obraz [23].

4.2. Postać danych

Omawiany zbiór danych na temat oglądalności telewizji udostępniony został w postaci plików tekstowych w formacie CSV (ang. *comma-separated values*). Każdy rekord w pliku składa się z czterech kolumn kolejno opisujących:

- unikalny, anonimowy identyfikator abonenta,
- datę podłączenia (norma *ISO8601* [24] z dokładnością do sekund) urządzenia abonenckiego do określonego programu telewizyjnego,
- datę odłączenia (norma *ISO8601* z dokładnością do sekund) urządzenia abonenckiego do określonego programu telewizyjnego,
- identyfikator programu telewizyjnego.

Przykład próbki danych przedstawiono w tabeli 4.1.

Tabela 4.1. Przykład rekordów na temat oglądalności (dla uproszczenia identyfikatory programów telewizyjnych zostały zastąpione ich nazwami).

Identyfikator abonenta	Data podłączenia	Data odłączenia	Program Telewizyjny
bt3riipx	*2021-09-15 18:29:35*	*2021-09-15 18:59:40*	Discovery Channel
2no6c6gm	*2021-09-15 18:59:02*	*2021-09-15 19:41:40*	Discovery Historia
bt3riipx	*2021-09-15 18:59:40*	*2021-09-15 20:39:47*	Animal Planet
i1z7toba	*2021-09-15 19:20:56*	*2021-09-15 19:59:44*	Eurosport 1

i1z7toba	*2021-09-15 19:59:44*	*2021-09-15 20:57:58*	Eurosport 2
njlwdja3	*2021-09-15 20:59:44*	*2021-09-15 21:37:22*	Discovery Science
bt3riipx	*2021-09-15 22:25:55*	*2021-09-15 23:15:44*	Discovery Channel
621h80tx	*2021-09-15 22:30:11*	*2021-09-15 23:10:32*	Discovery Channel

4.3. Statystyki zbioru danych

Wpisy zbioru danych pochodzą z przełomu września i października 2021 roku. Prezentują informacje na temat oglądalności 218 różnych programów telewizyjnych przez 50 tysięcy różnych użytkowników. Na rysunku 4.1 przedstawiono wykres liczby zgromadzonych rekordów dla poszczególnych dni, natomiast w całym zbiorze znajduję się ponad 95 milionów wpisów.

Rysunek 4.1. Wykres liczby zgromadzonych danych dla poszczególnych dni (sumarycznie 95683318 wpisów).

4.4. Wzbogacenie danych o informacje z bazy EPG

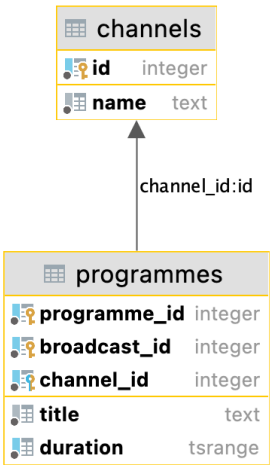
Niestety dane w takiej postaci nie pozwalają na analizę nawyków oglądania poszczególnych audycji telewizyjnych przez użytkowników, ponieważ zawierają jedynie informacje o preferencjach do konkretnych programów telewizyjnych. Z tego powodu zbiór danych został wzbogacony o dodatkowe informacje pochodzące z bazy danych EPG (ang. *Electro-*

nic Program Guide). EPG [25][26] to elektroniczny przewodnik po programach, nadawany w formie tekstowej, informujący o aktualnym programie ramowym w telewizji. Klasycznym zastosowaniem tego systemu jest możliwość szybkiego i efektywnego wyszukiwania z ekranu telewizora odpowiedniej audycji, która jest aktualnie emitowana lub będzie emitowana w najbliższej przyszłości.

Przykładowymi informacjami udostępnionymi przez EPG, poza standardową informacją o godzinach emisji, są:

- gatunek audycji (np. serial paradokumentalny, magazyn, serial komediowy, itp.),
- nazwiska aktorów w niej grających,
- czy kraj produkcji.

Na potrzeby opisywanego w tej pracy systemu rekomendacji, dane otrzymane poprzez EPG (z okresu pokrywającego zbiór danych wejściowych) zostały zaimportowane do relacyjnej bazy danych PostgreSQL, która umożliwia optymalne zapytania dotyczące audycji emitowanych na poszczególnych programach telewizyjnych w określonych przedziałach czasowych (przykład takiego zapytania widnieje na rysunku 4.3). Schemat wykorzystanej bazy został przedstawiony na rysunku 4.2.



Rysunek 4.2. Schemat bazy danych z informacjami z EPG.

```
1 select programmes.id, programmes.title, duration , channel_id
2 from programmes
3 inner join channel on channels.id = programmes.channel_id and
   ↪  channels.name = 'Discovery Channel'
4 where programmes.duration
5     && '[2021-09-15 14:30:00, 2010-09-15 14:35:00)'::tsrange
```

Rysunek 4.3. Przykład zapytania do bazy danych dotyczącego emitowanych audycji.

Surowe dane na temat oglądalności programów telewizyjnych zostają uzupełnione o dodatkowe informacje na temat emitowanych audycji. Proces ten można przedstawić w następujących krokach:

1. Każdy wpis w pliku zawierający dane o podłączeniu i odłączeniu od danego programu, zostaje rozszerzony (za pomocą zapytania do bazy EPG) do listy audycji nadawanych w tej sesji oglądania. Dla przykładu wpis przedstawiony w tabeli 4.2 zostaje przekształcony do postaci przedstawionej w tabeli 4.3.

Tabela 4.2. Przykład rekordu danych przed rozszerzeniem do listy audycji.

Identyfikator abonenta	Data podłączenia	Data odłączenia	Program Telewizyjny
xt3rci2x	*2021-09-15 18:19:35*	*2021-09-15 20:39:40*	Discovery Channel

Tabela 4.3. Przykład listy rekordów przed zagregowaniem sesji oglądania dotyczących jednej audycji.

Identyfikator abonenta	Rozpoczęcie sesji	Zakończenie sesji	Identyfikator audycji	Program Telewizyjny
xt3rci2x	*2021-09-15 18:29:35*	*2021-09-15 19:00:00*	1081977 (Kamperem przez świat)	Discovery Channel
xt3rci2x	*2021-09-15 19:00:00*	*2021-09-15 20:00:00*	1451785 (Polscy truckersi)	Discovery Channel
xt3rci2x	*2021-09-15 20:00:00*	*2021-09-15 20:39:40*	1441285 (Złomowisko PL)	Discovery Channel

2. Zostaje wyliczony czas (w sekundach) oglądania emitowanej audycji przez użytkownika w danej sesji oglądania.
3. Następnie wszystkie wpisy są zgrupowane według użytkowników i audycji oraz zagregowane do sumarycznego czasu ich oglądania we wszystkich sesjach oglądania. Dla zobrazowania, lista rekordów z tabeli 4.4 zostaje zagregowana do formy opisanej w tabeli 4.5.

Tabela 4.4. Przykład listy rekordów przed zagregowaniem sesji dotyczących jednej audycji.

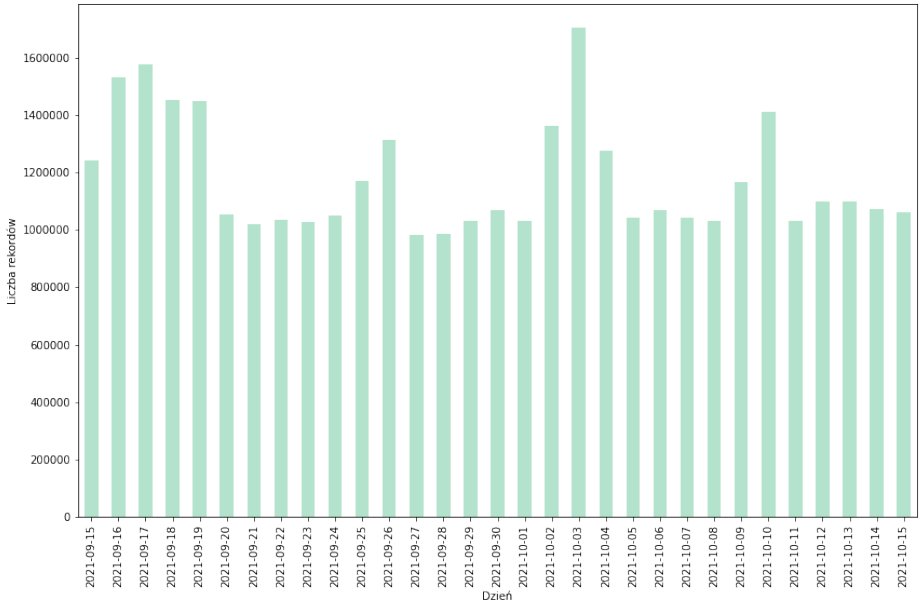
Identyfikator abonenta	Długość sesji	Identyfikator audycji	Program Telewizyjny
i1z7toba	1810s	1972865 (W sieci eksperymentów)	Discovery Science
i1z7toba	3540s	1264227 (Polscy truckersi)	Discovery Channel
i1z7toba	1245s	1972865 (W sieci eksperymentów)	Discovery Science

Tabela 4.5. Przykład listy rekordów po zagregowaniu sesji dotyczących jednej audycji.

Identyfikator abonenta	Długość okna	Identyfikator audycji	Program Telewizyjny
i1z7toba	3055s	1972865 (W sieci eksperymentów)	Discovery Science
i1z7toba	3540s	1264227 (Polscy truckersi)	Discovery Channel

4. Do każdego otrzymanego wpisu dodana została informacja o:
    - całkowitej długości danej audycji,
    - czasie między rozpoczęciem danej audycji, a rozpoczęciem oglądania audycji przez użytkownika,
    - czasie między zakończeniem oglądania danej audycji przez użytkownika, a zakończeniem danej audycji,
    - liczbie przełączeń użytkownika na inny program podczas emisji audycji.

Finalnie po tym całym procesie, całkowita wielkość zbioru danych to ponad 36 milionów rekordów. Na rysunku 4.4 przedstawiono wykres liczby zgromadzonych rekordów dla poszczególnych dni.



Rysunek 4.4. Wykres liczby zgromadzonych danych dla poszczególnych dni po procesie wzbogacania o informacje z bazy EPG (sumarycznie 36480846 wpisów).

4.5. Dyskusja

Kluczowym aspektem w tworzeniu systemu rekomendacji audycji telewizyjnych jest informacja o aktualnym programie ramowym na poszczególnych programach telewizyjnych. W początkowym etapie badań skorzystano z ogólnodostępnej, darmowej bazy danych EPG[27]. Niestety to rozwiązanie nie dostarczało na tyle dokładnych danych, aby mogły być wykorzystane w opisywanej pracy. W danych można było odnaleźć wiele przypadków błędów oraz sprzeczności, takich jak: brak ciągłości w informacjach o audycjach emitowanych na poszczególnych programach telewizyjnych (istniały przedziały czasowe, w których brakowało informacji o aktualnej emisji), czy też brak zgodności w emitowanej w danym czasie audycji (w bazie widniały sprzeczne informacje o kilku różnych audycjach nadawanych w tym samym czasie na jednym programie telewizyjnym). Powyższe problemy nie pozwalały na wykorzystanie tego dostawcy danych EPG w opisywanych badaniach. Ostatecznie zastosowano dokładniejsze dane przekazane za pośrednictwem instytutu badawczego NASK, które pozbawione były przypadków takich błędów.

5. Architektura systemu rekomendacji

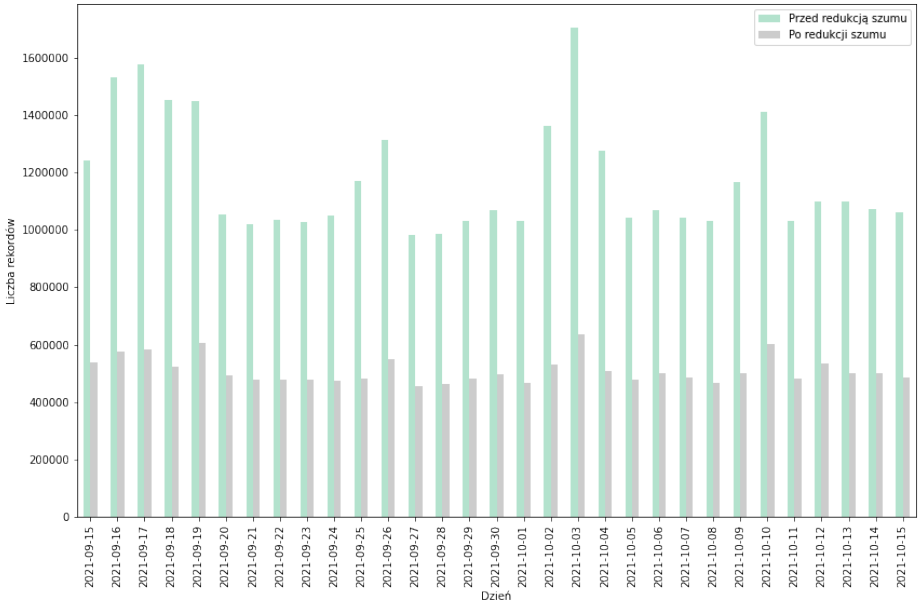
5.1. Redukcja szumu w zbiorze danych

Z uwagi na fakt, że zbiór danych wejściowych to rzeczywiste dane z urządzeń abonenckich, zawiera on w sobie wiele szumu, który może negatywnie wpływać na jakość rekomendacji. Przykładowo można zaobserwować wiele przypadków, w których jeden program telewizyjny był oglądany przez użytkownika przez wiele godzin. Można, więc przypuszczać, że odbiorca nie jest zainteresowany emitowanymi w tym czasie audycjami, a odbiornik telewizyjny pozostawał po prostu włączony w tle.

W opisywanych badaniach zastosowano dwie strategie redukcji szumu:

1. odfiltrowane zostały wszystkie rekordy dotyczące audycji oglądanych w bardzo krótkich sesjach (przyjęte zostały 2 minuty), ponieważ nie niesie to za sobą informacji o zainteresowaniu daną audycją, a może oznaczać tzw. *zapping* [28], czyli sposób oglądania polegający na ciągłym przełączeniu programów telewizyjnych (np. spowodowanym brakiem wyboru konkretnej pozycji programowej lub niechęcią odbiorcy do bloków reklamowych),
2. jeśli sesja, w której użytkownik oglądał określony program telewizyjny jest bardzo długa (przyjęte zostały 4 godziny), brana pod uwagę jest jedynie pierwsza audycja z czasu trwania tej sesji.

Na rysunku 5.1 przedstawiono wielkość zbioru danych przed i po procesie redukcji szumu.



Rysunek 5.1. Wykres liczby zgromadzonych danych dla poszczególnych dni przed i po procesie redukcji szumu.

5.2. Metody wyliczania ocen niejawnych

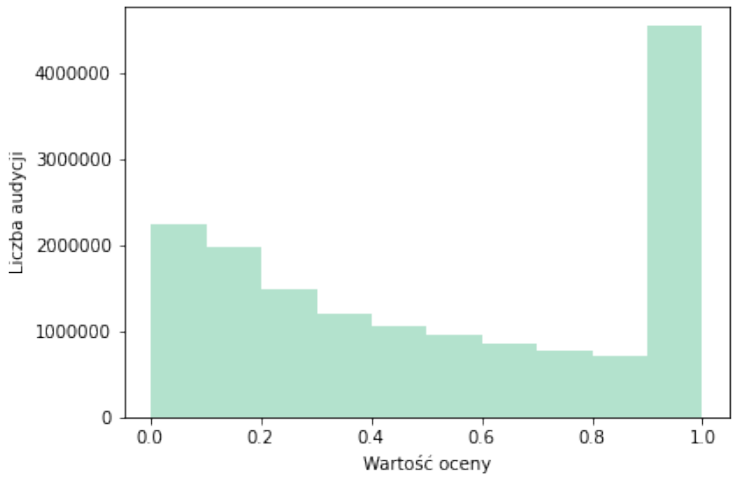
Tak jak opisano w rozdziale 2. systemy rekomendacji mogą korzystać z różnych typów danych wprowadzanych na wejściu, opisujących preferencje użytkowników do poszczególnych pozycji. Najczęściej wykorzystywanym typem są *oceny jawne* (ang. *explicit rating*), czyli takie, w których użytkownik wprost ocenia nastawienie do danego produktu. Jednak nie zawsze (np. przez limitacje systemu) takie informacje są łatwe do zebrania, również w przypadku odbiorców telewizji kablowej. W takim przypadku system rekomendacji może opierać się na tzw. *ocenach niejawnych* (ang. *implicit rating*), czyli ocenach pozyskiwanych poprzez obserwacje zachowań użytkowników.

W opisywanych w tej pracy badaniach porównano trzy metody wyliczania *ocen niejawnych* abonentów dla pojedynczej audycji telewizyjnej:

- *wariant pierwszy* uwzględnia czas spędzony na oglądaniu i opiera się na założeniu, że im większa jest część w jakiej użytkownik obejrzał daną audycję, tym większa jest jego preferencja do tej audycji [4]:

$$r_{ui} = \frac{t_{ui}}{T_i}, \tag{9}$$

gdzie t_{ui} to sumaryczny czas, który użytkownik u poświęcił na oglądanie audycji i. Natomiast T_i to całkowita długość audycji i. Dla przykładu wartość $r_{ui} = 0.6$ oznacza, że użytkownik u obejrzał audycje i w 60%. Na rysunku 5.2 przedstawiono rozkład liczby audycji w zależności od wartości uzyskanych ocen *wariantu pierwszego*.



Rysunek 5.2. Rozkład liczby audycji w zależności od oceny niejawnej w *wariancie pierwszym*.

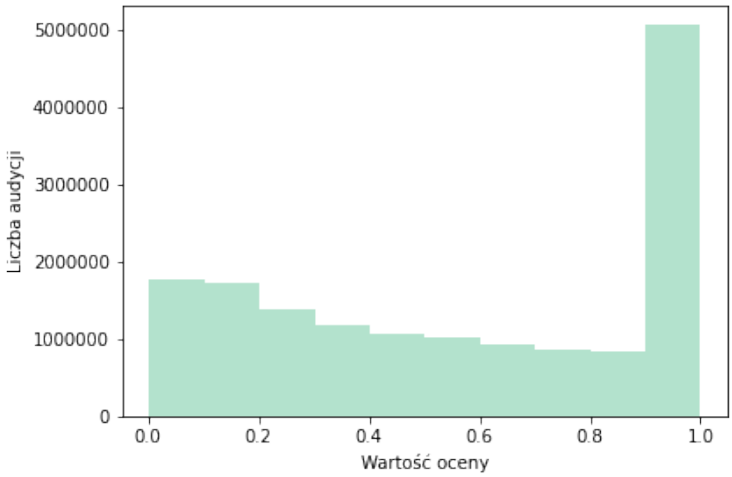
Jak można zauważyć rozkład jest nierównomierny. Największa liczba audycji (ponad cztery miliony) została obejrzana w ponad 90% ($r_{ui} \geq 0.9$), co może być związane z tendencją odbiorców do oglądania jednego programu telewizyjnego w długich sesjach (oglądanie wielu audycji na danym programie "pod rząd"). Można również

zaobserwować wiele audycji, które zostały obejrzane w niewielkim stopniu ($r_{ui} \leq 0.4$)). Prawdopodobnie obrazuje to sytuacje, w których użytkownicy intensywnie przełączają programy telewizyjne przy poszukiwaniu interesujących ich audycji.

- *wariant drugi* uwzględnia to, jaką część audycji użytkownik pominął od jej rozpoczęcia oraz jaką część audycji użytkownik ominął na jej zakończenie. Wariant zakłada, że jeśli użytkownik rozpoczął oglądanie audycji po jej starcie lub przełączył program telewizyjny przed jej zakończeniem, to w takim wypadku jego ocena danej audycji będzie mniejsza:

$$r'_{ui} = 1 - \frac{t^s_{ui} + t^e_{ui}}{T_i}, \tag{10}$$

gdzie t^s_{ui} to czas pomiędzy rozpoczęciem nadawania audycji i, a rozpoczęciem oglądania audycji i przez użytkownika u, a t^e_{ui} to czas pomiędzy zakończeniem oglądania audycji i przez użytkownika u, a zakończeniem nadawania danej audycji. Przykładowo, jeśli abonent obejrzał audycję w całości to r'_{ui} będzie równe 1, natomiast jeśli rozpoczął oglądanie w połowie to wartość r'_{ui} wyniesie 0.5. Na rysunku 5.2 przedstawiono rozkład liczby audycji w zależności od wartości uzyskanych ocen *wariantu drugiego*.



Rysunek 5.3. Rozkład liczby audycji w zależności od oceny niejawnej w *wariancie drugim*.

Można zaobserwować, że przedstawiony rozkład zbliżony jest do rozkładu z przypadku zastosowania *wariantu pierwszego*. Występuje jednak widoczny wzrost liczby ocen z zakresu $[0.9, 1.0]$ oraz spadek wśród niskich wartości. Prawdopodobnie jest to następstwem tego, że ocena nie uwzględnia części audycji, którą użytkownik pominął podczas jej trwania, co powoduje występowanie skrajnie wysokich wartości r'_{ui}.

- *wariant trzeci* jest to rozszerzenie *wariantu pierwszego* z uwzględnieniem kary za nadmiarową liczbę przełączeń programu telewizyjnego podczas nadawania danej audycji, gdyż duża liczba przełączeń może oznaczać mniejsze zainteresowanie.

$$r''_{ui} = r_{ui} \frac{e^{-0.7z_{ui}+3}}{1+e^{-0.7z_{ui}+3}}, \tag{11}$$

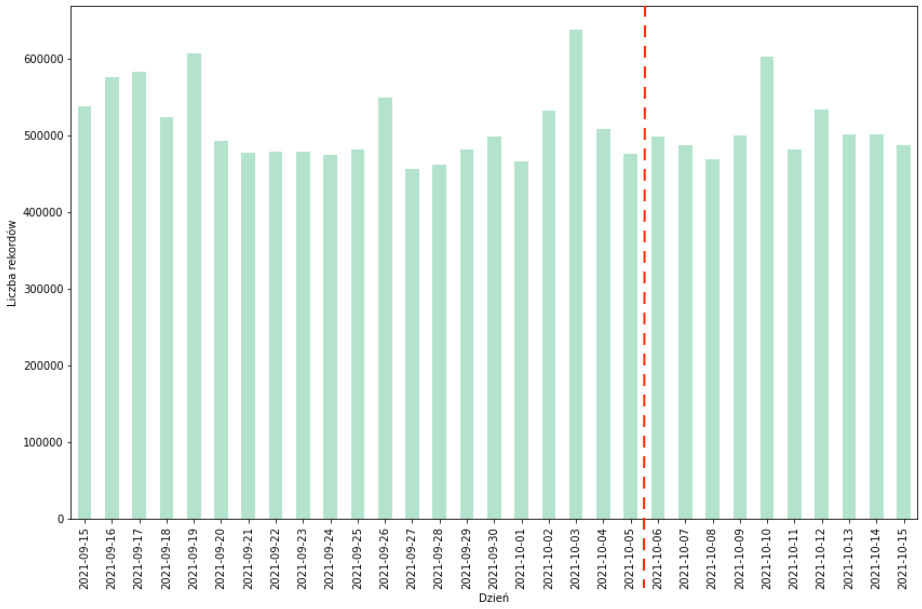
gdzie z_{ui} określa średnią liczbę przełączeń programu na godzinę przez użytkownika u podczas nadawania audycji i. Funkcja kary została dobrana na podstawie analizy wartości z_{ui} występujących w zbiorze danych. Zaobserwowano, że znaczna ich część zawiera się w przedziale od 0 do 10 przełączeń programu na godzinę. Funkcja kary stopniowo obniża ocenę r_{ui} wraz ze wzrostem średniej liczby przełączeń z_{ui} w tym przedziale.

5.3. Podział danych na zbiór danych treningowy oraz zbiór danych testowy

Zbiór danych opisywany w poprzednim rozdziale został podzielony na:

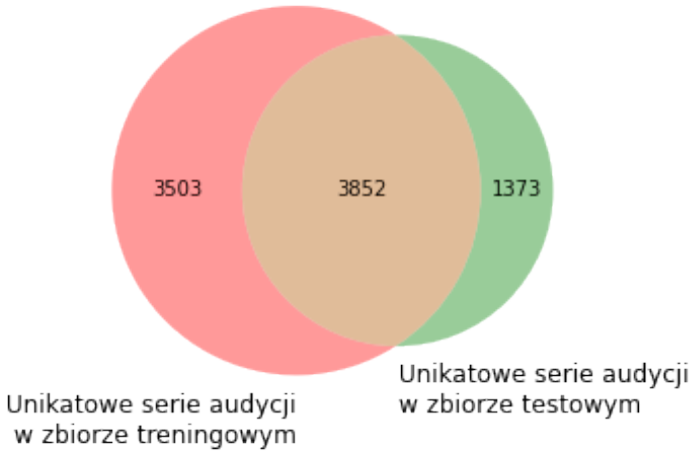
- *zbiór danych treningowych* - danych, które posłużą do stworzenia modelu systemu rekomendacji,
- *zbiór danych testowych* - danych, które posłużą do oceny jakości tego modelu.

Zaproponowanym podziałem jest podział na *trzy tygodnie* (21 dni) opisujące zbiór danych treningowych oraz *pozostały tydzień* (dokładnie 10 dni) na zbiór danych testowych. Podział ten przedstawiono na rysunku 5.4.



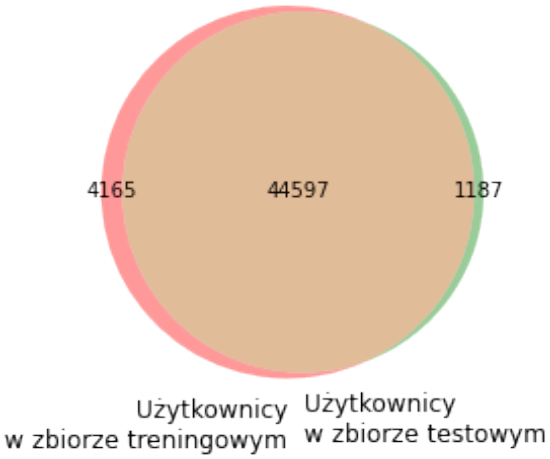
Rysunek 5.4. Wykres liczby zgromadzonych danych dla poszczególnych dni z uwzględnieniem podziału na zbiór treningowy i testowy (linia czerwona).

Dodatkowo na rysunku 5.5 zobrazowano relacje pomiędzy zbiorami unikatowych audycji w danych treningowych i danych testowych. Można zauważyć, że aż około 15% serii audycji jest unikatowe dla zbioru testowego (nowe seriale, filmy, które nie zostały zarejestrowane w zbiorze treningowym). Jednak z uwagi na to, że w badaniach została wykorzystana metoda *collaborative filtering*, która opiera się na historii zachowań odbiorców (przez co system nie jest w stanie zarekomendować pozycji, które nie pojawiają się w danych treningowych), badania skupiają się wyłącznie na części wspólnej przedstawionych zbiorów audycji.



Rysunek 5.5. Wykres relacji pomiędzy zbiorem unikatowych serii audycji w danych treningowych oraz zbiorem unikatowych serii audycji w danych testowych.

Analogicznie na rysunku 5.6 przedstawiono relacje pomiędzy zbiorami unikatowych użytkowników w danych treningowych oraz testowych. W tym przypadku można zaobserwować, że tylko niewielka część (około 2%) użytkowników nie występuję w zbiorze danych treningowych. Jednak ze względu na fakt, że wykorzystany system rekomendacji nie jest w stanie wspomagać użytkowników bez znajomości ich historycznych preferencji, w badaniach brani są pod uwagę jedynie użytkownicy występujący na przecięciu zbioru treningowego oraz testowego.



Rysunek 5.6. Wykres relacji pomiędzy zbiorem unikatowych użytkowników w danych treningowych oraz zbiorem unikatowych użytkowników w danych testowych.

5.4. Stworzenie modelu systemu rekomendacji

W celu stworzenia modelu systemu rekomendacji, wartości uzyskanych ocen niejawnych w zbiorze danych treningowych zostają, dla każdego użytkownika, zsumowane względem serii audycji, której dotyczą. Innymi słowami, dla każdego użytkownika wszystkie oceny audycji dotyczące tej samej serii audycji zostają zagregowane do postaci sumy. Jeśli użytkownik cyklicznie powraca do audycji z pewnej serii (np. cyklicznie ogląda kolejne odcinki serialu), suma ta będzie większa. Oznacza to większą pewność, że użytkownik znowu obejrzy audycję z tej serii. Natomiast niskie wartości zsumowanych ocen audycji opisywałyby serie, które użytkownik ogląda sporadycznie.

Uzyskane sumaryczne wartości ocen niejawnych wykorzystane są do zbudowania modelu systemu rekomendacji z wykorzystaniem wspomnianego w rozdziale 3.2 silnika *Apache Spark*. Model opiera się na opisanej w rozdziale 2.5 metodzie faktoryzacji macierzy oraz algorytmie *alternating least squares* (ALS).

5.5. Miary jakości rekomendacji

Głównymi stosowanymi w praktyce sposobami na ocenę jakości systemów rekomendacji są *testowanie online* oraz *testowanie offline*. Testowanie online bazuje na zebranej informacji zwrotnej na temat prezentowanych rekomendacji, a co za tym idzie użytkownik jest kluczowym elementem działania tej metody. Jednak taka interakcja z użytkownikiem często nie jest możliwa, na przykład tak jak w przypadku tej pracy - w projektach badawczych. Z tego powodu najczęściej wykorzystywaną metodą zmierzenia jakości rekomendacji jest testowanie offline. [29][30]

W testowaniu offline ocena działania systemów rekomendacji opiera się na zebranym zbiorze danych (zbiorze danych testowych) opisującym historyczne preferencje (oceny) użytkowników. Wykorzystanie takiego zbioru próbuje symulować realne interakcje użytkownika z systemem i opiera się na założeniu, że jego zachowanie będzie podobne w sytuacji, gdy taki system rekomendacji zostanie już docelowo wdrożony.

Najczęściej opisywaną w literaturze miarą jakości systemów rekomendacji jest dokładność (ang. *accuracy*), tego jak bardzo przewidywane dla produktów oceny zbliżone są do rzeczywiście odnotowanych w zbiorze danych testowych ocen. W tym celu wykorzystywane są takie metryki jak [30]:

- *średni błąd bezwzględny* (ang. *mean absolute error*, MAE),
- czy *pierwiastek błędu średniokwadratowego* (ang. *root-mean-square error*, RMSE).

Jednak w wielu przypadkach funkcjonowanie systemu rekomendacji nie opiera się wprost na estymowaniu tego, jak użytkownik oceni dany produkt. Mogą opierać się na zaprezentowaniu listy rankingowej rekomendacji uszeregowanych według preferencji użytkownika. W takich przypadkach nie jest ważne to, z jaką dokładnością system przewiduje konkretne wartości ocen, a interesuje nas to czy użytkownik skorzysta z produktów przedstawionych w liście rankingowej. Rezultat pojedynczej rekomendacji można opisać jako jedną z czterech wartości przedstawionych w tabeli 5.1.

Tabela 5.1. Klasyfikacja możliwych wartości rezultatów pojedynczej rekomendacji.

	Produkt zarekomendowany	Produkt niezarekomendowany
Produkt powinien być zarekomendowany	wartość prawdziwie pozytywna (ang. *true-positive*, **TP**)	wartość fałszywie negatywna (ang. *false-negative, **FN**)*
Produkt nie powinien być zarekomendowany	wartość fałszywie pozytywna (ang. *false-positive*, **FP**)	wartość prawdziwie pozytywna (ang. *true-nagative*, **TN**)

Następnie z wykorzystaniem informacji na temat rezultatów rekomendacji zawartych w liście rankingowej można obliczyć metryki jakości w postaci:

- *precyzji* (ang. *precision*), która określa jaki jest stosunek liczby rekomendacji trafionych względem liczby wszystkich rekomendacji,

$$precision = \frac{\|TP\|}{\|TP\| + \|FP\|}$$

- *skuteczność* (ang. *recall*), która określa stosunek liczby rekomendacji trafionych względem liczby produktów, które powinny być zarekomendowane.

$$recall = \frac{\|TP\|}{\|TP\| + \|FN\|}$$

Często również wykorzystuje się wariacje powyższych metryk w postaci $precision@k$ oraz $recall@k$, których działanie jest tożsame z odpowiednikami. Dotyczą one jednak podzbioru k pierwszych propozycji z listy rankingowej rekomendacji.

5.5.1. Zaproponowane miary jakości

Przedstawione w tej pracy badania wykorzystują system rekomendacji oparty na ocenach niejawnych. Jak zauważono w [4] i [31] oceny niejawne nie dostarczają *de facto* informacji o nastawieniu użytkownika do danej audycji. Sytuacja, w której użytkownik nie obejrzał danej audycji nie musi oznaczać, że nie jest nią zainteresowany (nie lubi jej), mógł na przykład być zajęty w czasie jej emisji. Wykorzystane oceny niejawne można natomiast interpretować jako pewien stopień pewności (ang. *confidence*) tego, że użytkownik obejrzy audycję z konkretnej serii (np. następny odcinek serialu) ponownie. Z tego powodu zastosowanie w tym przypadku metryk opartych na dokładności (ang. *accuracy*) predykcji ocen (na przykład metryk MAE lub RMSE) mogłoby prowadzić do niejednoznacznych wyników.

Dodatkowo za pośrednictwem ocen niejawnych uzyskanych ze zbioru danych na temat oglądalności nie mamy tak na prawdę informacji o tym, które audycje nie są przez użytkowników preferowane (których użytkownik nie chce oglądać). Z tego względu miary jakości w oparciu o precyzję (ang. *precision-based metrics*) nie zostały wykorzystane, gdyż wymagają znajomości audycji, które nie powinny być zarekomendowane (wartości *false-positives* oraz *true-negatives*). Wykorzystano zatem miary jakości oparte o skuteczność (ang. *recall-based metrics*), które takich informacji nie potrzebują. [4]

W opisywanych badaniach został zaprezentowany scenariusz, który bazuje na całym programie ramowych telewizji z okresu pokrywającego dane ze zbioru testowego, tzn. model tworzy rekomendacje na podstawie wszystkich programów emitowanych w telewizji w tym okresie. Posłużono się dwoma sposobami na ocenę jakości rekomendacji:

- Pierwszy z tych sposobów opiera się na metryce *mean percentage ranking*, MPR zaproponowanej w [31][4]. Jako $rank_{ui}$ oznaczymy ranking percentylowy (ang. *percentile-ranking*) audycji i w uszeregowanej liście rankingowej zaprezentowanej użytkownikowi u. Dla przykładu $rank_{ui} = 0\%$ oznaczałoby, że audycja i została wyznaczona jako najbardziej odpowiednia (najbardziej pewna) dla użytkownika u. Natomiast $rank_{ui} = 100\%$ oznaczałoby, że audycja i jest dla niego najmniej odpowiednia (została umieszczona na końcu listy rankingowej). Na tej podstawie metrykę *MPR* można przedstawić jako średni percentyl tych audycji, które zostały trafnie zarekomendowane użytkownikowi (wartości *true-positives*, TP) i można wyrazić ją

wzorem:

$$MPR = \frac{\sum_{u,i} d_{ui} \times rank_{ui}}{\sum_{u,i} d_{ui}} \quad , \tag{12}$$

gdzie zmienna d_{ui} przyjmuje wartości:

$$d_{ui} = \begin{cases} 1 & \text{dla } r_{ui} > 0 \\ 0 & \text{dla } r_{ui} = 0 \end{cases}$$

- Druga metoda została zaproponowana odpowiadając na konkretne potrzeby problemu opisywanego w tej pracy, tzn. problemu rekomendacji audycji telewizyjnych. Metoda próbuje w pewien sposób imitować zachowanie użytkownika (oczywiście w pewnym uproszczeniu), w którym użytkownik o pewnej porze dnia włącza odbiornik telewizyjny i otrzymuje listę rekomendowanych na następne kilka godzin audycji. Następnie z tej listy użytkownik wybiera te, które mu odpowiadają. Działanie tej metody można opisać w następujących krokach:
  1. Każdy dzień z okresu pokrywającego zbiór danych testowych został podzielony na siedem okien odpowiadających porom dnia. Podział ten został zaprezentowany w tabeli 5.2,

Tabela 5.2. Zaproponowany podział na okna odpowiadające porom dnia.

Okno czasowe	Pora dnia
6:00 - 9:00	poranek (ang. *morning*)
9:00 - 12:00	późny poranek (ang. *late morning*)
12:00 - 15:00	popołudnie (ang. *afternoon*)
15:00 - 18:00	późne popołudnie (ang. *late afternoon*)
18:00 - 21:00	wieczór (ang. *evening*)
21:00 - 24:00	późny wieczór (ang. *late evening*)
24:00 - 6:00	noc (ang. *night*)

  2. następnie każda rekomendacja z zaprezentowanej dla użytkownika listy rankingowej audycji zostaje przypisana do odpowiadającej jej pory dnia,

3. rezultaty w takiej postaci zostają pogrupowane kolejno według dnia i okna określającego porę dnia. Dla k najlepszych rekomendacji w każdej grupie wyliczona zostaje opisana w tym rozdziale miara $recall@k$ oraz dodatkowa miara $hit@k$, która określa, czy w danym oknie udało się poprawnie zarekomendować przynajmniej jedną audycję:

$$hit_{ui} = \begin{cases} 1 & \text{dla} \quad \|TP\| > 0 \\ 0 & \text{dla} \quad \|TP\| = 0 \end{cases}$$

W badaniach przyjęto $k = 10$, tzn. wymienione wyżej metryki biorą pod uwagę dziesięć najlepszych rekomendacji dla danego okna. W tabeli 5.3 przedstawiono przykładowy rezultat tej operacji,

Tabela 5.3. Przykład zastosowania metryk $recall@k$ oraz $hit@k$ w oknach określających porę dnia. W przykładzie przyjęto $k = 3$.

Użytkownik	Dzień	Pora dnia	Audycje rzeczywiście obejrzane	Audycje zarekomendowane	$recall@k$	$hit@k$
i1z7toba	07.10.2020	wieczór	[101, 109]	[**101**, 104, 108]	0.5	1
i1z7toba	07.10.2020	późny wieczór	[120]	[119, **120**, 111]	1	1
i1z7toba	08.10.2020	poranek	[201, 204]	[211, 209, 205]	0	0
i1z7toba	08.10.2020	popołudnie	[220, 223, 228]	[**220**, **223**, 230]	0.66	1

- finalnie uzyskane wartości $recall@k$ oraz $hit@k$ z każdego[2] okna określającego porę dnia w od każdego użytkownika u zostają zagregowane do postaci średnich:

$$\overline{recall@k} = \frac{\sum_{u,w}(recall@k)_{uw}}{\sum_u W_u} \tag{13}$$

$$\overline{hit@k} = \frac{\sum_{u,w}(hit@k)_{uw}}{\sum_u W_u} \tag{14}$$

Wartość W_u oznacza liczbę okien, w których użytkownik u obejrzał przynajmniej jedną audycje.

[2] Oczywiście brane pod uwagę są jedynie te okna określające porę dnia, w których użytkownik obejrzał przynajmniej jedną audycję.

6. Uzyskane rezultaty rekomendacji

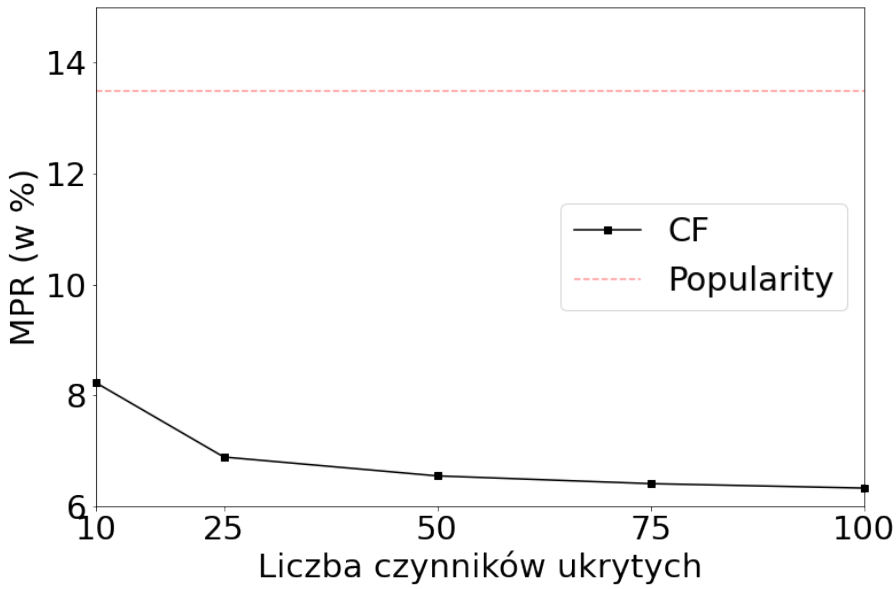
6.1. Wartość odniesienia rezultatów

Dla celów porównawczych zaimplementowano również prosty, niespersonalizowany system rekomendacji bazujący na statystykach popularności (ang. *popularity-based recommender system*). System ten tworzy listę rekomendacji na podstawie popularności audycji. Na początku takiej listy umiejscowiona jest audycja najczęściej oglądana wśród użytkowników. Natomiast na jej końcu znajduję się audycja, która oglądana jest najrzadziej.

Wartości miar MPR, $\overline{recall@10}$ i $\overline{hit@10}$ uzyskane z wykorzystaniem powyższego systemu rekomendacji posłużą jako wartość odniesienia dla rezultatów otrzymanych przez proponowany w opisywanych badaniach system *collaborative filtering*.

6.2. Rezultaty dla *wariantu pierwszego* ocen niejawnych (uwzględniającego czas spędzony na oglądaniu audycji)

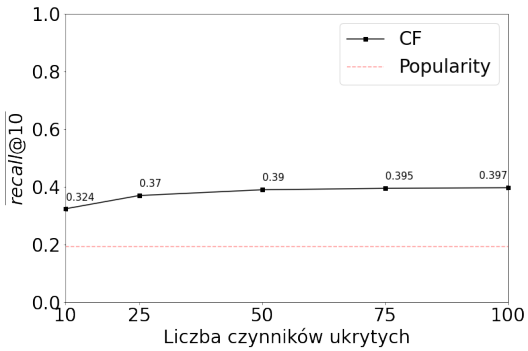
Na rysunku 6.1 przedstawiono uzyskane wartości metryki MPR dla modelu rekomendacji wykorzystującego oceny niejawne w *wariancie pierwszym*, opisanym wzorem (9). Wartości porównano względem liczby czynników ukrytych wykorzystanych w trenowaniu modelu systemu rekomendacji *collaborative filtering* (linia czarna). Dodatkowo oznaczono rezultat osiągnięty przez system bazujący na popularności (linia czerwona) jako punkt odniesienia.



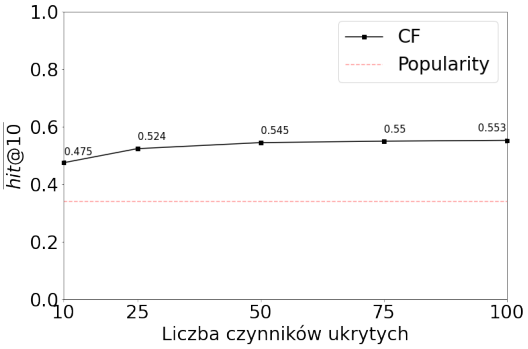
Rysunek 6.1. Uzyskane wartości miary MPR dla modelu opartego na ocenach niejawnych w *wariancie pierwszym*.

System rekomendacji bazujący na popularności uzyskał wynik $MPR = 13.5\%$. Jednak taki system nie jest spersonalizowany pod konkretnego użytkownika i proponuje wszystkim jednakowe rekomendacje, traktując każdego w ten sam sposób. Jak można zauważyć system *collaborative filtering*, personalizując rekomendacje, potrafi znacząco poprawić ten rezultat. Wartości MPR dla tego systemu spadają wraz z liczbą czynników ukrytych i osiągają najlepszy wynik dla $MPR = 6.33\%$.

Na rysunkach 6.2 i 6.3 przedstawiono natomiast rezultaty dla metryk $\overline{recall@10}$ oraz $\overline{hit@10}$.



Rysunek 6.2. Uzyskane wartości miary $\overline{recall@10}$ dla modelu opartego na ocenach niejawnych w *wariancie pierwszym*.

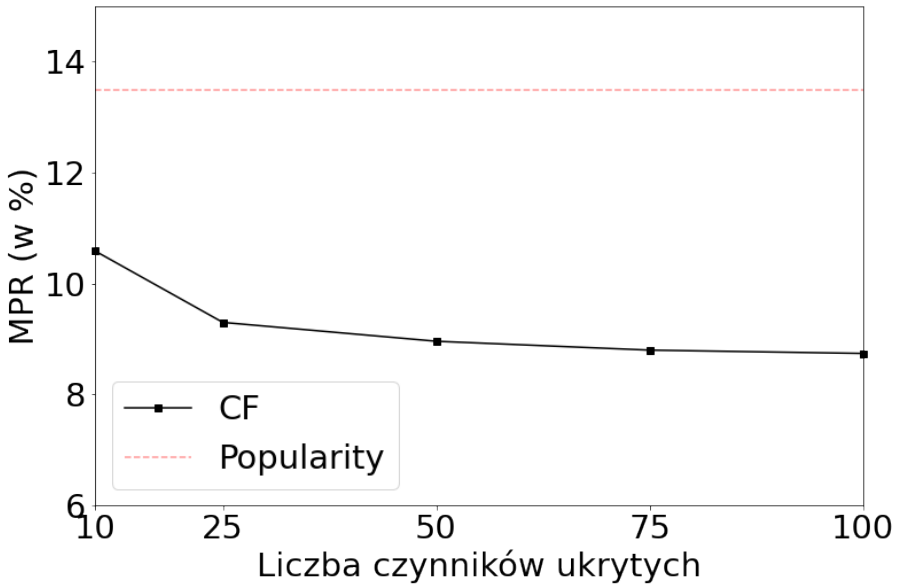


Rysunek 6.3. Uzyskane wartości miary $\overline{hit@10}$ dla modelu opartego na ocenach niejawnych w *wariancie pierwszym*.

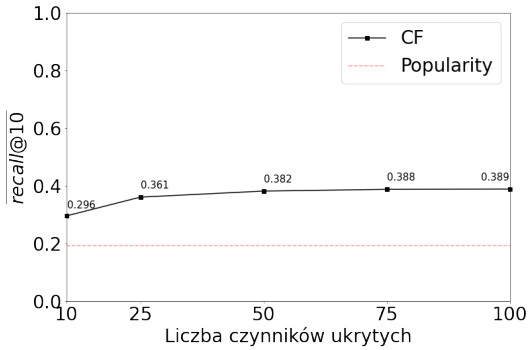
Można zaobserwować, że obie te metryki również rosną wraz z liczbą zastosowanych czynników ukrytych. Maksymalnie $\overline{recall@10}$ osiąga wartość 0.4, co oznacza, że średnio we wszystkich oknach określających porę dnia udaję się zarekomendować 40% audycji, które zostały rzeczywiście obejrzane. Wynik ten można porównać do systemu *popularity-based*, gdzie wartość ta jest prawie dwukrotnie mniejsza. Natomiast miara $\overline{hit@10}$ dochodzi do poziomu 0.55. To znaczy, że w 55% okien zaproponowano przynajmniej jedną trafioną rekomendację.

6.3. Rezultaty dla *wariantu drugiego* ocen niejawnych (uwzględniającego czas pominięty na początku oraz na końcu audycji)

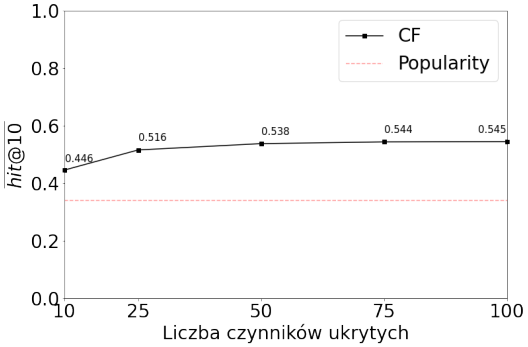
Uzyskane rezultaty dla modelu systemu rekomendacji wykorzystującego *wariant drugi* ocen niejawnych (opisanego wzorem (10)) zaprezentowano na rysunkach 6.4, 6.5 oraz 6.6. Można zaobserwować, że podobnie jak poprzednio, najlepsze wartości uzyskano dla modelu z największą liczbą czynników ukrytych. W najlepszym przypadku średnio w każdym oknie udaje się zarekomendować 39% obejrzanych audycji (wartość $\overline{recall@10} = 0.39$). Dodatkowo, w 54% okien określających porę dnia zasugerowano przynajmniej jedną trafioną audycję (wartość $\overline{hit@10} = 0.54$). Natomiast metryka MPR maksymalnie osiąga wartość $MPR = 8.74\%$



Rysunek 6.4. Uzyskane wartości miary MPR dla modelu opartego na ocenach niejawnych w *wariancie drugim.*



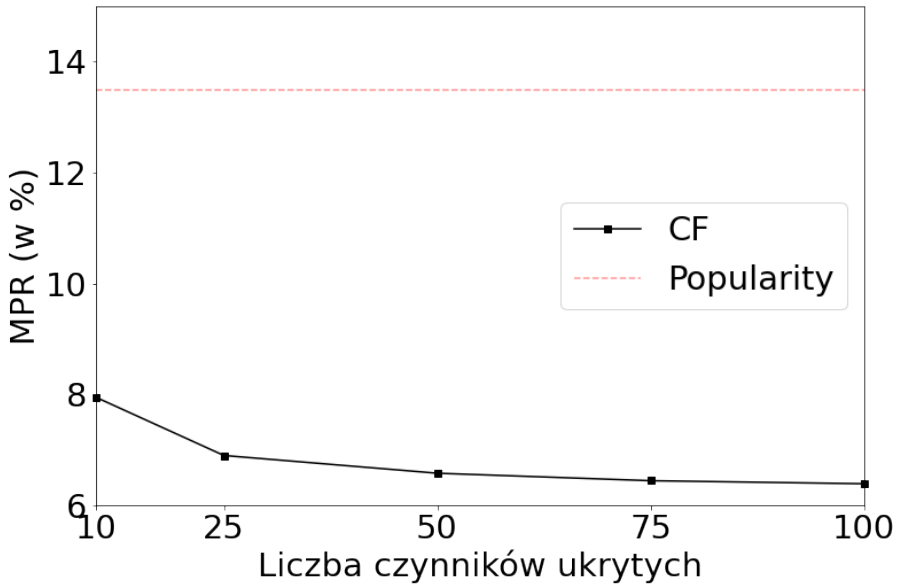
Rysunek 6.5. Uzyskane wartości miary $\overline{recall@10}$ dla modelu opartego na ocenach niejawnych w *wariancie drugim.*



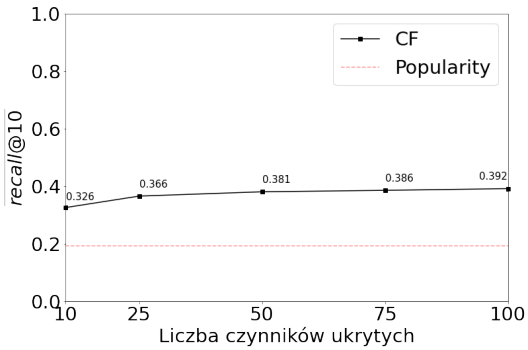
Rysunek 6.6. Uzyskane wartości miary $\overline{hit@10}$ dla modelu opartego na ocenach niejawnych w *wariancie drugim.*

6.4. Rezultaty dla *wariantu trzeciego* ocen niejawnych (uwzględniającego liczbę przełączeń programu podczas emisji audycji)

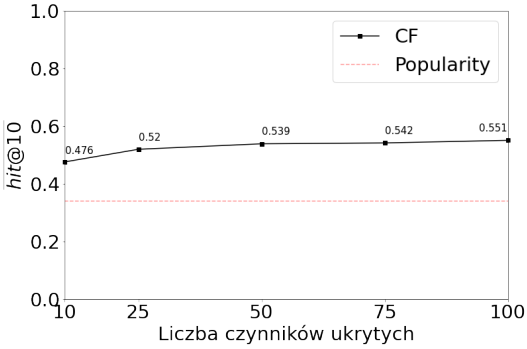
Uzyskane rezultaty dla modelu systemu rekomendacji wykorzystującego *wariant trzeci* ocen niejawnych (opisanego wzorem (11)) zaprezentowano na rysunkach 6.7, 6.8 oraz 6.6. W tym przypadku również najlepsze rezultaty uzyskano z wykorzystaniem największej liczby czynników ukrytych. W najlepszym przypadku średnio w każdym oknie udaje się zarekomendować 39% obejrzanych audycji (wartość $\overline{recall@10} = 0.39$). Dodatkowo, w 55% okien określających porę dnia zasugerowano przynajmniej jedną trafioną audycję (wartość $\overline{hit@10} = 0.55$). Metryka MPR maksymalnie osiąga wartość $MPR = 6.39\%$.



Rysunek 6.7. Uzyskane wartości miary MPR dla modelu opartego na ocenach niejawnych w *wariancie trzecim.*



Rysunek 6.8. Uzyskane wartości miary $\overline{recall@10}$ dla modelu opartego na ocenach niejawnych w *wariancie trzecim.*



Rysunek 6.9. Uzyskane wartości miary $\overline{hit@10}$ dla modelu opartego na ocenach niejawnych w *wariancie trzecim.*

6.5. Dyskusja

Jak można zauważyć najlepsze rezultaty osiągnięto z wykorzystaniem podstawowego wariantu ocen niejawnych (*wariantu pierwszego* opisanego wzorem (9)). Niestety uwzględnienie kary za nadmiarową liczbę przełączeń programu telewizyjnego podczas nadawania danej audycji (*wariant trzeci* opisany wzorem (11)) nie przyniosło poprawy i wyniki są bardzo zbliżone do tych z *wariantu pierwszego*. Najgorsze rezultaty uzyskano dla *wariantu drugiego* - uwzględniającego czas pominięty na początku oraz na końcu audycji.

Niestety, z uwagi na to, że jakość opisywanego systemu rekomendacji badana jest na podstawie rzeczywistych danych na temat oglądalności, istnieją potencjalne czynniki

zewnętrzne, które mogą wpływać na jakość uzyskanych wyników. Prezentowane badania opierają się na *testowaniu offline* opartym na zebranym zbiorze danych testowych opisującym historyczne preferencje odbiorców. Z tego względu rekomendacje, które nie były trafne mogą wynikać z tego, że:

- w tym samym czasie na innym programie telewizyjnym nadawana była audycja, która bardziej interesowała użytkownika,
- odbiorca nie był świadomy emisji danej audycji,
- czy też sytuacje, w których ktoś inny (na przykład inny członek rodziny) aktualnie korzysta z odbiornika i rekomendacje nie były kierowane do niego.

Takie czynniki nie mogą być w prosty sposób zaadresowane podczas oceny jakości systemu rekomendacji, a co za tym idzie mogą powodować gorsze rezultaty.

7. Podsumowanie

Celem badań omawianych w niniejszej pracy była weryfikacja działania oraz użyteczności systemu rekomendacji audycji telewizyjnych, który docelowo pomógłby odbiorcy telewizji w podjęciu decyzji o tym, co mógłby obejrzeć. Wykorzystane dane wejściowe przekazane zostały do celów badawczych przez instytut badawczy NASK. Są to rzeczywiste *dane telemetryczne* pochodzące z zamontowanych w gospodarstwach domowych urządzeń abonenckich. Opisują zarejestrowane informacje o identyfikatorze aktualnie odbieranego programu telewizyjnego oraz czasie podłączenia i odłączenia użytkownika do tego programu. Dane zostały dodatkowo wzbogacone o informacje na temat emitowanych audycji z wykorzystaniem bazy danych EPG.

W celu stworzenia omawianego systemu rekomendacji zastosowano metodę *collaborative filtering* z wykorzystaniem modelu czynników ukrytych. Model czynników ukrytych bazował na technice faktoryzacji macierzy, a dokładnie na algorytmie naprzemiennych najmniejszych kwadratów (ALS).

Do opisania preferencji odbiorców do poszczególnych audycji wykorzystano oceny niejawne, czyli oceny pozyskane poprzez obserwacje zachowań użytkowników. W badaniach porównano trzy warianty wyliczania takich ocen. Pierwszy z nich uwzględnia czas spędzony na oglądaniu danej audycji przez użytkownika. Drugi bierze pod uwagę czas jaki użytkownik pominął na początku oraz na końcu emisji. Natomiast trzeci ma na uwadze to, ile razy użytkownik przełączył program telewizyjny podczas nadawania danej audycji.

Jednym z wyzwań napotkanych podczas badań był wybór sposobu oceny jakości systemu rekomendacji. Zaproponowano miarę MPR, która opisuje średni percentyl trafnie zarekomendowanych audycji na uszeregowanej liście rankingowej wszystkich zaproponowanych użytkownikowi audycji. Przedstawiono również metodę odpowiadającą na konkretne potrzeby problemu rekomendacji audycji telewizyjnych. Metoda próbuje w pewien sposób imitować zachowanie użytkownika, w którym użytkownik o pewnej porze dnia włącza odbiornik telewizyjny i otrzymuje listę zaproponowanych na następne kilka godzin audycji. Dla każdej pory dnia zostaje zbadana jakość rekomendacji za pomocą miar $\overline{recall@k}$ oraz $\overline{hit@k}$.

Dalszym kierunkiem rozwoju badań może być stworzenie hybrydowego systemu rekomendacji poprzez zastosowanie opisanej w rozdziale 2.4.1 metody *content-based filtering*. Z uwagi na to, że przy użyciu jedynie metody *collaborative filtering*, w sytuacji gdy w ramówce pojawi się nowa seria audycji (na przykład nowy serial lub nowy film), audycja z tej serii nie będzie mogła zostać dla nikogo zarekomendowana (*problem zimnego startu*). Wynika to z tego, że metoda *collaborative filtering* opiera się jedynie na historii zachowań użytkowników w systemie. Metoda *content-based filtering* mogłaby rozwiązać ten problem poprzez tworzenie rekomendacji na podstawie podobieństwa (podobieństwo mogłoby być określone na przykład na podstawie programu telewizyjnego, gatunku produkcji, czy też pory emisji) nowych treści do audycji obejrzanych w przeszłości.

Bibliografia

[1] F. Ricci, L. Rokach i B. Shapira, "Recommender Systems: Introduction and Challenges", w sty. 2015, s. 1–5, ISBN: 978-1-4899-7636-9. DOI: 10.1007/978-1-4899-7637-6_1.

[2] F. Ricci, L. Rokach i B. Shapira, "Recommender Systems: Introduction and Challenges", w sty. 2015, s. 8–9, ISBN: 978-1-4899-7636-9. DOI: 10.1007/978-1-4899-7637-6_1.

[3] D. Oard i J. Kim, "Implicit Feedback for Recommender System", *Proceedings of the AAAI Workshop on Recommender Systems*, lip. 2000.

[4] Y. Hu, Y. Koren i C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets", grud. 2008, s. 263–272. DOI: 10.1109/ICDM.2008.22.

[5] C. C. Aggarwal, "An Introduction to Recommender Systems", w *Recommender Systems: The Textbook*. Cham: Springer International Publishing, 2016, s. 1–3, ISBN: 978-3-319-29659-3. DOI: 10.1007/978-3-319-29659-3_1. adr.: https://doi.org/10.1007/978-3-319-29659-3_1.

[6] C. Desrosiers i G. Karypis, "A Comprehensive Survey of Neighborhood-Based Recommendation Methods", w sty. 2011, s. 41–42. DOI: 10.1007/978-0-387-85820-3_4.

[7] M. de Gemmis, P. Lops, C. Musto, F. Narducci i G. Semeraro, "Semantics-Aware Content-Based Recommender Systems", w sty. 2015, s. 119–159, ISBN: 978-1-4899-7636-9. DOI: 10.1007/978-1-4899-7637-6_4.

[8] C. Aggarwal, "Content-Based Recommender Systems", w mar. 2016, s. 139–166, ISBN: 978-3-319-29657-9. DOI: 10.1007/978-3-319-29659-3_4.

[9] C. C. Aggarwal, "Neighborhood-Based Collaborative Filtering", w *Recommender Systems: The Textbook*. Cham: Springer International Publishing, 2016, s. 29–70, ISBN: 978-3-319-29659-3. DOI: 10.1007/978-3-319-29659-3_2. adr.: https://doi.org/10.1007/978-3-319-29659-3_2.

[10] X. Ning, C. Desrosiers i G. Karypis, "A Comprehensive Survey of Neighborhood-Based Recommendation Methods", w *Recommender Systems Handbook*, F. Ricci, L. Rokach i B. Shapira, red. Boston, MA: Springer US, 2015, s. 37–76, ISBN: 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_2. adr.: https://doi.org/10.1007/978-1-4899-7637-6_2.

[11] C. C. Aggarwal, "Model-Based Collaborative Filtering", w *Recommender Systems: The Textbook*. Cham: Springer International Publishing, 2016, s. 71–138. DOI: 10.1007/978-3-319-29659-3_3. adr.: https://doi.org/10.1007/978-3-319-29659-3_3.

[12] Y. Koren, R. Bell i C. Volinsky, "Matrix Factorization Techniques for Recommender Systems", *Computer*, t. 42, nr. 8, s. 30–37, 2009. DOI: 10.1109/MC.2009.263.

[13] C. C. Aggarwal, "Model-Based Collaborative Filtering", w *Recommender Systems: The Textbook*. Cham: Springer International Publishing, 2016, s. 90–106. DOI: 10.

1007/978-3-319-29659-3_3. adr.: https://doi.org/10.1007/978-3-319-29659-3_3.

[14] Y. Koren i R. Bell, "Advances in Collaborative Filtering", w *Recommender Systems Handbook*, F. Ricci, L. Rokach i B. Shapira, red. Boston, MA: Springer US, 2015, s. 77–118, ISBN: 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_3. adr.: https://doi.org/10.1007/978-1-4899-7637-6_3.

[15] C. C. Aggarwal, "Model-Based Collaborative Filtering", w *Recommender Systems: The Textbook*. Cham: Springer International Publishing, 2016, s. 105–106. DOI: 10.1007/978-3-319-29659-3_3. adr.: https://doi.org/10.1007/978-3-319-29659-3_3.

[16] Y. Zhou, D. Wilkinson, R. Schreiber i R. Pan, "Large-Scale Parallel Collaborative Filtering for the Netflix Prize", w *Algorithmic Aspects in Information and Management*, R. Fleischer i J. Xu, red., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, s. 337–348, ISBN: 978-3-540-68880-8.

[17] S. Gosh, N. Nahar, M. A. Wahab, M. Biswas, M. S. Hossain i K. Andersson, "Recommendation System for E-commerce Using Alternating Least Squares (ALS) on Apache Spark", w *Intelligent Computing and Optimization*, P. Vasant, I. Zelinka i G.-W. Weber, red., Cham: Springer International Publishing, 2021, s. 880–893, ISBN: 978-3-030-68154-8.

[18] "PostgreSQL", Dostęp zdalny (31.01.2023): www.postgresql.org.

[19] "PostgreSQL Range Types", Dostęp zdalny (31.01.2023): www.postgresql.org/docs/current/rangetypes.html.

[20] "Apache Spark", Dostęp zdalny (31.01.2023): www.spark.apache.org/docs/latest/index.html.

[21] "Apache Spark MLlib", Dostęp zdalny (31.01.2023): www.spark.apache.org/docs/latest/ml-guide.html.

[22] "Apache Spark MLlib Collaborative Filtering", Dostęp zdalny (31.01.2023): www.spark.apache.org/docs/latest/ml-collaborative-filtering.html.

[23] "Set-top box", Dostęp zdalny (20.08.2022): en.wikipedia.org/wiki/Set-top_box.

[24] "ISO8601", Dostęp zdalny (25.08.2022): www.iso.org/iso-8601-date-and-time-format.html.

[25] "Electronic Program Guide", Dostęp zdalny (25.08.2022): pl.wikipedia.org/wiki/Electronic_Program_Guide.

[26] "ETSI EN 300 707", Dostęp zdalny (25.08.2022): www.etsi.org/deliver/etsi_en/300700_300799/300707/01.02.01_40/en_300707v010201o.pdf.

[27] "Darmowy dostawca EPG", Dostęp zdalny (28.01.2023): epg.ovh.

[28] "Zapping", Dostęp zdalny (26.08.2022): pl.wikipedia.org/wiki/Zapping.

[29] C. C. Aggarwal, "Evaluating Recommender Systems", w *Recommender Systems: The Textbook*. Cham: Springer International Publishing, 2016, s. 225–254, ISBN:

978-3-319-29659-3. DOI: 10.1007/978-3-319-29659-3_7. adr.: https://doi.org/10.1007/978-3-319-29659-3_7.

[30] G. Shani i A. Gunawardana, "Evaluating Recommendation Systems", w sty. 2011, t. 12, s. 257–297. DOI: 10.1007/978-0-387-85820-3_8.

[31] Y. Li, J. Hu, C. Zhai i Y. Chen, "Improving One-Class Collaborative Filtering by Incorporating Rich User Information", sty. 2010, s. 959–968. DOI: 10.1145/1871437.1871559.

Spis rysunków

Spis tabel

Spis załączników