Spis treści

[1 Wstęp 3](#_Toc18873069)

[1.1 Typy systemów rekomendacyjnych 3](#_Toc18873070)

[1.1.1 Filtracja zbiorowa 3](#_Toc18873071)

[1.1.2 Filtracja na podstawie treści 4](#_Toc18873072)

[1.1.3 Systemy hybrydowe 5](#_Toc18873073)

[1.2 Cel i motywacja pracy 5](#_Toc18873074)

[1.3 Organizacja pracy 6](#_Toc18873075)

[2 Systemy rekomendacyjne 7](#_Toc18873076)

[2.1 Metody filtracji zbiorowej 7](#_Toc18873077)

[2.1.1 Użyte oznaczenia 7](#_Toc18873078)

[2.1.2 Metody najbliższego sąsiedztwa 8](#_Toc18873079)

[2.1.3 Metryki podobieństwa 9](#_Toc18873080)

[2.1.4 Zalety oraz problemy metod najbliższego sąsiedztwa 12](#_Toc18873081)

[2.1.5 Metody rozkładu macierzy 14](#_Toc18873082)

[2.1.6 Oszacowanie bazowe 15](#_Toc18873083)

[2.1.7 Simon Funk SVD 16](#_Toc18873084)

[2.1.8 ALS 16](#_Toc18873085)

[2.1.9 Zalety i wady algorytmów rozkładu macierzy 17](#_Toc18873086)

[2.2 Systemy oparte na treści 18](#_Toc18873087)

[2.3 Systemy hybrydowe 18](#_Toc18873088)

[2.3.1 Systemy ważone 19](#_Toc18873089)

[2.3.2 Systemy przełączane 19](#_Toc18873090)

[2.3.3 Systemy mieszające 19](#_Toc18873091)

[2.3.4 Systemy kaskadowe 19](#_Toc18873092)

[2.3.5 Systemy łączące źródła 19](#_Toc18873093)

[2.3.6 Systemy oparte na poszerzeniu danych wejściowych 19](#_Toc18873094)

[2.4 Sposób oceniania algorytmów 20](#_Toc18873095)

[2.4.1 Metryki online 20](#_Toc18873096)

[2.4.2 Metryki offline 20](#_Toc18873097)

[3 Algorytm ALS z wykorzystaniem treści 24](#_Toc18873098)

[3.1 Hybrydowy rozkład macierzy 24](#_Toc18873099)

[3.2 Użyte narzędzia 26](#_Toc18873100)

[3.2.1 Anaconda 26](#_Toc18873101)

[3.2.2 NumPy i Pandas 26](#_Toc18873102)

[3.2.3 Numba 26](#_Toc18873103)

[3.2.4 HyperOpt 27](#_Toc18873104)

[3.2.5 Surprise 27](#_Toc18873105)

[3.3 Uwagi implementacyjne 27](#_Toc18873106)

[3.4 Zbiory danych 29](#_Toc18873107)

[3.4.1 MovieLens 29](#_Toc18873108)

[3.4.2 Yelp 30](#_Toc18873109)

[3.5 Testy algorytmu hybrydowego 34](#_Toc18873110)

[3.5.1 Procedura eksperymentalna 34](#_Toc18873111)

[3.5.2 Uzyskane wyniki – przypadek ciepłego startu 36](#_Toc18873112)

[3.5.3 Wpływ cech jawnych na rekomendację 37](#_Toc18873113)

[3.5.4 Uzyskane wyniki – przypadek zimnego startu 40](#_Toc18873114)

[4 Podsumowanie 42](#_Toc18873115)

[5 Bibliografia 43](#_Toc18873116)

[6 Wykaz symboli i skrótów 45](#_Toc18873117)

[6.1 Symbole 45](#_Toc18873118)

[6.2 Skróty 46](#_Toc18873119)

[7 Spis rysunków 47](#_Toc18873120)

[8 Spis tabel 48](#_Toc18873121)

# Wstęp

Wraz z rozwojem internetu, konsumenci otrzymali większą możliwość wyboru niż kiedykolwiek wcześniej. Sklepy internetowe prezentują asortyment bez ograniczenia jakim była przestrzeń magazynowa, natomiast dostawcy treści udostępniają na swoich stronach nieprzebrane zasoby multimediów oraz informacji. Taka możliwość wyboru niesie za sobą jednak pewne konsekwencje. Dokonanie wyboru przy zbyt wielu opcjach przekracza możliwości przeciętnego użytkownika. Z tego też względu, strony rozszerzane są o funkcje, takie jak katalogi i wyszukiwarki, ułatwiające znalezienie poszukiwanych zasobów. Alternatywą do tych rozwiązań są systemy rekomendacyjne. Opierają się one na wiedzy na temat użytkowników i produktów oraz aktywności użytkownika. W wypadku tej ostatniej, jej śledzenie stało się o wiele prostsze niż jeszcze kilka lat temu ze względu na ilość udostępnianej informacji. Praktycznie każda akcja: odwiedzone strony, czas przeglądania, wpisywane zapytania czy nawet dokładna lokalizacja są zbierane i analizowane. Pozwala to na osiągnięcie większego zadowolenia oraz zainteresowania użytkownika, co ostatecznie ma wpływ na rentowność świadczonych usług. Z tego też względu firmy takie jak Amazon, Netflix czy Google wykorzystują systemy rekomendacyjne jako jeden z kluczowych elementów udostępnianych przez siebie serwisów [1].

## Typy systemów rekomendacyjnych

Badania nad systemami rekomendacyjnymi doprowadziły do powstania wielu, niezależnych podejść do problemu. Zależnie od typu informacji, z jakich korzystają systemy rekomendacyjne, mówi się o różnych ich typach [1]:

* systemy filtracji zbiorowej („collaborative filtering”),
* systemy oparte na treści („content-based filtering”),
* systemy hybrydowe.

Każdy z typów posiada swoje zalety i wady, a próba ich jak najlepszego zrównoważenia jest celem kolejnych prac badawczych. Dodatkowo, istnieją też systemy wykorzystujące inne źródła wiedzy, takie jak kontekst decyzji konsumenckiej czy sieci społecznościowe, jednak nie są one tematem tej pracy i zostały w niej pominięte.

### Filtracja zbiorowa

Algorytmy filtracji zbiorowej wykorzystują informację o aktywności lub opiniach użytkowników i wynikających z nich zależności. Prostym przykładem ich użycia może być sytuacja przedstawiona na rys. 1. Na podstawie opinii wszystkich użytkowników przewidywane jest zdanie ostatniego z nich na temat spaghetti. Predykcja ta może zostać podjęta ze względu na pełną zgodność w opiniach zaznaczonych użytkowników.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Hot dog | Taco | Pasta | Whole pizza |
| User | Thumbs up sign | Thumbs up sign | Thumbs up sign | Thumbs up sign |
| User |  | Thumbs up sign | Thumbs up sign | Thumbs up sign |
| User | Thumbs up sign | Thumbs up sign | Thumbs up sign |  |
| User | Thumbs up sign |  | Thumbs up sign |  |
| User | Thumbs up sign | Thumbs up sign | Question mark | Thumbs up sign |

Rys. 1 Preferencje użytkowników

Analogiczne podejście stosowane jest w wypadku ocen, wyrażonych w szerszej skali niż lubię/nie lubię. Przykładem tego mogą być „gwiazdki”, wystawiane filmom czy restauracjom, odpowiadające skali od 1 do 5. Wartości te można przedstawić jako macierz ocen, analogiczną do tej z rys. 1, w której wiersze odpowiadają ocenom użytkowników a kolumny ocenom obiektów.

Systemy filtracji zbiorowej nie sprawdzają się jednak w przypadku nowych użytkowników   
i obiektów. W wypadku braku lub szczątkowej informacji na temat przeszłej aktywności, przewidywania na temat upodobań użytkownika lub popularności produktu są obarczone dużą niepewnością. Przykładem tego może być sytuacja przedstawiona na rys. 2. Nowy użytkownik nie ocenił jeszcze żadnego z dań. Z tego względu nie można ustalić czy jego opinie są zbliżone do użytkownika który lubi spaghetti czy do drugiego z nich, o przeciwnej opinii. Przewidzenie opinii nowego użytkownika na temat spaghetti jest w takim wypadku niemożliwe.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Hot dog | Taco | Pasta | Whole pizza |
| User |  | Thumbs up sign | Thumbs up sign | Thumbs up sign |
| User | Thumbs up sign |  | Thumbs up sign |  |
| User |  |  | Question mark |  |

Rys. 2 Problem nowego użytkownika

### Filtracja na podstawie treści

Drugim z podejść jest analiza zbioru cech obiektów (rzadziej użytkowników). Przykładem takiego zbioru mogą być informacje na temat rekomendowanych filmów: gatunek, reżyser, obsada i tematyka, których wycinek został przedstawiony w tabeli 1.

Tab. 1 Zbiór cech wybranych filmów

|  |  |
| --- | --- |
| **Tytuł** | **Słowa kluczowe** |
| Skazany na Shawshank | kryminał, dramat, frank\_darabont, tim\_robbins, morgan\_fr… |
| Ojciec Chrzestny | kryminał, dramat, francis\_ford\_coppola, marlon\_brando, a… |
| Ojciec Chrzestny II | kryminał, dramat, francis\_ford\_coppola, al\_pacino, robert… |
| Mroczny Rycerz | akcja, kryminał, dramat, christopher\_nolan, christian\_… |
| 12 Gniewnych Ludzi | kryminał, dramat, sidney\_lumet, martin\_balsam, john\_fied… |
| … | … |

W momencie, gdy użytkownik korzysta z systemu, obiekty, którymi był zainteresowany, włączane są w jego profil. Na tej podstawie badane jest zainteresowanie już nie konkretnymi obiektami a cechami, którymi się one charakteryzują. Przy założeniu, że użytkownikowi spodobają się przedmioty o podobnej charakterystyce, pozwala to na przygotowanie rekomendacji. Jednocześnie, wybór ten może zostać uzasadniony użytkownikowi poprzez odwołanie do zrozumiałej dla niego informacji dotyczącej tego, jakimi cechami, rekomendowanego obiektu kierował się algorytm.

|  |  |  |  |
| --- | --- | --- | --- |
| User | Lubi | Music notes | - rock - lata 70-te |
| Rekomendowane | |  | Podobne obiekty |
| Music notation | - rock - lata 70-te |

Rys. 3 Rekomendacja oparta na treści

Główne założenie systemów opartych na analizie treści jest jednak również ich wadą. Mimo że możliwe jest zarekomendowanie nawet nowo dodanego obiektu, na podstawie pozytywnie odbieranych przez użytkownika cech, prowadzi to do zmniejszenia zróżnicowania rekomendowanych obiektów, poprzez nacisk na ten sam, niewielki zbiór cech.

### Systemy hybrydowe

Ostatnim z omawianych podejść są systemy hybrydowe. W odróżnieniu od wymienionych   
w dwóch poprzednich paragrafach, nie stanowią one samodzielnego bytu. Zgodnie ze swoją nazwą łączą one więcej niż jedno rozwiązanie w spójny proces. Definicja ta nie określa jednak ani użytych algorytmów ani sposobu ich łączenia. Pomimo to, stosowanie ich motywowane jest tym samym celem; próbą wykorzystania zalet różnych podejść przy jednoczesnym ograniczeniu związanych z nimi wad.

## Cel i motywacja pracy

Większość bibliotek rekomendacyjnych oferuje ten sam zestaw algorytmów. Zazwyczaj działają one w oparciu o pojedyncze źródło danych, takie jak przeszła aktywność użytkownika czy charakterystyka rekomendowanych obiektów. O ile jest to rozwiązanie proste, nie wykorzystuje ono pełnego potencjału wynikającego z dostępnych informacji.   
Z drugiej strony, algorytmy takie jak opisany w [2], zwycięzca konkursu Netflix, uzyskują dużo lepsze wyniki poprzez szczegółowe modelowanie zachowania użytkowników i ich preferencji. Charakteryzują się jednak wysokim poziomem skomplikowania, utrudniającym ich komercyjne użycie. Próba znalezienia rozwiązania pośredniego między tymi dwoma podejściami stała się motywacją do napisania tej pracy. Jej głównym założeniem jest stworzenie algorytmu będącego kompromisem między wykorzystaniem dostępnych danych   
a złożonością implementacji.

Główna część pracy, została oparta na algorytmie filtracji zbiorowej, opisanym w [1], którego predykcja nieznanych ocen powstaje na podstawie rozkładu macierzy znanych ocen. Podejście to polega na znalezienie podobieństw we wzorcach zachowania użytkowników aby na tej podstawie przewidzieć ich gusta i opinie na temat rekomendowanych obiektów.   
Nie wykorzystuje ono jednak wiedzy na temat charakterystyki obiektów i użytkowników.

Zaproponowane podejście jest próbą opracowania metody włączenia tego typu danych do algorytmu rozkładu macierzy ocen. Wymagało to opracowania sposobu przechowywania danych jak i modyfikacji samego procesu uczenia modelu.

Jako że, jest to nowo zaproponowane podejście, wymagało ono weryfikacji, czy wykorzystanie obu typów informacji (ocen oraz cech użytkowników i obiektów) wpłynie pozytywnie na uzyskane wyniki. Z tego względu w pracy zostały użyte dane testowe na temat interakcji   
z dwoma różnymi typami obiektów: filmami oraz restauracjami. Z tego względu, część pracy została poświęcona analizie i odpowiedniemu ich przygotowaniu, aby umożliwić ich wykorzystanie w procesie uczenia.

Na ich podstawie zostało sprawdzone czy algorytm może zostać użyty z powodzeniem   
w standardowym przypadku, dla algorytmów rozkładu macierzy, gdy znana jest historia aktywności użytkownika jak i w wypadku nowych przedmiotów lub użytkowników   
w systemie.

## Organizacja pracy

Dalsza część pracy została podzielona w następujący sposób.

* Rozdział 2 przedstawia ogólne założenia dotyczące systemów rekomendacyjnych oraz charakterystykę wybranych algorytmów: zasad ich działania, wad i zalet oraz sposobu weryfikacji ich wyników.
* Rozdział 3 zawiera opis algorytmu hybrydowego, opracowanego i zaimplementowa-nego dla celów pracy oraz narzędzi użytych przy jego implementacji, analizie danych   
  i procesie weryfikacji. Dodatkowo w rozdziale tym zostały opisane zbiory danych, użyte do eksperymentalnego badania jego skuteczności: ich zawartość i sposób jej analizy oraz wyniki uzyskane z użyciem proponowanego algorytmu jak   
  i algorytmów z rozdziału drugiego.
* Rozdział 4 stanowi podsumowanie pracy: wnioski jak i propozycję jej dalszego rozwoju.

# Systemy rekomendacyjne

Każdy z systemów rekomendacyjnych, niezależnie od algorytmu czy implementacji, działa   
w zbliżony sposób. Na podstawie danych wejściowych wybiera on, spośród wszystkich możliwych do polecenia obiektów, pewien ich podzbiór jak i kolejność zawartych w nim elementów. Decyzja ta podejmowana jest w oparciu o przewidywaną trafność rekomendacji, bazującą na dostępnej wiedzy. Sam wybór prezentowanych obiektów może zostać zrealizowany z użyciem jednego z wielu podejść, których sposób działania jak i weryfikacji zostały omówione w tym rozdziale.

## Metody filtracji zbiorowej

Systemy rekomendacyjne oparte na filtracji zbiorowej są najpopularniejszych z obecnie stosowanych podejść [1]. W swojej podstawowej formie, nie wykorzystują one informacji na temat rekomendowanych obiektów ani użytkowników. Pozwala to na pominięcie kosztownego procesu zbierania danych i budowania profili użytkowników jak i obiektów. Zamiast tego wykorzystywane są w nich dane historyczne na temat aktywności użytkowników w systemie, między innymi to jakie obiekty ich interesują, jak wysoko je oceniają czy jak wiele czasu spędzają na ich podstronach w serwisie internetowym. Wzorce w tym zachowaniu mogą być następnie użyte, aby odkryć powiązania między użytkownikami i obiektami i uzyskać rekomendacje niosące w sobie największą wartość.

Dane wykorzystywane w tym procesie mogą należeć do dwóch kategorii:

* oceny jawne (explicit rating) – są to wszystkie oceny wystawiane przez użytkowników w formie deklaracji co do wartości (np. lubię/nie lubię, 4 na 5 gwiazdek),
* oceny niejawnej (implicite rating) – zainteresowanie użytkownika może zostać jednocześnie wyrażone poprzez samo zachowanie bez jasno wyrażonej opinii; może to być na przykład procent w jakim obejrzało się film czy liczba odsłon strony.

Analiza obu typów danych wymaga istotnie różniących się względem siebie podejść. Z tego też względu, w pracy skupiam się jedynie na analizie ocen jawnych. W ich wypadku istnieją dwa główne podejścia oparte na różniących się od siebie założeniach: metody najbliższego sąsiedztwa i rozkładu macierzy, które omówię w dalszej części rozdziału.

### Użyte oznaczenia

Rozwiązania opisane w tym rozdziale, są przedstawione z użyciem poniższego zbioru oznaczeń:

– wybrani użytkownicy,

– wybrane obiekty,

– liczba użytkowników oraz obiektów w systemie,

– ocena wystawiona przez użytkownika dla obiektu ,

– ocena przewidywana przez system dla użytkownika i obiektu ,

– zbiór ocen wystawionych przez użytkownika oraz zbiór ocen obiektu ,

– średnia ocen w systemie,

– zbiór użytkowników,

– zbiór użytkowników, którzy ocenili obiekt ,

– zbiór użytkowników, którzy ocenili zarówno obiekt jak i ,

– Zbiór obiektów,

– zbiór obiektów ocenionych przez użytkownika ,

– zbiór obiektów ocenionych zarówno przez użytkownika jak i .

– zbiór najbardziej podobnych użytkowników do użytkownika

### Metody najbliższego sąsiedztwa

Powszechnie stosowanym podejściem do problemu rekomendacji są metody najbliższego sąsiedztwa, zamiennie nazywane metodami „pamięciowymi” (memory-based methods) [3], [7]. Bazują one na wyszukiwaniu użytkowników o podobnym guście lub potrzebach. Pozwala to na symulowanie zachowania znanego z życia, gdy dwie osoby o zbieżnych gustach, polecają sobie np. filmy czy restauracje. Dzięki temu mogą one odkrywać obiekty, o których wcześniej nie wiedziały albo nie były co do nich pewne, a które mimo tego mogą się im spodobać.

Opisane poniżej podejście, zaproponowane w [4], jest stosowane, w jednakowy sposób, zarówno do wyszukiwania podobieństwa użytkowników jak i obiektów. Z tego też względu skupię się jedynie na pierwszym z nich.

Aby odszukać użytkowników o podobnych gustach, wprowadza się metrykę podobieństwa między nimi, oznaczoną dalej jako . Podobieństwo to nie jest określane bezpośrednio przez informacje o użytkownikach jak wiek czy płeć lecz przez ich aktywność   
w systemie, reprezentowaną przez wystawione oceny. Wyliczenie podobieństwa między parami użytkowników pozwala następnie na znalezienie zbiorów najbardziej podobnych użytkowników do każdego z nich. W wypadku użytkownika zbiór ten można oznaczyć   
jako , gdzie wyznacza pewną ustaloną, niewielką liczbę użytkowników wybieranych jako najbliżsi sąsiedzi.

Posiadając zbiór tych użytkowników można następnie spróbować przewidzieć ocenę dla obiektu , którego użytkownik jeszcze nie widział. Ocenę tą wyraża się jako średnią ocen wystawionych obiektowi przez użytkowników ze zbioru , ważoną i znormalizowaną względem podobieństwa użytkowników należących do zbioru do użytkownika .

Przewidywaną ocenę wyznacza się wzorem (1).

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Wzór ten nie uwzględnia jednak dwóch zjawisk związanych z procesem wystawiania oceny: jej interpretacji oraz sposobu wykorzystania skali ocen. W pierwszym z wypadków, ta sama ocena może oznaczać dla różnych użytkowników zupełnie co innego. O ile jeden   
z użytkowników interpretuje 3 jako przeciętną ocenę, drugi może w ten sam sposób traktować ocenę 2. Aby uniknąć tego problemu, autorzy uwzględnili we wzorze na przewidywaną ocenę również średnią ocen wystawionych przez użytkownika:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Drugi problem został natomiast uwzględniony w pracy [3]. Użytkownicy mogą wystawiać oceny z wykorzystaniem pełnej ich skali. Jednak niektórzy z nich zachowują wartości krańcowe dla szczególnych obiektów i z tego względu nie wystawiać tych ocen nigdy lub sporadycznie. Zjawisko to jest ograniczone poprzez wprowadzenie odchylenia standardowego ocen wystawionych przez użytkownika jako współczynnika skalującego rozkład jego ocen, co   
w połączeniu z odjęciem średniej prowadzi w efekcie do standaryzacji ocen. Wzór na przewidywaną ocenę, zawierający tę i poprzednią zmianę można ostatecznie przedstawić jako:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Komplet przewidywanych w ten sposób ocen jest w ostatnim kroku sortowany a obiekty   
o najwyższej ocenie prezentuje się użytkownikowi.

Możliwe jest, że wartość podobieństwa dla pary sąsiadów , przyjmuje wartości ujemne.   
W wypadku wzoru (3) nie jest to jednak wadą opisanego podejścia. Ocena negatywna użytkownika będzie interpretowana jako pozytywny odbiór przez użytkownika . Wpływ na to ma ujemna wartość powodująca zmianę znaku wyrażenia w liczniku.

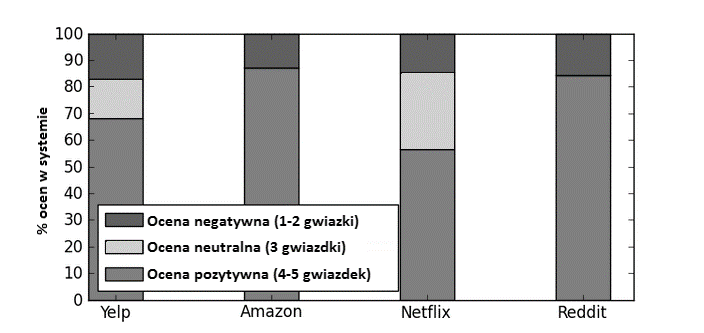
### Metryki podobieństwa

Wykorzystanie metody najbliższego sąsiedztwa wymaga wyznaczenia podobieństwa między użytkownikami. W dalszej części rozdziału opisuję kilka najpopularniejszych miar opisanych   
w [5].

**Indeks Jaccarda**Najprostszą z wybranych miar podobieństwa jest indeks Jaccarda , zdefiniowany jako proporcja liczby obiektów ocenionych przez obu użytkowników do liczby ocenionych przez dowolnego z nich. Funkcja ta przyjmuje wartości od 0 dla zbiorów rozłącznych do 1 przy ich równości.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Użycie indeksu Jaccarda, jako metryki podobieństwa, może wydawać się nieintuicyjne. Przykładowo, jeśli użytkownik ocenił 5 przedmiotów pozytywnie i 3 negatywnie   
a użytkownik ma o nich odwrotną opinię i ocenił dodatkowo 2 inne przedmioty, to wartość wynosi 0.8. Oznacza ona duże podobieństwo między użytkownikami, mimo istotnych różnic w gustach. Znając dystrybucję ocen w systemie, problem ten może zostać jednak zignorowany. Użytkownicy odbierają pozytywnie większość obiektów, które oceniają. Dlatego, jeśli dwóch użytkowników oceniło co najmniej kilka wspólnych obiektów, to z dużym prawdopodobieństwem można stwierdzić, że większość z nich podobała się obu z nich.



Rys. 4 Dystrybucje ocen w kilku z popularnych portali internetowych [6]

**Odległość Euklidesowa**W wypadku, gdy konieczne jest jednak uwzględnienie wartości ocen, można wziąć pod uwagę, jak bardzo różnią się oceny każdego z obiektów, dla wybranej pary użytkowników. Traktując oceny jako wartości kolejnych współrzędnych pewnej przestrzeni, możliwe jest zaprezentowanie ich w postaci pary wektorów odpowiadające obu użytkownikom. Odległość między tymi wektorami definiuje jak bardzo użytkownicy, różnią się od siebie. Ograniczając się do podprzestrzeni dla obiektów ocenionych przez obu użytkowników, odległość Euklidesowa wyraża się wzorem:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Na oceny wystawione przez użytkownika wpływa jednak kilka czynników. Jednym z nich jest obciążenie (bias). Ocena pozytywna, może być wyrażana przez inny zakres wartości   
u dwóch różnych użytkowników. Z tego też powodu przy obliczaniu odległości między użytkownikami, popularnym podejściem jest odjęcie średniej oceny wystawianej przez nich od wszystkich ich ocen i badanie samej wariancji wartości. Odległość między wektorami przyjmuje wtedy zmodyfikowaną formę:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Odległość nie jest jednak miarą podobieństwa. Z tego też względu przy wyborze sąsiadów danego użytkownika, wybiera się tych o jak najmniejszej odległości względem niego. Podobieństwo natomiast wyraża się jako:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

**Podobieństwo kosinusowe**Podobieństwo między użytkownikami może zostać również określone jako kosinus kąta między wektorami ocen. Podejście to porównuje proporcję między wartościami ocen zamiast ich bezpośredniej wartości.

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

**Współczynnik korelacji Pearsona**Współczynnik korelacji Pearsona jest to miara współliniowości między dwoma szeregami.

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Przy podstawieniu:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |
|  |  | (11) |

wzór ten przyjmuje postać identyczną do wzoru na podobieństwo kosinusowe.

Wartość PCC należy do przedziału [-1; 1]. Dopuszcza to przypadek pary sąsiadów   
o negatywnie skorelowanych preferencjach. Zostało to jednak rozpatrzone w procesie wyznaczania przewidywanej wartości ocen w oparciu na równaniu (3).

**Współczynnik korelacji rang Spearmana**Ostatnią z omawianych miar podobieństwa jest współczynnik korelacji rang Spearmana . Przypomina ona w założeniach korelację Pearsona, jednak w odróżnieniu od niej nie wskazuje na stopień zależności liniowej a dowolną zależność monotoniczną. W tym celu wprowadza się funkcję porządkującą , przypisującą rangę dla każdej oceny dla której obiekt został oceniony przez obu użytkowników.

Przy podstawieniu:

|  |  |  |
| --- | --- | --- |
|  |  | (12) |
|  |  | (13) |

wzór na korelację rang Spearmana przyjmuje postać identyczną do wzoru na współczynnik korelację Pearsona.

### Zalety oraz problemy metod najbliższego sąsiedztwa

Metody najbliższego sąsiedztwa są dobrze znanym i stosowanym rozwiązaniem, czego najlepszym przykładem może być użycie zmodyfikowanej metody podobieństwa   
obiekt-obiekt w rekomendacjach sklepu Amazon.com [7]. Ich główną zaletą jest niezależność od dziedziny w której są stosowane, całość algorytmu działa w oparciu o oceny, bez konieczności analizowania czego one dotyczą. Jednocześnie jednak algorytmy te borykają się z grupą istotnych problemów.

**Zimny start**Problem określany jako zimny start [1] (cold start) występuje w wypadku obiektów, które dopiero co zostały dodane do systemu i świeżo stworzonych kont użytkowników.   
Bez dostatecznej ilości danych, systemy nie może wygenerować rekomendacji ani zagwarantować ich jakości. Aby uniknąć tego problemu, stosuje się inne metody rekomendacji, do czasu uzyskania odpowiedniej liczby ocen.

**Wiarygodność sąsiedztwa**Tabela poniżej przedstawia oceny trójki użytkowników. W momencie działania algorytmu znane są jedynie oceny w kolorze czarnym.

Tab. 2 Przykładowe oceny wystawione przez użytkowników

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Anna | 5 | 5 | 1 | 3 | 3 | 5 | 1 | 3 |
| Marcin | 1 | 2 | 4 | 3 | 3 | 2 | 4 | 2 |
| Adam | 1 | 2 | 5 | 3 | 3 | 2 | 5 | 1 |

Wyliczając podobieństwo innych użytkowników do Marcina, metryki przyjmują wartości   
z tabeli 3.

Tab. 3 Metryki podobieństwa między ocenami użytkowników

|  |  |  |
| --- | --- | --- |
|  | Anna | Adam |
|  | 0.400 | 0.750 |
|  | 1.000 | 0.500 |
|  | 1.000 | 0.500 |
|  | 1.000 | 0.994 |
|  | 1.000 | 0.816 |

Mimo neutralnego i pozytywnego odbioru, tych samych obiektów przez Marcina i Adama to Anna uznana jest za bardziej podobną do pierwszego z nich. Jednak w wypadku gdyby, znana była większa liczba ocen, sytuacja uległaby odwróceniu. Jedną z przyczyn tego zjawiska jest niewystarczająca liczba wspólnych ocen. W przypadku, gdy znana jest tylko jedna wspólna ocena, problem ten jest jeszcze bardziej widoczny, gdyż dwie ostatnie metryki przyjmują maksymalną wartość niezależnie od wartości ocen.

W celu ograniczenia skutków tego zjawiska, wprowadza się dodatkową miarę pewności , której przykładem może być:

|  |  |  |
| --- | --- | --- |
|  |  | (14) |

Dla pewnego ustalonego , oznaczającego minimalną liczbę ocen, potrzebnych do uzyskania pewności co do podobieństwa, zastępuje się funkcję zarówno przy wyszukiwaniu sąsiadów jak i wyliczaniu przewidywanych ocen funkcją :

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

**Bańka filtrująca**W sytuacji, gdy polecane, obiekty podobają się tym samym użytkownikom, upodabniają się oni coraz bardziej do siebie. Z tego względu pula sugerowanych obiektów zawęża się do coraz mniejszego podzbioru. Gwarantuje to pozytywny odbiór prezentowanej treści, ale jednocześnie zmniejsza szansę na zapoznanie użytkownika z nowym, szerszym wyborem.

**Szara owca**Pojęciem szarej owcy (grey sheep) określa się zazwyczaj użytkownika, którego zachowanie łączy w sobie charakterystykę więcej niż jednej grupy w populacji. Z tego też względu, znalezienie użytkowników podobnych do niego jest utrudnione a dobrana rekomendacja może niekoniecznie wpasowywać się w gusta tej osoby.

**Przechodniość sąsiedztwa**

Tab. 4 Przypadek pesymistyczny dla wyszukiwania sąsiedztwa

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Marcin | 5 | 2 | 5 | 3 | 3 | 2 | 5 | 1 |
| Adam | 5 | 2 | 5 | 3 | 3 | 2 | 5 | 1 |
| Karol | 5 | 2 | 5 | 3 | 3 | 2 | 5 | 1 |

Mimo, że wszyscy trzej użytkownicy wystawiliby te same oceny, niemożliwe jest porównanie Marcina z Adamem. W przypadku prawdziwych danych, sytuacja ta jest dosyć częsta, jako że użytkownicy o tym samym guście mogli ocenić zupełnie inne obiekty. Problemowi temu towarzyszy zjawisko synonimu, gdy ten sam lub bliźniaczo podobny obiekt może mieć   
w zbiorze danych co najmniej dwie oddzielne reprezentacje. Przykładem tego może być np. książka sprzedawana w różnych językach.

**Złożoność obliczeniowa**Wyznaczenie predykcji ocen dla wszystkich par (użytkownik, obiekt), wymaga obliczeń   
o złożonościgdzie oznacza liczbę zarekomendowanych obiektów, a kolejne składniki odpowiadają za złożoność wyliczenia podobieństwa dla wszystkich par użytkowników, wyznaczenie przewidywanych ocen oraz wybranie najlepszych rekomendacji dla każdego z użytkowników. W wypadku systemów z milionami obiektów   
i użytkowników, przedstawiony algorytm nie może zostać użyty w swojej podstawowej postaci.

### Metody rozkładu macierzy

Alternatywą do podejścia opartego na sąsiedztwie są metody rozkładu macierzy (matrix factorization) [1]. W ich wypadku, zebrane oceny można przedstawić w postaci macierzy rzadkiej , której wiersze oznaczają oceny kolejnych użytkowników a kolumny oceny obiektów.

W wypadku gdyby wszystkie wartości tej macierzy byłyby znane, możliwy byłby jej rozkład na wartości osobliwe (singular value decomposition, SVD) [8]:

|  |  |  |
| --- | --- | --- |
|  |  | (16) |
|  |  | (17) |
|  |  | (18) |
|  |  | (19) |

gdzie to macierz jednostkowa o wymiarze .

Macierze , , niosą w sobie pełną informację na temat macierzy . Mogą one zostać również użyte do przybliżenia jej wartości poprzez zastąpienie macierzami , , spełniającymi warunki:

|  |  |  |
| --- | --- | --- |
|  |  | (20) |
|  |  | (21) |
|  |  | (22) |
|  |  | (23) |

gdzie dla danego norma Frobeniusa (pierwiastek z sumy kwadratów elementów macierzy) jest minimalna.

W celu wyznaczenia powyższych macierzy, konieczne jest jednak dodatkowe założenie co do monotoniczności wartości na głównej przekątnej macierzy .

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

Warunek ten można spełnić poprzez posortowanie wartości na wspomnianej przekątnej, przy jednoczesnej odpowiedniej zamianie kolejności kolumn macierzy i . Przykładowo, zamiana miejscami wartości z wiąże się z zamianą miejscami kolumn z orazz.

Na mocy twierdzenia Eckarta-Younga-Mirsky’ego [9] najlepsze przybliżenie można uzyskać poprzez wybranie pierwszych kolumn macierzy i oraz pierwszych wierszy i kolumn macierzy , dla których spełnione są założenia (20)-(24). Zachowane wartości niosą w sobie najistotniejszą część informacji na temat macierzy .

Podejście to zostało zaadoptowane na cele systemów rekomendacyjnych, przez przyjęcie założenia, że na ocenę obiektu wpływa zbiór cech ukrytych (latent factors). Mogą one opisywać łatwo interpretowalne aspekty charakterystyki obiektów, takie jak ogólna jakość dań w restauracji czy przynależność filmu do kina akcji, jak i ich własności niemożliwe do oczywistej identyfikacji przez człowieka. Analogiczna sytuacja zachodzi w wypadku użytkowników. Cechy te można przedstawić w postaci dwóch macierzy odpowiadających kolejno użytkownikom i przedmiotom, powstałe przez włączenie wartości macierzy do pozostałych.

|  |  |  |
| --- | --- | --- |
|  |  | (25) |
|  |  | (26) |
|  |  | (27) |

Mimo, że matematyczne sformułowanie SVD nie dopuszcza macierzy rzadkich, możliwe jest wykorzystanie inspirowanych nim algorytmów również w wypadku częściowo wypełnionej macierzy ocen. Algorytmy te mogą zostać użyte nie tylko do wyznaczenia macierzy i , pozwalających na odtworzenie znanych wartości ale i próby przewidzenia pozostałych z nich. Wymaga to jednak zastosowania algorytmu mogącego działać na macierzach rzadkich.   
Dwa z takich podejść zostały opisane w dalszej części pracy.

### Oszacowanie bazowe

Algorytmy inspirowane SVD są poprzedzane dodatkowym krokiem. Tak jak w wypadku metod opartych na najbliższym sąsiedztwie, dystrybucja ocen zależna jest od użytkowników   
i obiektów, których dotyczą. Jednym ze stosowanych podejść jest wyznaczenie obciążeń (bias) metodą opisaną w [10].

Każdą ocenę w systemie można w pewnym stopniu przybliżyć przez średnią wartość ze wszystkich ocen , obciążenie związane z obiektem , oraz obciążenie związane   
z ocenami użytkownika . Podejście to pozwala na uchwycenie takich aspektów jak przeciętny odbiór obiektu. Przykładowo film „Skazany na Shawshank” oceniony został   
w portalu IMDB średnio na 9.3 gwiazdki z 10, co jest zdecydowanie wyższą oceną niż średnia 6.4. Podobny efekt uwzględniony jest w wypadku użytkowników, którzy posiadają tendencję do zawyżania lub zaniżania swoich ocen względem reszty populacji. Wartość wyjściowa oceny (baseline) użytkownika oraz wyliczana jest na podstawie wzorów (28)-(30).

|  |  |  |
| --- | --- | --- |
|  |  | (28) |
|  |  | (29) |
|  |  | (30) |

Współczynniki tłumienia uwzględniają wpływ liczby ocen na obciążenie.   
Im więcej ocen zostało wystawionych danemu obiektowi / przez danego użytkownika, tym pewniejsza jest wyliczona wartość. Dla niewielkiej liczby ocen oraz przyjmują wartość bliską 0.

Podejście to zapobiega sytuacji w której na podstawie niewielkiej liczby ocen, określana jest jakość obiektu lub tendencja użytkownika do oceniania obiektów powyżej lub poniżej średniej. Przykładowo jeśli użytkownik ocenił jedynie dwa obiekty na 4 i 5 przy średniej ocenie   
w systemie równej 3, nie oznacza to że pierwszy obiekt ocenia poniżej swojej przeciętnej oceny (4.5) a drugi powyżej. Dla odpowiednio wysokiego współczynnika oceny te zostaną odebrane jako dwie pozytywne wartości, jako że średnia ocena dla tego użytkownika przyjmie wartość mniejszą od 4 (dla ).

### Simon Funk SVD

Jednym ze sposobów na wyznaczenie wartości macierzy cech ukrytych jest podejście stochastyczne zaproponowane przez Simona Funka [11] w trakcie konkursu, zorganizowanego przez Netflix.

Podstawą tego rozwiązania jest równoległa aktualizacja kolejnych kolumn obu macierzy dla każdej oceny w zbiorze:

|  |  |  |
| --- | --- | --- |
|  |  | (31) |
|  |  | (32) |

gdzie to wybrana stała uczenia.

Oba te kroki są jednak dodatkowo zmodyfikowane o współczynnik regularyzacji , mający zapobiegać przeuczeniu się modelu, na podstawie zbyt małej liczby ocen. W ostatecznej formie (przy uwzględnieniu oszacowania bazowego) przyjmują one postać:

|  |  |  |
| --- | --- | --- |
|  |  | (33) |
|  |  | (34) |

Algorytm może zostać przedstawiony w formie poniższego pseudokodu.

Simon Funk SVD

Wylicz wartości , ,

Wypełnij macierze , niewielką dodatnią wartością

Powtarzaj do osiągnięcia zbieżności:

Dla każdej oceny z permutacji zbioru ocen :

Dla każdego ze zbioru :

### ALS

Metoda naprzemiennych najmniejszych kwadratów (alternating least square, ALS) jest drugim z podejść do wyznaczenia macierzy i , opisanym w [12]. W jej wypadku, minimalizowana jest wartość funkcji kosztu , wyrażona wzorem:

|  |  |  |
| --- | --- | --- |
|  |  | (35) |

Wartość , tak jak w wypadku algorytmu SVD, ma zapobiec przeuczeniu się modelu   
w oparciu na (normie Frobeniusa/pierwiastku z sumy kwadratów elementów macierzy) macierzy i .

Funkcja ta nie jest funkcją wypukłą, ze względu na występowanie wyrazu . Z tego też względu, nie posiada ona minimum lokalnego. Działanie algorytmu ALS polega jednak na częściowym zignorowaniu tego problemu. Zakładając, że wartości jednej z macierzy są ustalone, równanie to przyjmuje postać funkcji kwadratowej, o dobrze zdefiniowanym rozwiązaniu.

Algorytm ALS zawdzięcza swoją nazwę temu, że w kolejnych krokach, naprzemiennie, ustalana jest wartość jednej spośród macierzy i po czym przy tej ustalonej wartości uaktualniana jest wartość drugiej z nich. Podejście to wykonuje przejście do optimum lokalnego w jednym kroku i znane jest jako zstępowanie po współrzędnych (coordinate descent). Sam algorytm opisany jest poniższym pseudokodem. Za pomocą oznaczona została w nim macierz jednostkowa wymiaru .

ALS

Wypełnij macierze , losowymi wartościami z rozkładu jednorodnego

Powtarzaj do osiągnięcia zbieżności:

Dla każdego użytkownika :

Dla każdego obiektu :

W odróżnieniu do algorytmu SVD, w którym wartości cech ukrytych są aktualizowane jedna po drugiej dla kolejnych ocen w systemie, algorytm ALS należy do klasy rozwiązań „żenująco” prostych do zrównoleglenia (embarrassingly parallel). Wyliczenie wartości kolejnych wierszy macierzy, może zostać wykonane jednocześnie dla wszystkich użytkowników albo obiektów. Dzięki temu, popularne są wersje algorytmu w formie zrównoleglonej [13] oraz rozproszonej [14].

### Zalety i wady algorytmów rozkładu macierzy

Czołowe rozwiązania zaproponowane w ramach konkursu Netflix wykorzystywały algorytmy rozkładu macierzy. Łączą one wysoką wydajność oraz dobre wyniki przewidywań. Tak jak   
w wypadku metod najbliższego sąsiedztwa, nie zależą one od dziedziny rekomendacji ale   
i posiadają te same problemy jak zimny start, szara owca czy bańka filtrująca. Jednocześnie posiadają inną istotną wadą jaką jest objaśnialność rekomendacji.

O ile w wypadku metod najbliższego sąsiedztwa, rekomendacja może zostać zinterpretowana, na przykład poprzez zdania jak „użytkownikom podobnym do Ciebie podobało się …” albo „obiekty podobne do oglądanego to …”, to w wypadku rekomendacji powstałych z rozkładu macierzy nie istnieje taka forma uzasadnienia. Wiąże się to ze sposobem wyznaczenia cech ukrytych, które reprezentują pewne wzorce występujące w danych, ale nie mają jawnie określonego sensu. Dla części cech, możliwe jest niekiedy określenie ich znaczenia na przykład przez analizę tych obiektów, które zostały ocenione skrajnie pozytywnie i negatywnie. Jednak pozostałe z nich mogą nie znaleźć odzwierciedlenia w łatwo opisywalnym zestawie pojęć.

## Systemy oparte na treści

Alternatywnym podejściem do algorytmów opartych o dane historyczne są rozwiązania oparte na treści (content-based recommendation). Działają one wykorzystując informacje zgromadzone w systemie na temat użytkowników oraz obiektów aby na tej podstawie dobrane zostały najbardziej trafne rekomendacje [15].

Przykładem takiej reprezentacji może być wektor cech charakteryzujący restauracje   
w portalu Yelp. Częścią z nich są informacje takie jak typ kuchni, godziny otwarcia, średnia cena, klimat lokalu i jego lokalizacja czy możliwość rezerwacji stołu. Tak zebrane informacje mogą następnie posłużyć do rekomendacji w oparciu o warunki zapytania: „restauracja, otwarta przez najbliższą godzinę, w centrum” jak i z użyciem podobieństwa miedzy obiektami, analogicznego do metod najbliższego sąsiedztwa.

Podobny zbiór danych może także opisać użytkownika, jego płeć, wiek, narodowość, wykształcenie czy zainteresowania. Informacje te zbierane są zarówno z jawnie podanych informacji, jak i z metadanych oraz aktywności w systemie. Na podstawie tych informacji można następnie zbadać szansę na „polubienie” wybranego obiektu na podstawie naiwnych klasyfikatorów bayesowskich czy metod opartych o drzewa decyzyjne i lasy losowe [15].

Rozwiązania oparte na zawartości niwelują problem zimnego startu, jaki występował   
w wypadku metod filtracji zbiorowej. Rekomendacja może zostać wyznaczona na podstawie profil użytkownika albo obiektu, w momencie jego stworzenia i bez jakichkolwiek dodatkowych danych historycznych. Dodatkowo, rekomendacja ta może zostać zaprezentowana z jednoczesnym uzasadnieniem, na podstawie jakich cech została ona sformułowana.

Z drugiej strony wymaga to jednak stworzenia profilu na tyle wiarygodnego i kompletnego, aby możliwe było jego spożytkowanie. Zadanie to może być kosztowne i czasochłonne, ze względu na konieczność opracowania go dla każdego nowego obiektu w systemie.   
W wypadku profilu użytkownika, dodatkowym wymogiem jest jego gotowość do wprowadzenia informacji o sobie, czy trafność ich przybliżenia na podstawie innych źródeł. Należy wziąć ponadto uwagę, że informacje zawarte w profilach ograniczają możliwości modelowania rzeczywistości jedynie do reprezentowanych przez nie cech. W takim wypadku, konieczne jest jak najdokładniejsze zapoznanie się z dziedziną w której dany system ma zostać zastosowany i wybranie cech, które są istotne dla dobrego działania algorytmu.

## Systemy hybrydowe

Ostatni typ systemów nie tworzy oddzielnej kategorii. Zamiast tego, łączy on aspekty systemów opartych o zawartość i metod filtracji zbiorowej. Głównym celem ich wprowadzenia jest złagodzenie tego jak wady algorytmów bazowych wpływają na ostateczną rekomendację. Istnieje kilka typów podejść hybrydowych opisanych w [15].

### Systemy ważone

Podstawą dla systemów ważonych jest użycie kilku algorytmów rekomendacyjnych jako niezależnie działających systemów i połączenie końcowych wyników ich działania. Operacja ta może zostać oparta zarówno o prostą kombinację liniową, jak i bardziej wyszukane podejścia, jak uaktualnienie wagi w wyniku sprzężenia zwrotnego wynikającego z wystawienia oceny przez użytkownika [15].

### Systemy przełączane

Podobnym do poprzedniego podejścia są metody wybierające algorytm, który ma zostać zastosowany do rekomendacji w danym momencie. Decyzja ta podejmowana jest   
z uwzględnieniem czynników takich jak wiarygodność i jakość zwracanych wyników czy możliwość wyznaczenia rekomendacji dla wybranego użytkownika [15].

### Systemy mieszające

Podejście to łączy wyniki, użytych algorytmów, poprzez jednoczesną ich prezentację użytkownikowi. Ich prezentacja, w niezmienionej formie, daje użytkownikowi wybór oparty na zaletach każdego z podejść i jednocześnie nie wiąże całej rekomendacji z wadami pojedynczego podejścia [15].

### Systemy kaskadowe

W systemach tych, algorytmom rekomendacyjnym przypisywane są priorytety. Definiują one kolejność tworzenia lub łączenia rekomendacji w taki sposób, że podzbiór rekomendowany przez algorytm o niższym priorytecie musi zawierać się w podzbiorze rekomendowanym przez algorytm o wyższej wartości [15]. Podejście to pozwala na działanie w wypadku gdy pojedynczy algorytm nie może rozstrzygnąć, która rekomendacja jest najtrafniejsza.

### Systemy łączące źródła

Systemy łączące źródła odbiegają w swoim działaniu od wcześniej wymienionych kategorii algorytmów hybrydowych. Ich działanie, zamiast modyfikowania wynik rekomendacji, polega na użyciu kilku źródeł danych w procesie uczenia pojedynczego algorytmu. Pozwala to na uchwycenie wzorców w niezależnych źródłach danych, które zostałyby pominięte   
w wypadku użycia tylko jednego z nich [15].

### Systemy oparte na poszerzeniu danych wejściowych

Ostatnią z kategorii są systemy, których działanie opiera się na rozszerzeniu zbioru danych wejściowych przez dane wyjściowe z poprzedniego algorytmu. Mogą to być na przykład przewidywane wartości oceny, które po przekazaniu do następnego z algorytmów traktowane są jako faktycznie wystawione przez użytkownika [15].

## Sposób oceniania algorytmów

W celu porównania wyników działania algorytmów, konieczna jest definicja miary ich jakości. Systemy rekomendacyjne posiadają dwie zasadnicze grupy miar, których zastosowanie zależy od możliwości projektowych i celu ich użycia.

### Metryki online

W wypadku gdy sprawdzane jest ostateczne rozwiązanie, częstym podejściem jest wprowadzenie testów A/B, które weryfikują działanie z uwzględnieniem wynikającego   
z niego sprzężenia zwrotnego. Dopiero gdy dostępna jest informacja zwrotna, jak rekomendacja wpływa na sposób korzystania użytkownika z systemu, możliwe jest ocenienie jego skuteczności. Zależnie od celu biznesowego, zjawisko to może być mierzone na różne sposoby jak średnia ilość kliknięć czy retencja użytkownika i zyski ze sprzedaży [16].

### Metryki offline

Zastosowanie oceny online może nie być możliwe, np. ze względu na wczesną fazę rozwijanego projektu czy działanie na użyczonych z zewnątrz danych. Sytuacja taka występuje w szczególności w pracach badawczych porównujących skuteczność różnych algorytmów bez ich faktycznego wdrożenia do funkcjonującego w rzeczywistych warunkach systemu. Z tego też powodu, istnieje szereg metryk, określanych jako offline, których wartość może być   
w całości wyliczona na podstawie danych zebranych w danym okresie.

W tym celu, konieczny jest standardowy podział na zbiór treningowy i testowy. Dane te można podzielić stosując jeden z kilku sposobów opisanych w [17].

Podział ze wstrzymaniem próbki (holdout method) używany jest do wytrenowania algorytmu na części danych, nazywanej zbiorem treningowym, oraz zweryfikowania działania na rozłącznym względem niego zbiorze testowym. W najprostszym przypadku może to być losowy podział zbioru ocen z ustaloną proporcją (często stosowanym jest np. 70%/30%). Alternatywą dla tego podziału jest użycie losowania warstwowego (stratified sampling) [18], którego celem jest zachowanie dystrybucji wybranych cech dla obu zbiorów   
(w wypadku systemów rekomendacyjnych może to być dystrybucja wartości ocen).

Kolejnym z podejść jest użycie informacji o czasie wystawienia oceny. Zbiór testowy składa się w takim wypadku z najnowszych ocen, losowo wybranej grupy użytkowników. Podobny proces podziału oparty jest na pozostawieniu co najwyżej jednej oceny (niekoniecznie ostatniej) w zbiorze testowym (leave-one-out). Podejście oparte o wstrzymanie próbki obarczone jest jednak sporym ryzykiem niedoszacowania lub przeszacowania wartości wykorzystanych metryk, ze względu na dużą losowość tego które oceny trafią do którego ze zbiorów.

Innym ze sposobów jest k-krotna weryfikacja krzyżowa (k-fold cross validation), polegająca na podziale zbioru ocen na k w miarę równych zbiorów (np. przy użyciu holdout method lub stratified sampling). Następnie, każdy ze zbiorów jest traktowany jako zbiór testowy   
a połączenie pozostałych jako zbiór treningowy. Dla modelu wytrenowanego dla każdego   
z podziałów, metryki jakości wyliczane są oddzielnie. Następnie mogą one zostać zagregowane w średnią wartość jak i dodatkowe informacje takie jak wariancja otrzymanych wyników. Metoda ta daje bardziej wiarygodny wgląd w jakość i stabilność rozwiązania   
w porównaniu do podziału ze wstrzymaniem próbki.

W wypadku wszystkich podejść, wymienione w dalszej części rozdziału metryki mogą zostać zastosowane zamiennie. Ich użycie zależy jedynie od badanych aspektów, otrzymanych rekomendacji. Wyjątkiem od tego jest podejścia leave-one-out dla którego nie można wykorzystać metryk wymagających więcej niż jednej oceny. W tym wypadku, możliwe jest użycie jedynie tych opartych o błąd.

**Błąd średniokwadratowy**Błąd średniokwadratowy (root mean square error, RMSE) jest to miara określona dla pary uporządkowanych zbiorów wartości oraz ich predykcji , wyrażona wzorem [19]:

|  |  |  |
| --- | --- | --- |
|  |  | (36) |

Miary tej używa się przede wszystkim aby uwydatnić wpływ oszacowań obarczonych dużym błędem. Im jest on większy tym szybciej wzrasta wartość błędu, co pozwala na faworyzowanie stabilniejszych modeli o mniejszej liczbie wartości odstających jak i niższej wartości związanego z nimi błędu.

**Błąd bezwzględny**Błąd bezwzględny (mean absolute error, MAE) określa się w podobny sposób do RMSE jako [19]:

|  |  |  |
| --- | --- | --- |
|  |  | (37) |

Wartość ta odpowiada średniemu błędowi, tak więc nie jest w niej uwzględnione rozróżnienie między jego dużymi wartościami a proporcjonalnie większą liczbą błędów   
o małej wartości.

**Precyzja od k oraz czułość od k**

Precyzja od k (precision at k, ) jest metryką określającą średnią proporcję tego jak wiele spośród rekomendowanych obiektów, zostanie rzeczywiście ocenione pozytywnie przez użytkownika, dla którego rekomendacje były przeznaczone. Wymaga to jednak zdefiniowania pewnej funkcji progu , określającej minimalną wartość dla której ocena może zostać uznana za pozytywną. Metryka dla pojedynczego użytkownika   
i zbiorów , , uporządkowanych rosnąco względem wartości drugiego z nich, wyrażona jest wzorem [20]:

|  |  |  |
| --- | --- | --- |
|  |  | (38) |

Czułość od k (recall at k, ) to bliźniacza metryka dla wyżej wymienionej. Bada ona średni stosunek liczby obiektów które są odbierane pozytywnie i zostały zarekomendowane jako   
k najlepszych do liczby wszystkich odbieranych pozytywnie przez użytkownika (39) [20].

|  |  |  |
| --- | --- | --- |
|  |  | (39) |

Miara ta często jednak nie jest stosowana ze względu na to, że zbiór pozytywnie odbieranych obiektów jest zbyt liczny i istotne jest polecenie jedynie jego podzbioru.

**Średnia precyzja**Średnia precyzja (average precision, ) jest to miara określana jako pole powierzchni pod krzywą zależności precyzji od czułości. Jej wartość określa się jako [19]:

|  |  |  |
| --- | --- | --- |
|  |  | (40) |

gdzie jest to zmiana czułości między i jej wartością dla zbioru bez ostatniego elementu (k-1 pierwszych rekomendacji).

Dla 5 ocen pozytywnych (oznaczonych przez +) i 3 negatywnych (oznaczonych jako -) poniższa tabela przedstawia sposób w jaki wyliczana jest wartość .

Tab. 5 Przykład wyznaczania wartości średniej precyzji

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| ocena | + | + | - | + | - | - | + | + |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  | 0 |  | 0 | 0 |  |  |
|  | 0.200 | 0.400 | 0.400 | 0.550 | 0.550 | 0.550 | 0.664 | 0.789 |

Funkcja ta w odróżnieniu od uwzględnia nie tylko jakość zwróconej rekomendacji, ale   
i sposób uporządkowania obiektów w niej zawartych. Przykładowo, przy zamianie kolejności rekomendacji trzeciego i czwartego obiektu, wartość wzrosłaby.

**Przyrost wartości z dyskontem**Przyrost wartości z dyskontem (discounted cumulative gain, ) jest to miara, która łączy dwa aspekty ocen: ich wartości oraz kolejność w rekomendacji. Wyraża się ona wzorem [19]:

|  |  |  |
| --- | --- | --- |
|  |  | (41) |

gdzie jest to ocena -tego w kolejności obiektu, rekomendowanego użytkownikowi. Ocena obiektów ma proporcjonalny do swojej wartości wpływ na ocenę, natomiast jego umiejscowienie na liście obniża ten efekt.

Metryka ta nie jest jednak ograniczona z góry i jej wartość nie może zostać porównana między użytkownikami. Z tego też względu dokonuje się jej normalizacji do metryki względem wartości , równej dla idealnego uporządkowania obiektów (tzn. porządku zgodnego z prawdziwą wartością ocen):

|  |  |  |
| --- | --- | --- |
|  |  | (42) |

# Algorytm ALS z wykorzystaniem treści

Algorytmy filtracji zbiorowej nie mogą zostać zastosowane w wypadku, gdy nie są znane dane historyczne związane z nowym użytkownikiem lub obiektem. Z drugiej strony, systemy oparte na treści wymagają specjalnie zebranych i przygotowanych do ich użycia informacji.

Może się jednak zdarzyć, że oba typy danych nie są wystarczające. Przykładem tego mógłby być film oceniony przez zaledwie kilku użytkowników, dla którego znany jest jedynie jego gatunek i reżyser. Oba systemy jak i wymienione wcześniej podejścia hybrydowe mogą nie być wystarczające aby dopasować rekomendację tego filmu do użytkowników. Nasuwa to jednak pomysł zastosowania obu typów informacji w jednym algorytmie rekomendacyjnym i próbę ich wzajemnego uzupełnienia w celu poprawy rekomendacji. Może to również pozwolić na rekomendację w wypadku gdy znane są tylko informacje z jednego źródła (treści albo historii aktywności).

## Hybrydowy rozkład macierzy

Proponowane podejście opiera się na założeniu, że cechy ukryte, wyliczane z użyciem algorytmów rozkładu macierzy, opisanych w paragrafach 2.1.7 i 2.1.8, mogą być traktowane na równi z jawnie zadeklarowanymi cechami obiektów bądź użytkowników. Jednak sposób trenowania modelu w algorytmie Funk SVD jak i sam rozkład na wartości osobliwe zakłada, że cechy te są uporządkowane ze względu na wpływ jaki mają na ocenę. W wypadku gdy znany jest profil użytkownika lub opis przedmiotu nie jest zawsze oczywiste, która z cech jest istotniejsza dla rekomendacji. Z tego względu, jako podstawa do rozważań, wybrany został algorytm oparty o metodę naprzemiennych, najmniejszych kwadratów oraz oszacowanie bazowe z paragrafu 2.1.6. Podejście to można przypisać do zbioru systemów hybrydowych, łączących źródła, opisanych w paragrafie 2.3.5.

Algorytm ten, jak i inne algorytmy rozkładu macierzy, nie mogą zostać użyte do przewidzenia upodobań użytkownika, przed wystawieniem przez niego kilku pierwszych ocen ani do uzasadnienia samej rekomendacji. Cechy ukryte opisują pewne prawidłowości w zachowaniu użytkowników, nie są jednak powiązane z ich jawnymi odpowiednikami.   
Z tego względu, w dalszej części pracy, sprawdzony został wpływ jawnie podanych cech na przewidywaną wartość oceny oraz zdolność przewidzenia gustu użytkownika na podstawie samego profilu gdy nie znana jest żadna z wystawionych przez niego ocen.

Tak jak w wypadku algorytmu ALS, danymi wejściowymi są oceny pomniejszone   
o wartości wyliczone z oszacowania bazowego. Proponowana zmiana w algorytmie polega jednak na innym sposobie wypełnienia i uaktualniania macierzy cech X i Y. Macierze te są podzielone na 3 typy kolumn: , , takie, że i ,  
gdzie oznacza operator rozszerzenia macierzy. Macierze , , oraz zostają uzupełnione niewielkimi wartościami losowymi, natomiast macierze oraz kolejno cechami użytkowników oraz obiektów. Każda kolumna odpowiada w takim wypadku pojedynczej cesze. Wartości macierzy oraz nie będą uaktualniane w trakcie działania algorytmu.

Rozmieszczenie znanych cech jako początkowych i końcowych kolumn macierzy jest jednym z kroków optymalizacyjnych, przy indeksowaniu macierzy, w trakcie działania algorytmu.   
W pojedynczym kroku, aktualizowana jest wartość macierzy   
lub (w dalszej części pracy, macierze rozszerzone będą oznaczane w ten sposób).

Cały proces trenowania może zostać opisany poniższym pseudokodem.

Content Boosted ALS

Wypełnij macierze , losowymi wartościami z rozkładu jednorodnego

Wypełnij macierze , cechami użytkowników i przedmiotów

Powtarzaj do osiągnięcia zbieżności:

Dla każdego użytkownika :

|  |  |  |
| --- | --- | --- |
|  |  | (43) |

Dla każdego obiektu :

|  |  |  |
| --- | --- | --- |
|  |  | (44) |

W celu uzyskania jak najlepszej wydajności algorytmu, macierz nie jest przedstawiona   
w postaci gęstej. W jej miejsce podstawiana jest macierz rzadka , podczas etapu aktualizacji cech użytkowników oraz macierz , podczas aktualizacji cech obiektów. Podstawienie to zmniejsza ilość pamięci wymaganej do działania algorytmu. W wypadku macierzy ocen, dodatkowy narzut pamięciowy, wynikający z użycia postaci rzadkiej, jest zaniedbywalnie mały w porównaniu do pamięci nieużytej do przechowania wartości zerowych.

Użycie macierzy rzadkich i zamiast pojedynczej macierzy rzadkiej (niezależnie od jej postaci) ma na celu usprawnienie dostępu do pamięci. W wypadku aktualizacji cech użytkownika, potrzebne w tym celu wartości z macierzy R należą do jednego z jej wierszy. Natomiast w wypadku aktualizacji cech obiektu, interesujące wartości macierzy należą do jednej z jej kolumn. Odwoływanie się do kolejnych wartości w kolumnie macierzy jest mniej efektywne niż to samo działanie w wypadku kolejnych wartości w jej wierszu. Narzut obliczeniowy może zostać zrównoważony poprzez stworzenie kopii transponowanej macierzy , na samym początku działania algorytmu. Wymaga to podwojenia użytej pamięci, jednak przy uwzględnieniu liczby operacji, przeprowadzanych z użyciem wartości z macierzy R, jest to korzystny kompromis.

Jako że kolejne kroki algorytmu korzystają z wartości pojedynczych wierszy macierzy R, naturalnym wyborem jej typu jest format skompresowany wierszowo (CSR, compressed sparse row). Dla macierzy gęstej , jej odpowiednik w postaci CSR składa się z 3 wektorów , i . Wektor zawiera niezerowe wartości macierzy , natomiast odpowiadające im indeksy kolumn w kolejności ich występowania dla porządku leksykograficznego: wiersz, kolumna. Wektor składa się natomiast z wartości, spełniających równanie (45).

|  |  |  |
| --- | --- | --- |
|  |  | (45) |

## Użyte narzędzia

Implementacja oraz weryfikacja algorytmów rekomendacyjnych wymaga użycia szeregu instrumentów matematycznych. Idealnym wyborem dla tego celu jest wykorzystanie, rozbudowanego środowiska, języka Python. W pracach użyta została jego wersja 3.7, wraz   
z wspierającymi go środowiskiem VSCode, interaktywnym notatnikiem Jupyter oraz systemem kontroli wersji GIT.

Wszystkie obliczenia zostały przeprowadzone na komputerze z procesorem Intel Core i7, 2.40GHz z 8 GB pamięci RAM.

### Anaconda

Jest to dystrybucja języka Python, skierowana przede wszystkim do analityków danych   
i specjalistów z dziedziny uczenia maszynowego [21]. Udostępnia ona ponad 1500 bibliotek oraz własny menadżer zależności Conda. Pozwala on na zarządzanie obecnie zainstalowanymi wersjami języka jak i bibliotek, w oparciu o środowiska wirtualne.

### NumPy i Pandas

Głównym założeniem tych bibliotek jest efektywne przechowywanie i przetwarzania danych. W wypadku biblioteki NumPy są to wielowymiarowe tablice danych, przede wszystkim   
o wartościach liczbowych. Poza strukturami danych, biblioteka ta dostarcza szereg operacji matematycznych, zoptymalizowanych pod kątem obliczeń wektorowych [22].

Biblioteka Pandas jest rozszerzeniem funkcjonalności NumPy o operacje tabularyczne [23].   
Jej użycie pozwala na analizę danych w sposób podobny do znanego z baz opartych na języku SQL. Z tego względu, została użyta do analizy i wstępnego przetworzenia danych.

### Numba

Wiele operacji matematycznych, z wymienionych w poprzednim rozdziale algorytmów, może zostać zoptymalizowana poprzez zrównoleglenie obliczeń. Jednak implementacja języka Python, oparta na języku C/C++, posiada ograniczenie w postaci semafora (global interpreter lock), pozwalającego na jednoczesne istnienie tylko jednego aktywnego wątku. Wpływ tego ograniczenia można jednak zredukować poprzez użycie kompilatora Numba [24]. Pozwala on na kompilację funkcji, napisanych z użyciem większości składowych języka Python oraz biblioteki NumPy, w momencie pierwszego odwołania. Głównym celem takiego zabiegu jest zrównoleglenie działania kodu, z pominięciem wcześniej wymienionego semafora, a nawet przeniesienie obliczeń na procesory graficzne.

### HyperOpt

Kluczowym aspektem pracy nad algorytmami uczenia maszynowego jest dobór hiperparametrów. Ręczna obsługa tego procesu jak i testowanie wszystkich zakładanych kombinacji, z użyciem przeszukiwania kratowego (grid search), mogą być czasochłonne   
i nieefektywne. Z tego powodu, powstają biblioteki takie jak HyperOpt [25], służące do automatyzacji całego procesu. W oparciu o zdefiniowane ograniczenia, co do typu, wartości czy przestrzeni do jakiej należy każdy z parametrów, tworzone są kolejne ich zestawy   
a otrzymany wynik optymalizacji wpływa na dalszy proces ich doboru.

### Surprise

Aby ocenić implementację proponowanego rozwiązania, zostanie ono porównane z już istniejącymi realizacjami algorytmów, wymienionych w rozdziale 2. W tym celu została użyta biblioteka Surprise [26], udostępniająca między innymi algorytmy najbliższego sąsiedztwa, rozkład macierzy oparty na Funk SVD oraz podział zbioru ocen na próbki treningową i testową czy zbiory ocen do weryfikacji krzyżowej.

## Uwagi implementacyjne

Znaczące usprawnienie wydajności algorytmu uzyskane zostało przy użyciu biblioteki Numba. Kluczowy etap aktualizacji macierzy X i Y może być wykonywany jednocześnie dla kilku użytkowników lub obiektów (ale nie użytkownika i obiektu jednocześnie). Modyfikacja niezależnych bloków pamięci pozwala na wykonanie wewnętrznej pętli w sposób równoległy. Efekt ten został uzyskany poprzez użycie poniższej sygnatury funkcji, odpowiadającej za aktualizacje macierzy lub (zależnie od tego, wartość której z nich została podana jako fixed\_factors a która jako updated\_factors).

@staticmethod

@numba.jit(numba.void(numba.float64[::1], numba.int32[::1],

numba.int64, numba.float64,

numba.float64[:,:], numba.float64[:,:],

numba.float64[:,:], numba.float64[:,:]),

nopython=True)

def \_calculation\_step(ratings, indices, indptr,

update\_rank, reg\_loss,

fixed\_factors, updated\_factors,

side\_fixed\_factors, side\_updated\_factors):

W wypadku tej funkcji, użyte zostały dwa dekoratory, poprzedzone znakiem @. Staticmethod tworzy zgodnie ze swoją nazwą metodę statyczną klasy. Jest to sprzeczne   
z celem jej normalnego zastosowania, jednak w wypadku użycia metody instancji, wymagane byłoby przekazanie obiektu klasy jako pierwszego argumentu. Nie jest to możliwe ze względu na wymóg serializacji wszystkich argumentów skompilowanej funkcji w momencie jej wywołania. Użyte w klasie funkcjonalności nie są wspierane przez bibliotekę Numba tak więc próba jej kompilacji spowodowałaby błąd. Z tego względu, stan obiektu przekazywany jest przez argumenty.

Drugi z dekoratorów odpowiada za samą kompilację funkcji \_calculation\_step. Pierwszy   
z jego argumentów mówi o sygnaturze funkcji: jej typie, typie wartości zwracanej i typie argumentów. Wartość nopython oznacza natomiast, że funkcja ta ma zostać skompilowana do kodu natywnego. Decyzja, które fragmenty kodu mają zostać zrównoleglone, podejmowana jest poprzez zastąpienie funkcji range w pętli for … in range(…) przez jej odpowiednik numba.prange. W momencie rozpoczęcia iteracji, dostępna pula wątków zostanie rozdysponowana do niezależnego wykonania kolejnych iteracji pętli.

rows\_num = indptr.shape[0] - 1

for idx in numba.prange(rows\_num): # pylint: disable=not-an-iterable

…

Aktualizacja wartości macierzy i wykonana jest z użyciem poniższego kodu. Został on uogólniony do wyznaczania wartości oraz (updated\_factors[idx, :]) poprzez bliźniaczy ciąg obliczeń odpowiadających równaniom (43) i (44). W wypadku implementacji, uwzględnienie oszacowania bazowego zostało wyłączone do etapu inicjalizacji modelu, aby poprawić jego wydajność.

lambda\_eye = np.eye(update\_rank) \* reg\_loss

rows\_num = indptr.shape[0] - 1

for idx in numba.prange(rows\_num): # pylint: disable=not-an-iterable

row\_indices = indices[indptr[idx]:indptr[idx+1]]

indices\_num = len(row\_indices)

if indices\_num == 0:

continue

MTM = fixed\_factors[row\_indices].T.dot(fixed\_factors[row\_indices])

conv\_ratings = ratings[indptr[idx]:indptr[idx+1]]

conv\_ratings -= side\_updated\_factors[idx,:]\  
 .dot(side\_fixed\_factors[row\_indices, :].T)

updated\_factors[idx, :] = \

solve(MTM + lambda\_eye \* indices\_num,

conv\_ratings.dot(fixed\_factors[row\_indices]))

Przekształcenia te wykonane są na podstawie wybranych parametrów i podmacierzy, zależnie od trybu wykonania (mode), który może odpowiadać aktualizacji macierzy użytkowników (user mode) albo obiektów (item mode).

if mode == 'user':

reg = self.\_user\_reg\_loss

ratings = self.\_ratings\_ui

update\_rank = self.\_item\_profile\_rank + self.\_base\_rank

updated = self.\_user\_factors[:, self.\_user\_profile\_rank:]

updated\_c = self.\_user\_factors[:, :self.\_user\_profile\_rank]

constant = self.\_item\_factors[:, self.\_user\_profile\_rank:]

constant\_c = self.\_item\_factors[:, :self.\_user\_profile\_rank]

else:

...

## Zbiory danych

Badania algorytmu zostały przeprowadzone na dwóch niezależnych zbiorach danych w celu porównania jak algorytm sprawdza się zależnie od domeny i dostępnych dla niego informacji.

### MovieLens

Dane w zbiorze MovieLens [27] zawierają oceny w skali od 1-5, wystawione przez losowych użytkowników, którzy ocenili co najmniej 20 filmów. Zbiór ten jest stale aktualizowany, jednak informacja demograficzne zostały z niego trwale usunięte w jednej z najnowszych wersji.   
Z tego też względu, do testów użyłem starszej wersji ml-100k, zawierającej 100 000 ocen oraz zbiór dodatkowych informacji. Pełen opis zbioru został przedstawiony poniżej:

Profil użytkownika:

* identyfikator
* wiek
* płeć
* zawód
* kod pocztowy\*.

Cechy filmów:

* identyfikator,
* tytuł,
* datę premiery\*,
* odnośnik do profilu na IMDb\*,
* przynależność do gatunków filmowych,
* genom\*\*.

Oceny:

* identyfikator użytkownika,
* identyfikator filmu,
* ocena numeryczna,
* czas wystawienia oceny\*.

\*Informacje niewykorzystane w trakcie prac

\*\*Genom mimo zawierania przydatnych informacji, takich jak aktorzy, nagrody czy tematyka filmu, nie może zostać użyty w procesie uczenia. Częściowo powstał on w oparciu o komplet ocen dostępnych w zbiorze danych co mogłoby spowodować ich „przeciek” ze zbioru testowego do etapu trenowania [28].

Dane te zostały wstępnie przetworzone na cele użycia w algorytmie. Płeć użytkownika została przypisana do jednej z dwóch wartości: -1 oraz 1. Wiek natomiast jako wektor wartości   
„jedna z” (one-hot) dla przedziałów , … oraz wartość ustandaryzowana względem całej populacji.

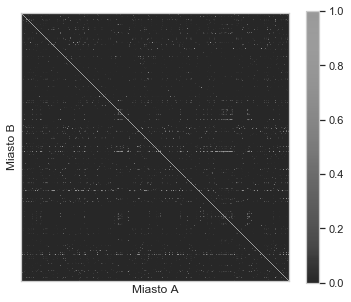
Dodatkowo każdemu zawodowi wykonywanemu przez użytkownika jak i każdemu gatunkowi filmu została przypisana kolumna o wartościach binarnych (w wypadku gatunków, wartość niezerowa dla jednej z cech nie wyklucza jej dla innych).

### Yelp

Zbiór danych Yelp [20] nie posiada jawnie podanej struktury i wymaga dokładniejszej analizy w celu jego wykorzystania. Zawarto w nim strony informacyjne wraz z ocenami   
i recenzjami dotyczącymi firm usługowych z kilku miast. Aby zawęzić dziedzinę jakiej ma dotyczyć system rekomendacyjny, zbiór danych został ograniczony jedynie do restauracji.   
W tym celu wybrałem wszystkie firmy, których co najmniej jedna kategoria działalności znalazła się na liście: jedzenie, restauracje, kawiarnie, piekarnie, bary, puby, food truck, kawa i herbata.

Daje to zbiór 75 447 restauracji, 1 090 542 użytkowników oraz 7 061 578 ocen. Oznacza to, że zaledwie 0.0086% ocen dla każdej możliwej pary użytkownik, restauracja jest znane.   
Tak niska gęstość zbioru ocen stanowi problem, któremu można zaradzić poprzez podzielenie tego zbioru na mniejsze podzbiory o wyższej gęstości.

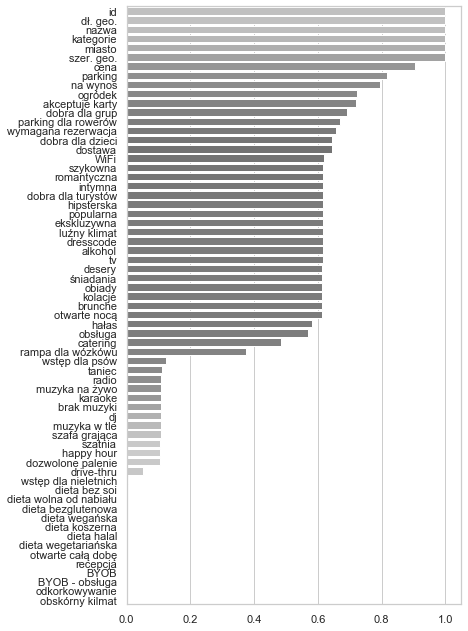
Jednym ze sposobów podziału tego zbioru może być analiza miasta w którym znajduje się każda z restauracji. Poniższy wykres przedstawia jak wielu użytkowników oceniło co najmniej jedną restaurację zarówno w mieście A jak i B względem liczby użytkowników oceniających co najmniej jedno miejsce w mieście B. W większości przypadków, niewielka liczba użytkowników oceniła miejsca w obu miastach, co uzasadnia rozważenie podzbioru tego zbioru danych ograniczonych do jednego wybranego miasta.



Rys. 5 Użytkownicy oceniający restauracje w różnych miastach

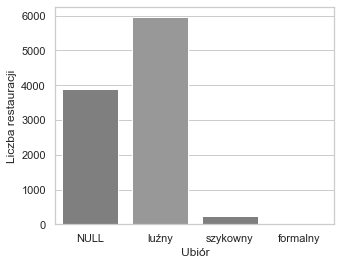
Z tego też względu, dalsza analiza ograniczona jest do restauracji umiejscowionych   
w Toronto, gdzie 85 811 użytkowników wystawiło 800 700 ocen dla 10 100 lokali. Daje to wypełnienie zbioru ocen w 0.09%, dziesięciokrotnie wyższe w porównaniu do zbioru wejściowego.

Każda z restauracji w systemie opisana jest szeregiem cech, których wybór jak i wartość ograniczona jest przez szablon serwisu. Nie wszystkie z tych informacji są jednak na tyle kompletne oraz istotne aby uwzględniać je w procesie trenowania modelu. Przykładem takich cech mogą być informacje co do dostępności posiłków związanych z ograniczeniami żywieniowymi (kuchnia wegańska, bez glutenu, koszerna…). Informacje te są istotne, jednak niski poziom uzupełnienia zbioru danych nie pozwala na ich efektywne użycie. Wykres przedstawiony na rysunku 6 pokazuje jaka część, spośród wszystkich profili restauracji, ma zdefiniowaną dowolną, niepustą wartość dla każdej z wymienionych cech.



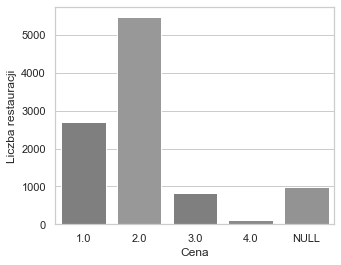
Rys. 6 Stopień wypełnienia cech restauracji

Kolejnym krokiem jest sprawdzenie dystrybucji dla najczęściej podawanych cech. Dla części   
z nich dystrybucja cechy jest zdominowana przez jedną z nich. Przykładem tego może być styl ubioru zalecany w lokalu. W wypadku gdy znana jest informacja na ten temat, większość restauratorów podaje, że nie narzuca wymogów na ubiór. Podobnie jak   
w wypadku cech o niskim poziomie wypełnienia, informacja ta nie wnosi wiele w proces trenowania modelu.



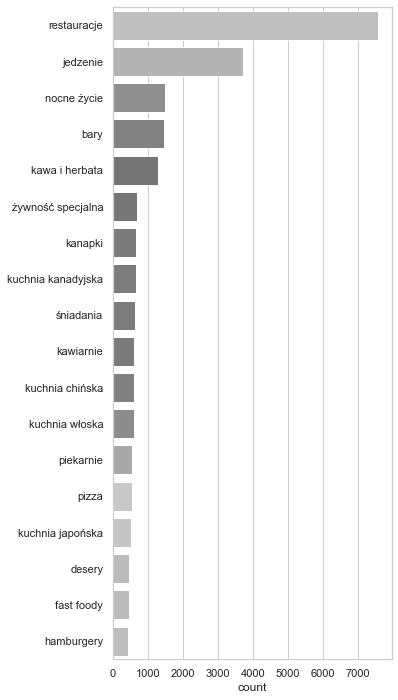
Rys. 7 Zalecany ubiór

Przeciwnym do tego przykładem jest próg cenowy w lokalu. Cecha ta posiada bardziej zrównoważoną dystrybucję. Dla większości restauracji zadeklarowany jest niski (1) albo średni (2) próg cenowy. Z tego powodu można założyć, że wartości nieokreślone należą głównie do jednej z tych dwóch grup. Uwzględniając to, można wprowadzić nową numerację progów cenowych: -1 – niska, 0 – brak informacji, 1 – średnia, 2 – wysoka,   
3 – bardzo wysoka. Ma to swoje zastosowanie w rekomendacjach opartych o proponowany algorytm hybrydowy. Wartość zerowa nie będzie wpływała na ocenę lokalu, -1 pozytywnie wśród osób preferujących tanie lokale, natomiast wartości dodatnie, wśród osób   
o przeciwnych gustach.



Rys. 8 Dystrybucja cen w lokalach

Podobna analiza, do tej dotyczącej cech lokalu, została przeprowadzona również co do przypisania restauracji do poszczególnych kategorii. Część najpopularniejszych z 420 możliwych kategorii, została przedstawiona na rysunku 9. Wszystkie kategorie, poza kategorią restauracje, zostały użyte jako wartości binarne w procesie trenowania algorytmu.



Rys. 9 Liczba restauracji przypisanych do wybranych kategorii

Ostateczny zbiór cech restauracji został przedstawiony poniżej.

* Wartości ze zbioru {0, 1}: akceptuje karty, parking, ogródek, wymagana rezerwacja, obsługa, catering, obiady, lunche, nocne życie, bary, kawa i herbata, żywność specjalna, kanapki, kuchnia kanadyjska, śniadania, kawiarnie, kuchnia chińska, kuchnia włoska, piekarnie, pizza, kuchnia japońska, desery, fast foody, hamburgery.
* Wartości ze zbioru {-1, 0, 1, 2, 3}: cena, głośność.
* Wartości ze zbioru {-1, 0, 1}: wifi, alkohol

W celu przyspieszenia obliczeń, wyznaczony został rdzeń macierzy ocen, rzędu 20. Jest to podzbiór ocen w którym każda restauracja została oceniona przez co najmniej 20 użytkowników jak i każdy użytkownik ocenił co najmniej 20 pozostałych w nim restauracji. Zbiór ten powstał przez naprzemienne odfiltrowywanie użytkowników i restauracji,   
nie spełniających tego wymogu. Zbiór ocen, użyty w procesie trenowania i testowania, został w ten sposób ograniczony do 2 160 użytkowników, 1 993 restauracji i 99 586 ocen, co daje zbiór o gęstości 2.31%.

## Testy algorytmu hybrydowego

Weryfikacja założeń stojących za algorytmem jak i poprawności jego implementacji, odbyła się z użyciem zbiorów danych opisanych w poprzednim podrozdziale oraz algorytmów   
z podrozdziału 2.1. Głównym jej celem było sprawdzenie czy wyniki uzyskane w wypadku ciepłego startu są co najmniej tak dobre jak te uzyskiwane z użyciem popularnych algorytmów oraz to jak zmienia się zdolność algorytmu do rekomendacji w wypadku zimnego startu.

### Procedura eksperymentalna

Weryfikacja algorytmu hybrydowego została przeprowadzona dla dwóch przypadków użycia. Pierwszym z nich jest zagadnienie ciepłego startu, gdy dostępna jest informacja   
o aktywności użytkowników. W tym celu, zbiór ocen został podzielony na 4 równoliczne części, przy jednakowym prawdopodobieństwie przyporządkowania każdej z ocen do dowolnego z nich. Dla tego podziału została dokonana 4-krotna weryfikacja krzyżowa   
z użyciem metryk RMSE, MAE, NDCG@5, NDCG@20 i NDCG@50. Wyniki uzyskane   
z użyciem algorytmu hybrydowego zostały porównane względem otrzymanych przy użyciu algorytmu SVD i k-NN z biblioteki Surprise jak i implemetnacji hybrydowego ALS bez podania informacji o cechach obiektów i użytkowników.

Proces trenowania i doboru hiperparametrów został zautomatyzowany z użyciem biblioteki HyperOpt. Dla każdego algorytmu wykonane zostało 100 iteracji pełnej weryfikacji krzyżowej, w trakcie których minimalizowana była uśredniona po zbiorach testowych wartość błędu średniokwadratowego. Parametry uwzględnione przy optymalizacji jak i zakres oraz rozkład ich wartości zostały przedstawione w tabeli 6.

Zmienna , zdefiniowana została zgodnie z wymaganiami biblioteki HyperOpt jako należąca do zbioru o rozkładzie jednorodnym, koniecznego do użycia, zaimplementowanego w niej, algorytmu optymalizacji. Jednak aby zwiększyć szanse na znalezienie najlepszego rozwiązania, zmienna ta jest w niektórych przypadkach przekształcana z użyciem funkcji nieliniowych, przed przekazaniem do algorytmu. Ma to na celu zredukowanie liczby iteracji poświęconych na sprawdzenie rozwiązań, które teoretycznie powinny zwracać zbliżone do siebie wyniki. Przykładowo, zmiana liczby cech ukrytych z 10 na 15 powinna mieć dużo większy wpływ na uzyskane rezultaty niż zmiana wartości tego parametru z 395 na 400.

Tab. 6 Zakres parametrów użytych w procesie trenowania

|  |  |  |
| --- | --- | --- |
| Algorytm | Hiperparametr | Badane wartości |
| SVD | Liczba cech ukrytych |  |
| Użyto oszacowania bazowego |  |
| Współczynnik uczenia |  |
| Współczynnik regularyzacji |  |
| k-NN | Liczba sąsiadów |  |
| Metryka podobieństwa |  |
| Typ sąsiedztwa |  |
| (Hybrydowy) ALS | Liczba cech ukrytych |  |
| Współczynnik regularyzacji |  |
| Współczynnik tłumienia oszacowania bazowego |  |

Drugim z badanych przypadków jest zjawisko zimnego startu. Klasyczna jego analiza   
i porównanie z wynikami uzyskanymi dla ciepłego startu jest jednak problematyczne. Próba zachowania podziału zbioru ocen do weryfikacji krzyżowej i usunięcia z niego ocen dla użytkowników ze zbioru testowego wiązałaby się z usunięciem wszystkich z nich. Z drugiej strony, wprowadzenie nowego podziału, poprzez przydział wszystkich ocen wybranej części użytkowników do zbioru testowego prowadziłoby do zmiany dystrybucji ocen i cech użytkowników oraz przedmiotów co nie pozwoliłoby na porównanie wyników ze względu na wpływ tych czynników na uzyskane wyniki.

Z tego względu, aby sprawdzić jakość rekomendacji, z modelu wytrenowanego z użyciem parametrów dających najlepsze wyniki, zostały usunięte wartości odpowiedzialne za cechy ukryte (ich wartość została ustalona na wartość średnią cechy). W wypadku symulacji zimnego startu dla obiektów, uśrednione zostało również obciążenie ich ocen oraz cech związanych   
z wpływem profilu użytkownika na odbiór obiektów. Ma to pozwolić na ich pominięcie, jako że zależność ta nie jest znana w przypadku zimnego startu. Analogiczne podejście zostało użyte również w przypadku użytkowników. Tak zmodyfikowany model został zbadany z użyciem weryfikacji krzyżowej z tym samym podziałem co w wypadku przypadku ciepłego startu.   
Ma to pozwolić na jak najlepsze porównanie wyników, na tym sam zbiorze i tą samą metodą co w wypadku ciepłego startu. Podejście to nie jest przypadkiem analizy prawdziwego zimnego startu (do trenowania modelu użyte są oceny każdego użytkownika i obiektu jak w przypadku ciepłego startu). Ma ono jednak na celu pewną przybliżoną jego symulację. Przykładowo, dla użytkowników nie jest znana ich preferencja wobec cech przedmiotów jak i powiązane cechy ukryte, jednak wpływ cech użytkowników na odbiór przedmiotów został wyznaczony   
w oparciu na wcześniejszych ocenach, znajdujących się w systemie.

### Uzyskane wyniki – przypadek ciepłego startu

Najlepszy wyniki, każdego z algorytmów, zostały przedstawione w tabeli 7 (wartości   
w nawiasach odpowiadają odchyleniu standardowemu przy weryfikacji krzyżowej   
a pogrubieniem oznaczony został najlepszy wynik). Wartości hiperparametrów, dla których uzyskane zostały wymienione wyniki, zebrane zostały w tabeli 8.

Tab. 7 Wyniki dla przypadku ciepłego startu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MovieLens | | | | | |
| Algorytm | RMSE | MAE | NDCG@5 | NDCG@20 | NDCG@50 |
| SVD | 0.9169 (0.0023) | 0.7251 (0.0019) | 0.2077 (0.0064) | 0.1007 (0.0030) | 0.1004 (0.0029) |
| k-NN | 0.9380 (0.0030) | 0.7465 (0.0029) | 0.1823 (0.0549) | 0.0513 (0.0057) | 0.0444 (0.0028) |
| ALS | 0.9186 (0.0022) | 0.7234 (0.0016) | 0.2314 (0.0102) | **0.1189** (0.0042) | **0.1227** (0.0030) |
| Hybrydowy ALS | **0.9127** (0.0019) | **0.7187** (0.0012) | **0.2388** (0.0016) | 0.1175 (0.0024) | 0.1195 (0.0026) |
| YELP | | | | | |
| Algorytm | RMSE | MAE | NDCG@5 | NDCG@20 | NDCG@50 |
| SVD | **0.9190** (0.0020) | **0.7217** (0.0017) | **0.2209** (0.0322) | 0.0925 (0.0063) | 0.0740 (0.0036) |
| k-NN | 0.9294 (0.0017) | 0.7318 (0.0014) | 0.1975 (0.0080) | 0.0793 (0.0018) | 0.0630 (0.0014) |
| ALS | 0.9193 (0.0023) | 0.7219 (0.0017) | 0.2143 (0.0172) | **0.0963** (0.0055) | 0.0765 (0.0040) |
| Hybrydowy ALS | 0.9192 (0.0023) | **0.7217** (0.0017) | 0.2124 (0.0166) | 0.0953 (0.0046) | **0.0766** (0.0039) |

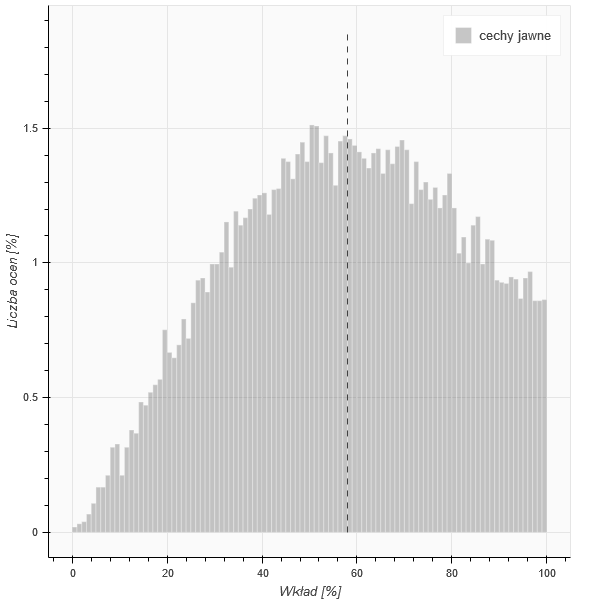
Analizując wyniki, nie jest możliwe wyłonienie najlepszego z algorytmów. Wszystkie trzy, oparte na rozkładzie macierzy, uzyskały podobne rezultaty. Różnica między najlepszym algorytmem a drugim w kolejności nigdy nie przekracza 1% i zazwyczaj mieści się   
w granicach odchylenia standardowego, wartości wybranej metryki. Można domniemać, że uzyskane wyniki wynikają z podobnej zdolności predykcyjnej albo z niedostatecznego przeszukania przestrzeni hiperparametrów jak i stanu początkowego algorytmów.

Tab. 8 Wyznaczone wartości hiperparametrów

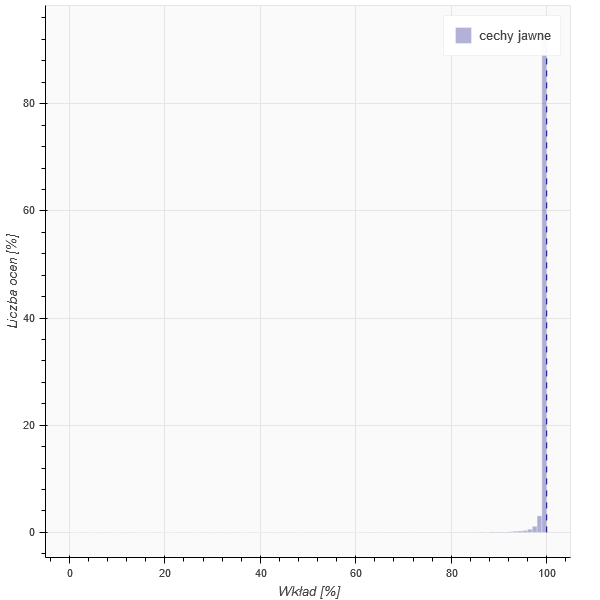
|  |  |  |  |
| --- | --- | --- | --- |
| Algorytm | Hiperparametr | Wyznaczona wartości | |
| MovieLens | YELP |
| SVD | Liczba cech ukrytych | 135 | 16 |
| Użyto oszacowania bazowego | Prawda | Prawda |
| Współczynnik uczenia | 0.0252 | 0.0046 |
| Współczynnik regularyzacji | 0.1056 | 0.1564 |
| k-NN | Liczba sąsiadów | 61 | 99 |
| Metryka podobieństwa | Euklidesowa | Kosinusowa |
| Typ sąsiedztwa | Obiekty | Obiekty |
| ALS | Liczba cech ukrytych | 147 | 89 |
| Współczynnik regularyzacji | 0.1393 | 0.7641 |
| Współczynnik tłumienia oszacowania bazowego | 5 | 6 |
| Hybrydowy ALS | Liczba cech ukrytych | 72 | 19 |
| Współczynnik regularyzacji | 0.1140 | 0.3685 |
| Współczynnik tłumienia oszacowania bazowego | 15 | 6 |

### Wpływ cech jawnych na rekomendację

Liczba cech ukrytych użytych w hybrydowym algorytmie ALS jest mniejsza od wybranej   
dla klasycznej wersji algorytmu, niezależnie od zbioru danych. Może to sugerować, że część   
z nich została zastąpiona poprzez cechy jawne. Dodatkowo, można to argumentować wpływem cech jawnych na ocenę końcową. Wykresy przedstawione na rysunkach 10 oraz 11 przedstawiają procent ocen dla których wartość bezwzględna (wkład) dla podanego typu cechy stanowi daną część całego wkładu cech jawnych i ukrytych w przewidywaną ocenę. Przykładowo dla zbioru MovieLens, około 1.4% uzyskanych ocen było ustalonych w 40%   
w oparciu o cechy ukryte i 60% przez cechy jawne. Liniami przerywanymi oznaczone zostały mediany każdej z dystrybucji. Z obu wykresów wynika, że większy wpływ na oceny miały cechy jawne. Zbliżone wyniki dla algorytmu ALS i jego hybrydowej wersji jak i wpływ cech jawnych na ostateczną ocenę, może być interpretowany jako możliwość wymiennego użycia obu typów cech.



Rys. 10 Wkład cech jawnych do oceny w zbiorze Movielens



Rys. 11 Wkład cech jawnych do oceny w zbiorze YELP

W oparciu o to założenie, został wykonany dodatkowy test w którym liczba cech ukrytych została ustalona na 0 przy zachowaniu pozostałych założeń procedury testowej jak i dla przypadku gdy wyznaczone zostało jedynie oszacowanie bazowe. Wyniki tego testu ujęte zostały w tabeli 9 natomiast parametry algorytmu w tabeli 10.

Tab. 9 Wyniki zastąpienia cech ukrytych cechami jawnymi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MovieLens | | | | | |
| Liczba cech | RMSE | MAE | NDCG@5 | NDCG@20 | NDCG@50 |
| 72 ukryte  18 jawnych | **0.9127** (0.0019) | **0.7187** (0.0012) | **0.2388** (0.0016) | **0.1175** (0.0024) | **0.1195** (0.0026) |
| 0 ukrytych  18 jawnych | 0.9321 (0.0024) | 0.7357 (0.0018) | 0.1934 (0.0172) | 0.0946 (0.0065) | 0.0973 (0.0050) |
| 0 ukrytych  0 jawnych | 0.9430 (0.0020) | 0.7463 (0.0016) | 0.2239 (0.0273) | 0.1114 (0.0048) | 0.1154 (0.0022) |
| YELP | | | | | |
| Liczba cech | RMSE | MAE | NDCG@5 | NDCG@20 | NDCG@50 |
| 19 ukrytych  30 jawnych | **0.9192** (0.0023) | **0.7217** (0.0017) | 0.2124 (0.0166) | 0.0953 (0.0046) | 0.0766 (0.0039) |
| 0 ukrytych  30 jawnych | **0.9192** (0.0023) | **0.7217** (0.0017) | 0.2152 (0.0162) | 0.0963 (0.0049) | 0.0767 (0.0037) |
| 0 ukrytych  0 jawnych | 0.9219 (0.0023) | 0.7266 (0.0017) | **0.2213** (0.0115) | **0.1045** (0.0029) | **0.0832** (0.0014) |

Zgodnie z przewidywaniami, ograniczenie się do cech jawnych, w procesie trenowania modelu, pogorszyło uzyskane wyniki. O ile w wypadku zbioru MovieLens, różnica ta jest widoczna   
o tyle dla zbioru YELP jest ona zaniedbywalnie mała. Takie zachowanie obu modeli można uznać za skutek mocy predykcyjnej cech jawnych w obu zbiorach danych, gdzie w zbiorze MovieLens gdzie mediana ich wkładu w ocenę wynosiła 60% w odniesieniu do 100% dla zbioru YELP.

Interesująca jest wysoka wartość metryk NDCG@k dla oszacowania bazowego. Wynika ona jednak z wady jednego z dwóch znanych sposobów na jej wyliczenie. W wypadku gdy uwzględniane są rekomendacje spoza zbioru testowego, polecenie tych obiektów pogarsza uzyskany wynik (uznawane jest, że ocena dla nich wynosi 0). Teoretycznie obiekty te mogłyby być odebrane pozytywnie przez użytkownika jednak w momencie trenowania modelu nie dysponujemy tą wiedzą. Oszacowanie bazowe poleca obiekty dobrze odbierane przez populację. Ponieważ są one równocześnie popularne, zwiększa to szansę na wybranie pozytywnie odbieranego obiektu. Podejście to nie uwzględnia jednak w żadnym stopniu preferencji użytkownika.

Tab. 10 Wyznaczone wartości hiperparametrów

|  |  |  |  |
| --- | --- | --- | --- |
| Liczba cech | Hiperparametr | Wyznaczona wartość | |
| MovieLens | YELP |
| 72 ukryte  18 jawnych | Współczynnik regularyzacji | 0.1140 | 0.3685 |
| Współczynnik tłumienia oszacowania bazowego | 15 | 6 |
| 0 ukrytych  18 jawnych | Współczynnik regularyzacji | 0.1373 | 0.3592 |
| Współczynnik tłumienia oszacowania bazowego | 6 | 6 |
| 0 ukrytych  0 jawnych | Współczynnik regularyzacji | 0.0000 | 0.0000 |
| Współczynnik tłumienia oszacowania bazowego | 3 | 13 |

Jedną z zalet wpływu cech jawnych na rekomendację jest możliwość uzasadnienia oceny, która w przetworzonej postaci mogłaby zostać zaprezentowana użytkownikowi. Przykładowy wynik takiej analizy dla filmu „Faceci w czerni” i 50-letniego programisty został zaprezentowany   
w tabeli 11.

Tab. 11 Przykładowy wpływ cech na ocenę

|  |  |  |  |
| --- | --- | --- | --- |
| Cecha | Wartość | Cecha | Wartość |
| Średnia ocena | 3.53 | Gatunek: akcja | -0.01 |
| Obciążenie filmu | 0.16 | Gatunek: przygoda | 0.04 |
| Obciążenie użytkownika | -0.37 | Gatunek: komedia | -0.07 |
| Płeć: mężczyzna | 0.07 | Gatunek: sci-fi | -0.10 |
| Zawód: programista | 0.05 | Cechy ukryte | -0.13 |
| Wiek: 50 | -0.01 |  |  |

### Uzyskane wyniki – przypadek zimnego startu

Algorytm został zweryfikowany zgodnie z procedurą eksperymentalną opisanej   
w paragrafie 3.5.1. W wypadku zbioru MovieLens możliwe było zbadanie zarówno przypadku nowych użytkowników jak i obiektów, natomiast dla zbioru YELP – jedynie zimnego startu dla obiektów. Wyniki tego eksperymentu zebrane zostały w tabeli 12.

Uzyskanie wyższej wartości błędu w wypadku metryk RMSE i MAE jest zgodna   
z oczekiwaniem, ze względu na wykorzystanie tylko części z dostępnych informacji. Minimalnie wyższa wartość NDCG@20 i NDCG@50 dla zbioru MovieLens w wypadku nowych użytkowników jest jednak zastanawiająca. Może ona wynikać, tak jak w wypadku oszacowania bazowego, z większego nacisku kładzionego na pozytywny odbiór najpopularniejszych obiektów w populacji, przeuczenia modelu lub szumu. Usunięcie wartości cech ukrytych i części cech jawnych mogło skutkować, w drugim wypadku, uogólnieniem modelu. Mimo tych niejednoznaczności, przy wyborze metryki NDCG@k można założyć, że zaproponowany algorytm mógłby sprawdzić się w przypadku zimnego startu. Rekomendacje   
o wartościach metryk zbliżonych do tych uzyskanych dla ciepłego startu są znaczącym usprawnieniem w porównaniu do braku jakiejkolwiek rekomendacji z klasycznych algorytmów filtracji zbiorowej.

Tab. 12 Wyniki dla przypadku zimnego startu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MovieLens | | | | | |
| Przypadek testowy | RMSE | MAE | NDCG@5 | NDCG@20 | NDCG@50 |
| Ciepły start | **0.9127** (0.0019) | **0.7187** (0.0012) | **0.2388** (0.0016) | 0.1175 (0.0024) | 0.1195 (0.0026) |
| Zimny start (obiekty) | 1.1151 (0.0057) | 0.9323 (0.0048) | 0.1923 (0.0120) | 0.0860 (0.0027) | 0.0770 (0.0018) |
| Zimny start (użytkownicy) | 0.9421 (0.0042) | 0.7474 (0.0037) | 0.2215 (0.0014) | **0.1205** (0.0048) | **0.1221** (0.0026) |
| YELP | | | | | |
| Przypadek testowy | RMSE | MAE | NDCG@5 | NDCG@20 | NDCG@50 |
| Ciepły start | **0.9192** (0.0023) | **0.7217** (0.0017) | **0.2124** (0.0166) | **0.0953** (0.0046) | **0.0766** (0.0039) |
| Zimny start (obiekty) | **0.9192** (0.0023) | **0.7217** (0.0017) | **0.2124** (0.0166) | **0.0953** (0.0046) | **0.0766** (0.0039) |

# Podsumowanie

Efektem pracy było stworzenie hybrydowego algorytmu rekomendacyjnego, którego działanie pozwoliło na uzyskanie wyników porównywalnych z otrzymanymi przy użyciu najpopularniejszych algorytmy filtracji zbiorowej. Ma on jednak przewagę nad takimi rozwiązaniami w warunkach zimnego startu (dla nowych użytkowników i obiektów), którego scenariusz nie jest wspierany przez powyższą grupę algorytmów. Uzyskane, w tym wypadku, wyniki sugerują zasadności założenia, że cechy rekomendowanych obiektów jak   
i użytkowników mogą zostać użyte w procesie rekomendacji na równi z cechami ukrytymi   
i przyczynić się do rozszerzenia zastosowań algorytmu ALS o przypadek zimnego startu.

Dodatkowo, wszystkie rekomendacje, otrzymane z użyciem proponowanego podejścia, mogły zostać, w pewnym stopniu, uzasadnione poprzez wyznaczenie wpływu każdej z cech jawnych na przewidywaną ocenę. W wypadku rekomendacji może to podnieść jej wiarygodność   
w oczach użytkownika, co nie jest możliwe dla algorytmów filtracji zbiorowej.

Korzyści te uzyskane są jednak kosztem wyższej złożoności obliczeniowej i poziomu skomplikowania algorytmu. Jest to jeden z możliwych kompromisów pomiędzy możliwościami algorytmów rekomendacyjnych a problematycznością ich użycia. Proponowane podejście jest dzięki temu rozwiązaniem pośrednim pomiędzy algorytmami dostarczanymi przez biblioteki rekomendacyjne a wysoce wyspecjalizowanymi rozwiązaniami z najnowszych prac badawczych.

Dalsza weryfikacja wpływu dodatkowych informacji na algorytmy filtracji zbiorowej może być uzasadnieniem dla kontynuacji prac nad pochodną opisanego algorytmu. Część informacji, która może wpływać na oceny, nie została użyta w tej pracy lub nie mogła być uzyskana   
z dostępnych publicznie zbiorów danych. Proces rekomendacji może zostać wzbogacony   
o informacje takie jak kontekst podejmowanej decyzji (porę dnia, pogodę, osoby towarzyszące…) czy zmianę upodobań w czasie. Dodatkowo w pracy nie została zbadana zależność między cechami obiektów a cechami użytkowników, które mogą na przykład zawężać grupę docelową dla obiektów o wybranym zbiorze cech poprzez pozytywny odbiór lub brak zainteresowania.

Istotnym dla dalszych prac może się, w takim przypadku, okazać próba zebrania zbioru danych, który dostarczy większości potrzebnych informacji jak i systemu służącego do weryfikacji działania algorytmu w oparciu o pętlę zwrotną, wynikającą z wpływu rekomendacji na zachowanie użytkownika. Wymagałoby to jednak zbudowania rozwiązania w którym,   
w sposób ciągły, badana byłaby reakcja użytkownika na otrzymywane rekomendacje. Jest to czasochłonny proces ze względu na konieczność implementacji rozwiązania jak i czasu potrzebnego na zebranie danych od prawdziwych użytkowników. Kluczowym byłoby również uwzględnienie regulacji prawnych, dotyczących zarządzania danymi osobowymi. Powyższy proces stanowi na tyle rozległe zagadnienie, że nie mogło ono zostać objęte w ramy pracy inżynierskiej.

# Bibliografia

|  |  |
| --- | --- |
| [1] | Koren Y., Bell R., Volinsky C.: Matrix factorization techniques for recommender systems  w *Computer 2019 (8)* s. 30-37 DOI : 10.1109/MC.2009.263 |
| [2] | Koren Y.: The BellKor solution to the Netflix Grand Prize [online]. Dostępny w internecie: https://www.netflixprize.com/assets/GrandPrize2009\_BPC\_BellKor.pdf |
| [3] | Herlocker J., Konstan A. J., Riedl J.: An empirical analysis of design choices  in neighborhood-based collaborative filtering algorithms w *Information Retrieval 2002 (5)*  s. 287–310 DOI: 10.1145/1020443909834 |
| [4] | Resnick P., Iacovou N., Suchak M., Bergstrom P, Riedl J.: GroupLens: An open architecture for collaborative filtering of netnews w *Proceedings of the 1994 ACM conference on Computer supported cooperative work* s. 175-186  DOI: 10.1145/192844.192905 |
| [5] | Agarwal A., Chauhan M.: Similarity measures used in recommender systems: a study [online]. [dostęp: 10.06.2019]. Dostępny w internecie: https://pdfs.semanticscholar.org/943a/e455fafc3d36ae4ce68f1a60ae4f85623e2a.pdf |
| [6] | Frederickson B.: Why you can't dislike something on Facebook [online].  [dostęp: 10.06.2019]. Dostępny w internecie: https://www.benfrederickson.com/rating-set-distributions/ |
| [7] | Linden G., Smith B., York J.: Amazon.com recommendations item-to-item collaborative filtering w *IEEE Internet Computing 2003 (1)* s. 76-80 DOI: 10.1109/MIC.2003.1167344 |
| [8] | Bell R., Koren Y., Volinsky C.: Modeling relationships at multiple scales to improve accuracy of large recommender systems w *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining 2007* s. 95-104  DOI: 10.1145/1281192.1281206 |
| [9] | Eckart C., Young G.: The approximation of one matrix by another of lower rank  w *Psychometrika*, 1936 (1) s. 211–218 DOI:10.1007/BF02288367 |
| [10] | Koren Y.: Factor in the neighbors: scalable and accurate collaborative filtering  w *ACM Transactions on Knowledge Discovery from Data 2010 (1)* s. 1:1-1:24  DOI: 10.1145/1644873.1644874 |
| [11] | Funk S.: Netflix update: try this at home [online]. [dostęp: 10.06.2019]. Dostępny w internecie: https://sifter.org/simon/journal/20061211.html |
| [12] | Bell R., Koren Y.: Scalable collaborative filtering with jointly derived neighbourhood interpolation weights w *Seventh IEEE International Conference on Data Mining 2007* DOI: 10.1109/ICDM.2007.90 |
| [13] | Biblioteka Implicit [online]. [dostęp: 10.06.2019]. Dostępny w internecie: https://github.com/benfred/implicit |
| [14] | Dokumentacja algorytmu ALS w bibliotece Spark [online]. [dostęp: 10.06.2019]. Dostępny w internecie: https://spark.apache.org/docs/2.2.0/ml-collaborative-filtering.html |
| [15] | Burke R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction w *User Modeling and User-Adapted Interaction 2002*  DOI: 10.1023/A:1021240730564. |

|  |  |
| --- | --- |
| [16] | Gunawardana A., Shani G.: A survey of accuracy evaluation metrics of recommendation tasks w *The Journal of Machine Learning Research 2009* s. 2935-2962 DOI: 10.1145/1577069.1755883 |
| [17] | Cremonesi P., Turrin R., Lentini E., Metteucci M.: An evaluation methodology for collaborative recommender systems w *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution 2008* DOI: 10.1109/AXMEDIS.2008.13 |
| [18] | Botev Z., Ridder A.: Variance Reduction w *Wiley StatsRef: Statistics Reference Online 2017*  DOI: 10.1002/9781118445112.stat07975 |
| [19] | Knijnenburg B.: Evaluating recommender systems [online]. [dostęp: 10.06.2019].  Dostępny w internecie: https://www.math.uci.edu/icamp/courses/math77b/lecture\_12w/pdfs/Chapter%2007%20-%20Evaluating%20recommender%20systems.pdf |
| [20] | Manning Ch., Raghavan P., Schütze H.: Introduction to information retrieval s. 151-175 Cambridge University Press 2008 ISBN: 0521865719 |
| [21] | Anaconda Software Distribution. Computer software. Ver. 2-2.4.0. Anaconda, Nov. 2016, https://anaconda.com |
| [22] | van der Walt S., Colbert Ch., Varoquaux G.: The NumPy array: a structure for efficient numerical computation w *Computing in Science & Engineering* *2011 (3)* s. 22-30, DOI:10.1109/MCSE.2011.37 |
| [23] | McKinney W.: Data structures for statistical computing in Python w *Proceedings of the 9th Python in Science Conference 2010* s. 51-56 |
| [24] | Lam S. K., Pitrou A., Seibert S.: Numba: a LLVM-based Python JIT compiler  w *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC (LLVM '15)* s. 7:1-7:6  DOI: 10.1145/2833157.2833162 |
| [25] | Bergstra J., Yamins D., Cox D. D. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures w *Proc. of the 30th International Conference on Machine Learning (ICML 2013) (28)*. |
| [26] | Nicolas H., Surprise, a Python library for recommender systems [online].  [dostęp: 10.06.2019]. Dostępny w internecie: http://surpriselib.com |
| [27] | Harper F., Konstan J.: The MovieLens Datasets: History and Context w *ACM Transactions on Interactive Intelligent Systems (TiiS) 2015 (5)* s. 19 DOI: 10.1145/2827872 |
| [28] | Vig J., Sen S., Riedl J.: The Tag Genome: Encoding community knowledge to support novel interaction [online]. [dostęp: 10.06.2019].  Dostępny w internecie: http://files.grouplens.org/papers/tag\_genome.pdf |
| [29] | Zbiór danych Yelp [online]. [dostęp: 10.06.2019].  Dostępny w internecie: https://www.yelp.com/dataset/challenge |

# Wykaz symboli i skrótów

## Symbole

– użytkownicy

– obiekty

– liczba użytkowników oraz obiektów w systemie

– ocena wystawiona przez użytkownika dla obiektu

– ocena przewidywana przez system dla użytkownika i obiektu

– zbiór ocen wystawionych przez użytkownika oraz zbiór ocen obiektu

– średnia ocen w systemie

– zbiór użytkowników

– zbiór użytkowników, którzy ocenili obiekt

– zbiór użytkowników, którzy ocenili zarówno obiekt jak i

– zbiór obiektów

– zbiór obiektów ocenionych przez użytkownika

– zbiór obiektów ocenionych zarówno przez użytkownika jak i

– zbiór najbardziej podobnych użytkowników do użytkownika

– miara podobieństwa

– indeks Jaccarda

– odległość Euklidesowa

– odległość kosinusowa

– współczynnik korelacji Pearsona

– współczynnik korealcji rang Spearmana

– miara pewności podobieństwa

U, V – macierze ortogonalne

X, Y – macierze cech użytkowników oraz przedmiotów

– norma Frobeniusa

– oszacowanie bazowe oceny użytkownika i przedmiotu

– funkcja kosztu

– współczynnik regularyzacji

– stała uczenia

– macierz rozszerzona

## Skróty

– (z ang. root mean squere error) błąd średniokwadratowy

– (z ang. mean average error) błąd bezwzględny

– (z ang. precision at k) precyzja od k

– (z ang. recall at k) czułość od k

– (z ang. average precision) średnia precyzja

– (z ang. discounted cumulative gain) przyrost wartości z dyskontem

– (z ang. ideal discounted cumulative gain) wartość idealna przyrost wartości   
 z dyskontem

– (z ang. normalized discounted cumulative gain) unormowany przyrost  
 wartości z dyskontem

– (z ang. singular value decomposition) metoda rozkładu na wartości osobliwe

*k-* – (z ang. k-nearest neighbourhood) metoda najbliższego sąsiedztwa

– (z ang. alternate least square) metoda naprzemiennych, najmniejszych  
 kwadratów

# Spis rysunków

Rysunek 1: Preferencje użytkowników.....................................................................................................4

Rysunek 2: Problem nowego użytkownika...............................................................................................4

Rysunek 3: Rekomendacja oparta na treści...............................................................................................5

Rysunek 4: Dystrybucje ocen w kilku z popularnych portali internetowych...........................................10

Rysunek 5: Użytkownicy oceniający restauracje w różnych miastach...................................................30

Rysunek 6: Stopień wypełnienia cech restauracji....................................................................................31

Rysunek 7: Zalecany ubiór......................................................................................................................32

Rysunek 8: Dystrybucja cen w lokalach..................................................................................................32

Rysunek 9: Liczba restauracji przypisanych do wybranych kategorii.....................................................33

Rysunek 10: Wkład cech jawnych do oceny w zbiorze Movielens..........................................................38

Rysunek 11: Wkład cech jawnych do oceny w zbiorze YELP.................................................................38

# Spis tabel

Tabela 1: Zbiór cech wybranych filmów...................................................................................................5

Tabela 2: Przykładowe oceny wystawione przez użytkowników............................................................12

Tabela 3: Metryki podobieństwa między ocenami użytkowników.........................................................12

Tabela 4: Przypadek pesymistyczny dla wyszukiwania sąsiedztwa........................................................13

Tabela 5: Przykład wyznaczania wartości średniej precyzji....................................................................22

Tabela 6: Zakres parametrów użytych w procesie trenowania.................................................................35

Tabela 7: Wyniki dla przypadku ciepłego startu......................................................................................36

Tabela 8: Wyznaczone wartości hiperparametrów..................................................................................37

Tabela 9: Wyniki zastąpienia cech ukrytych cechami jawnymi...............................................................39

Tabela 10: Wyznaczone wartości hiperparametrów................................................................................40

Tabela 11: Przykładowy wpływ cech na ocenę........................................................................................40

Tabela 12: Wyniki dla przypadku zimnego startu....................................................................................41