Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie
Wydział Zarządzania

Studia Stacjonarne
Informatyka i Ekonometria

PRACA DYPLOMOWA
licencjacka

Mikołaj Zapalski

Systemy rekomendacji - analiza i porównanie algorytmów na przykładzie danych filmowych.

Recommender systems - analysis and algorithms comparison based on movie data.

Promotor: dr inż. Bartłomiej Gaweł

“Zatwierdzam do rejestracji i dopuszczam do obrony”

…….
Data i podpis promotora

Kraków, 2020

Spis treści

Streszczenie

Abstrakt

W dobie napływających z każdej strony informacji, ciężko filtrować te treści, które w szczególności nas interesują. Rozwiązaniem są systemy rekomendacji pomagające w doborze odpowiednich pozycji pasujących do naszego gustu. Szereg zastosowań wydaję się być nie ograniczony, począwszy od proponowanych produktów w sklepach internetowych, poprzez ranking wpisów w mediach społecznościowych, a na rekomendacjach filmowych kończąc. Za tymi modelami stoją algorytmy rekomendacji, których działanie zostanie w tej pracy bliżej przedstawione. Opisane zostaną m.in. algorytmy z rodziny najbliższych sąsiadów, techniki filtrowania zbiorowego, metody rozkładu macierzy niepełnych, czy algorytmy wykorzystujące rekurencyjne sieci neuronowe. Dodatkowo przyblizonę będą metodyki ewaluacji tych modeli. Wspomniane teoretyczne zagadnienia zostaną przedstawione w badaniu empirycznym na danych filmowych. Stworzone zostaną różnorakie modele, które następnie będą porównane ze sobą w celu odnalezienia najlepszego rozwiązania oraz finalnie wyciągnięcia wniosków z całości badania.

Abstract

In data-driven world, it is difficult to filter only the content that we are interested in. Solution to this problem are recommendation systems that help in choosing items that suit our taste. A number of applications seems to be unlimited, the systems are used in products recommended in online stores, the ranking of posts on social media platforms and movie recommendations. These models are powered by the recommendations algorithms, which will be explained in this paper. Covered topics are k-nearest neighbors algorithms, collaborative filtering, singular value decomposition methods and recursive neural networks. In addition, methodologies for evaluating these models will be introduced. These theoretical topics, will be built into real recommendation engines based on movie data, which will be later evaluated and compared to each other and finally after that, the conclusions will be presented.

Wprowadzenie

Problem badawczy

W dzisiejszych czasach nie można narzekać na brak dostępnych treści. W internecie w ciągu sekundy powstaje prawie 10 tys. nowych tweetów oraz użytkownicy serwisu YouTube odtwarzają jednocześnie około 85 tys. filmów[1]. Każdego dnia serwis ten odnotowuje 5 miliardów odsłon, a co minutę przybywa średnio 300 godzin materiału[2]. Przy tak dużym natłoku informacji, pojedynczy użytkownik może czuć się przytłaczany treścią, która nie nie wpada w jego krąg zainteresowań. Aby zapobiec takim sytuacjom oraz polepszyć jego wrażenie użytkownika (ang. user experience) stosuje się systemy rekomendacji.

Zadaniem tych systemów jest przefiltrowanie obszernych zasobów oraz zwrócenie uwagi użytkownika ku parunastu wybranym dla niego pozycjom. Dzięki takiemu zabiegowi możemy skierować jego uwagę na produkt, który może go zainteresować, stworzyć spersonalizowaną pod niego playlistę lub zaproponować filmy z jego ulubionego gatunku. Przekrój branży korzystających z tego rodzaju systemów jest bardzo szeroki, proponować również można wyniki wyszukiwania, artykuły prasowe i naukowe, posty w mediach społecznościowych, pozycje z wewnętrznych baz wiedzy w dużych firmach albo nawet innych użytkowników w serwisach randkowych.

Użytkownik po otrzymaniu spersonalizowanej rekomendacji czuje się wyróżniony i zrozumiany, spędza więcej czasu na platformie, co przekłada się na jego wrażenie i zadowolenie z usługi. Badania pokazują, że 80% klientów jest bardziej skłonnych do złożenia zamówienia, jeżeli dana marka oferuje mu produkty pod niego spersonalizowane[3]. Oprócz tego stosowanie takiego modelu, pomaga rozwiązać tzw. problem długiego ogona (w rozkładzie popularności produktu), poprzez przedstawianie mniej popularnych pozycji dostępnych w bazie.

Przykładem, że inwestycja w systemy rekomendacji może przynieść świetne efekty może być znana platforma Netflix. W ubiegłym roku na badania R&D przeznaczyła 38% więcej środków, co przenosi się na 651 milionów dolarów[4]. To około 10% rocznych przychodów firmy. Carlos Uribe-Gomez, wiceprezes ds. strumieniowego przesyłania wideo, i dyrektor ds. produktu Neil Hunt opublikowali artykuł, w którym przedstawiają, że stosowanie algorytmów rekomendacji pozwala im zaoszczędzić ponad **miliard dolarów** rocznie, poprzez zmniejszenie współczynnika osób anulujących subskrypcje[5]. Oprócz tego filmy mniej popularne, często również tańsze, trafiają bezpośrednio do osób które są zainteresowane

---

[1] *Internet live stats, https://www.internetlivestats.com/one-second/,* [dostęp 16.06.2020]

[2] *YouTube Facts, Figures and Statistics – 2020, https://merchdope.com/youtube-stats/,* [dostęp 16.06.2020]

[3] *New Epsilon research indicates 80% of consumers are more likely to make a purchase when brands offer personalized experiences, https://us.epsilon.com/pressroom/new-epsilon-research-indicates-80-of-consumers-are-more-likely-to-make-a-purchase-when-brands-offer-personalized-experiences,* [dostęp 16.06.2020]

[4] *How Netflix's AI Saves It $1 Billion Every Year, https://www.fool.com/investing/2016/06/19/how-netflixs-ai-saves-it-1-billion-every-year.aspx,* [dostęp 14.01.2020]

[5] *The Netflix Recommender System, https://dl.acm.org/doi/pdf/10.1145/2843948,* [dostęp 14.01.2020]

konkretną niszą, przez co nie muszą inwestować w drogie, popularne tytuły aby zasilić nimi platformę.

Cel pracy

Celem tej pracy jest przedstawienie stosowanych metod systemów oferujących sugestie oraz ich ewaluacja na przykładzie danych filmowych. Przedstawione zostaną podejścia klasyczne takie jak metoda K-najbliższych sąsiadów z podziałem na filtrowanie oparte na zawartości i filtrowanie zbiorowe bazujące na użytkownikach i pozycjach. Algorytmy używające faktoryzacji macierzy (inspirowane SVD) oraz te stosujące sztuczne sieci neuronowe, aktualnie wykorzystywane przez największe firmy oprócz tego przedstawione zostaną metody eksperymentalne, jeszcze nie stosowane w praktyce, ale prezentowane w fazach badawczych (takie jak np. metoda *Mise-en-Scene)*. Przybliżone zostaną również metody pomiaru sprawności takich systemów.

Powyższe algorytmy po przedstawieniu zasady działania, posłużą do stworzenia silnika rekomendacji opartego na danych filmowych. Modele zostaną porównane wraz z różnymi ich wariantami stosującymi różne techniki i parametry. Opierając się o wymienione metodyki ewaluacji, zostaną wyłonione najlepsze podejścia, które następnie zostaną omówione, starając się odpowiedzieć na pytanie co było przyczyną ich sukcesu lub porażki konkurentów.

Zawartość

Praca została stworzona w oparciu o źródła naukowe, zarówno te starsze oraz te prezentujące innowacyjne rozwiązania. Oprócz tego zaczerpnięto informacji z książek traktujących na ten temat, artykułów internetowych oraz wiedzy wyciągniętej z kursu o tematyce budowania systemów rekomendacji.

W rozdziale II, przedstawiono kilka najbardziej popularnych podejść do problemu rekomendacji treści. Pierwszym z nich jest filtrowanie oparte na zawartości, które do działania potrzebuje jedynie danych o produktach. Następnie przybliżone zostają warianty filtrowania zbiorowego, czyli podejścia bazującego na użytkownikach i ich zachowaniu względem treści. Wariant oparty na użytkownikach stosuje taktykę znalezienia użytkowników podobnych do samego odbiorcy, a wariant oparty na pozycjach znajduje obiekty podobne do tych które odbiorcy się podobały. Kolejna część poświęcona jest podejściu opartemu o faktoryzację macierzy i próbę predykcji tego jak danemu użytkownikowi przypadnie go gustu konkretny obiekt. To podejście inspirowane jest metodą rozkładu według wartości osobliwych (ang. singular value decomposition - SVD) i zyskało popularność, przy okazji nagrody Netflix'a, gdzie osiągało bardzo wysokie wyniki. Ostatnia omawiana metoda korzysta z rekurencyjnych sieci neuronowych, również próbując przewidzieć jak dane obiekty zostaną odebrane przez użytkownika. Jest to jedna z wielu testowanych obecnie metod wykorzystujących sztuczne sieci neuronowe. Podobne rozwiązanie wykorzystywane jest przez Google do maksymalizacji czasu spędzanego przed filmami w serwisie YouTube.

Ostatnia część rozdziału teoretycznego przedstawia techniki, jakimi mierzone są osiągi systemów sugerujących treści.

W rozdziale III, opisano budowę modeli rekomendacji przy użyciu ww. technik. Przedstawiono jak wygląda zbiór danych oraz kto jest jego autorem. Oprócz tego zaprezentowane zostały narzędzia, którymi się posłużono oraz końcowe wyniki dla różnych wariantów zestawione w tabeli wraz z wnioskami płynącymi z badania empirycznego.

W rozdziale praktycznym nie przedstawiono kodu źródłowego użytego do otrzymania wyników badania, znaleźć go można w załącznikach dołączonych do tej pracy. W suplemencie zamieszczono również grafiki i rezultaty testów w obszerniejszej formie.

1. Algorytmy rekomendacji w literaturze

1.1. Systemy sugerujące treść

Systemy rekomendacji najprościej można przedstawić jako algorytmy, mające na celu przedstawić użytkownikowi aplikacji, pozycje najbardziej dla niego istotne. We współczesnych aplikacjach internetowych wdrożony system rekomendacji może być kluczem do osiągnięcia przez nią sukcesu na rynku, poprzez wyróżnienie się na tle konkurencji innowacyjnością lub zwyczajne zwiększenie obrotów w przedsiębiorstwie[6].

Obserwując największe światowe korporacje, każda z nich w pewnym stopniu stosuje lub inwestuje w rozwój tego typu rozwiązań. Najpowszechniejsze ich wykorzystanie to m.in. sugerowanie wideo i muzyki w serwisach takich jak YouTube, Spotify, czy Netflix, oferowanie podobnych produktów w sklepach internetowych czego przykładem jest Amazon lub sortowanie treści na platformach społecznościowych takich jak Facebook lub Twitter[7].

Do tego tematu rekomendacji można podchodzić od kliku stron np. próbując rozwiązać problem predykcji ocen lub znajdując pozycje jak najbardziej do siebie zbliżone. W poniższych podrozdziałach przedstawione zostanie kilka znanych sposobów wraz z krótkim opisem działania tych algorytmów.

1.2. Filtrowanie oparte na zawartości

Filtrowanie oparte na zawartości (ang. Content Based Filtering) korzysta z atrybutów wybranego obiektu w celu znalezienia innych podobnych do niego treści, bazując na poprzednich akcjach lub zachowaniach użytkownika[8]. Jest to podstawowe i najprostsze podejście do tematu rekomendacji pozycji dla użytkownika.

W przypadku rekomendacji filmowych interesujące są takie informacje jak gatunek, rok produkcji, przyjęcie krytyków lub osoby zaangażowane w powstawanie obrazu. Na podstawie tych danych dla każdego użytkownika tworzony jest osobny profil, który składuje informację z jego preferencjami dotyczącymi np. ulubionych gatunków czy krajów pochodzenia produkcji[9].

Aby zmierzyć jak blisko siebie są dwa obiekty, należy stworzyć macierz podobieństw. W tym celu można wykorzystać np. *one-hot encoding,* czyli informację o gatunkach przedstawić w postaci tabeli, gdzie wierszami będą poszczególne tytuły, a kolumnami wszystkie gatunki występujące w zbiorze danych[10]. Jeżeli dana pozycja należy do gatunku otrzymuje wartość 1, w przeciwnym 0. Załóżmy że chcielibyśmy opisać film animowany *Toy Story (1995)*, w tym przypadku wartość 1 mogłaby wystąpić w kolumnach „Animacja” oraz „Familijny”, kolumny „Horror” lub „Dramat” zawierałby wartość 0. Podejściem

---

[6] *Introduction to recommender systems,* https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada, [dostęp 18.06.2020]

[7] *Recommender system,* https://en.wikipedia.org/wiki/Recommender_system, [dostęp 18.06.2020]

[8] *Content-based Filtering,* https://developers.google.com/machine-learning/recommendation/content-based/basics, [dostęp 25.05.2020]

[9] L. Candillier, K. Jack, F. Fessant, F. Meyer, *State of the Art Recommender System,* 2009

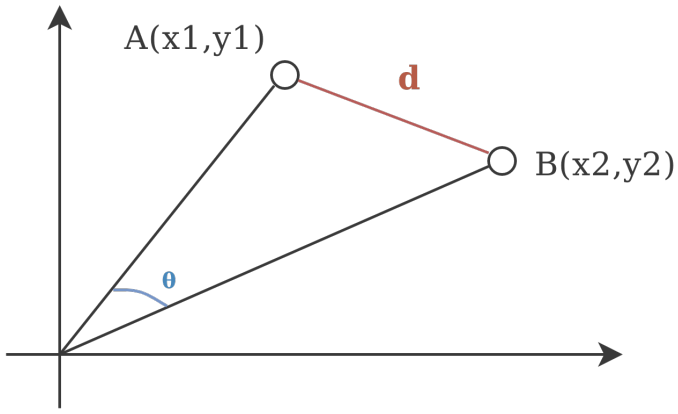
[10] *One-hot encoding,* https://en.wikipedia.org/wiki/One-hot, [dostęp 25.05.2020]

alternatywnym do powyższego, może być posłużenie się wiedzą ekspercką i nadanie poszczególnym wystąpieniom odpowiednich wag w skali np. 0-1. W tym przypadku przykład mógłby wyglądać następująco - „Animacja” - 0.9, „Komedia” - 0.6 oraz „Dokument” - 0. To podeście jednak wymaga o wiele większego zaangażowania w przygotowanie danych, ponieważ gatunek najczęściej zapisuję się w formie binarnej.

Następnie należy obliczyć dla każdej pary miarę podobieństwa, można to zrobić na kilka sposobów np. wartością cosinusa między dwoma obiektami w przestrzeni dwuwymiarowej (θ na Rys. 1) lub zwykłą odległością euklidesową (d na Rys. 1)[11]. Wzory na podobieństwo obliczane za pomocą obu podejść wyglądają następująco:

$$EucSim(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2},$$

$$CosSim(x,y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$$



Rys. 1, Miary podobieństwa przedstawione na wykresie, źródło: https://cmry.github.io/notes/euclidean-v-cosine

Osie wykresu przedstawiają stopień natężenia danej cechy w wybranym obiekcie. Warto zaznaczyć, że liczba unikalnych par wynosi $\frac{n^2 - n}{2}$, gdzie n to liczba wszystkich atrybutów. Dla dużych zbiorów danych, wielokrotne powtarzanie tej operacji np. po dodaniu oceny nowego obiektu, może być czasochłonne i nie dające zbyt wielu nowych informacji.

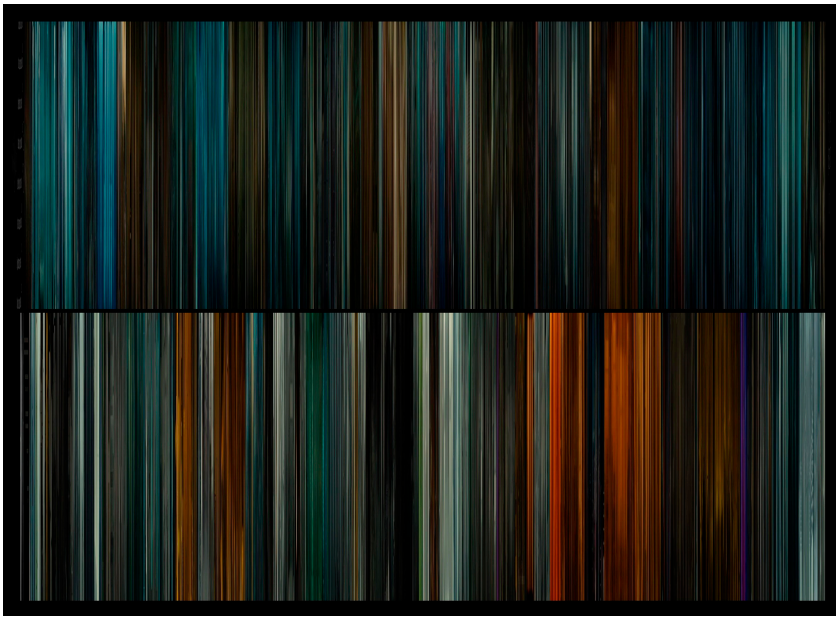
Filmy z lat 80 znacząco różnią się od tych produkowanych w dzisiejszych czasach, dlatego nic nie stoi na przeszkodzie aby wprowadzić inne miary podobieństwa takie jak rok produkcji lub chociażby czas trwania. W doborze funkcji istnieje duża dowolność, można skorzystać np. z przekształcenia funkcji eksponencjalnej lub innych dowolnych dających satysfakcjonujące rezultaty.

Podejście *Mise-en-Scene* zagłębia się jeszcze niżej, badając elementy stylistyczne takie jak obiekty na ekranie i ich ruch, oświetlenie, pracę kamery, montaż czy paletę barw klatek filmowych[12]. Interesujące jest to, że zwiastuny filmowe świetnie sprawdzają się jako

[11] F. Kane, *Building Recommender Systems with Machine Learning and AI*, https://www.udemy.com/course/building-recommender-systems-with-machine-learning-and-ai/, [dostęp 21.05.2020]

[12] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Bakhshandegan-Moghaddam, A. L. E. Caielli, *How to Combine Visual Features with Tags to Improve the Movie Recommendation Accuracy, Springer – Proceedings of the 17h International conference on Electronic Commerce and Web Technologies,* 2016

próbka stylistyki całego filmu[13]. Technika ta jest cały czas udoskonalana, aktualnie tworzone są narzędzia pozwalające w łatwy sposób na wyliczenie stopnia podobieństwa tym sposobem.



Rys. 2, Paleta kolorystyczna filmu *Łowca Androidów (1982)* oraz *Blade Runner 2049 (2017)*, źródło: https://www.reddit.com/r/dataisbeautiful/comments/d8ue6x/inspired_by_recent_blade_runner_barcode_both/

Mając obliczone miary podobieństwa, należy zastosować algorytm K najbliższych sąsiadów. Następnie bazując na stworzonym wcześniej profilu użytkownika, wyniki należy posortować malejąco i odfiltrować pozycje już widziane przez użytkownika. Efektem tej operacji jest lista top-n rekomendacji zawierająca pozycje najbliższe gustowi użytkownika.

Zastosowanie systemu opartego na samych cechach obiektu sprawdza się kiedy chcemy skalować nasz silnik rekomendujący na większą ilość użytkowników. Dane o innych użytkownikach nie są wykorzystywane, a co z tym idzie możliwe jest wychwycenie niszowych zainteresowań konkretnego użytkownika i polecenie mu mało popularnych obiektów.

Jednak do poprawnego działania modelu, potrzebna jest duża ilość konkretnych danych, czasami trudno dostępnych lub czasochłonnych do pozyskania. Dodatkowo proponowane pozycje tworzone są na podstawie zainteresowań użytkownika, co wyklucza proponowanie mu obiektów z którymi jeszcze się nie zapoznał, a mógłby mu się spodobać[14].

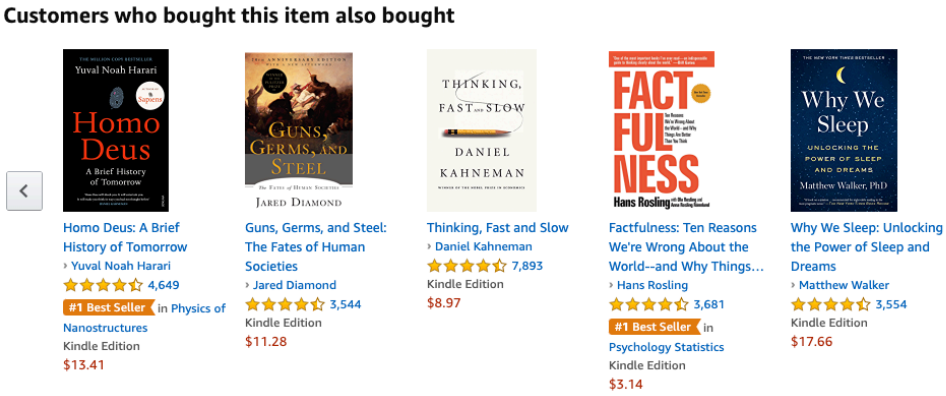
1.3. Filtrowanie grupowe

Celem filtrowania grupowego (ang. Collaborative Filtering) jest znalezienie użytkowników podobnych do odbiorcy lub znalezienie obiektów podobnych do obiektów które użytkownikowi się podobały. To podejście rekomenduje pozycje bazując na danych wielu użytkowników i ich odzewu, nie zważając na zawartość proponowanej treści. Techniki

[13] Y. Deldjoo, F. Garzotto, M. Elahi, P. Piazzolla, P. Cremonesi, *Recommending Movies Based on Mise-en-Scene Design,* 2016

[14] *Content-based Filtering Advantages & Disadvantages, https://developers.google.com/machine-learning/recommendation/content-based/summary,* [dostęp 31.05.2020]

stosowane w tym podejściu są podobne do tych przedstawionych w poprzednim rozdziale, z tą różnica że na wejściu przedstawiane są innego rodzaju dane. Jednym z najbardziej popularnych silników rekomendacji tego typu jest ten stosowany przez sklep internetowy Amazon.



Rys. 3, Przykład Filtrowania Grupowego w praktyce, źródło: https://www.amazon.com/

1.3.1. Oparte na użytkownikach

Podejście bazujące na użytkownikach (ang. user based) zostało zaproponowane pod koniec lat 90 przez Jonathana L. Herlockera z Uniwersytetu w Minnesocie[15]. Chcąc znaleźć innych użytkowników o guście podobnym do odbiorcy, należy posiadać informację o osobach i ocenionych przez nich pozycjach. Oceny użytkowników mogą pochodzić wprost od nich (recenzja filmu na 5 gwiazdek) lub mogą być domniemane (dodanie produktu do koszyka świadczy o tym, że klient jest zainteresowany produktem). W tym podejściu o wiele lepiej sprawdza się pierwszy typ ocen, jednak w publikacji o tym w jaki sposób serwis YouTube rekomenduje treści zawarto informację o tym, że oceny filmów nie są przez nich brane pod uwagę, lecz bazują na czasie spędzonym na oglądaniu filmu i historii wyszukiwania[16]. Jednak stosowane przez nich metody są o wiele bardziej skomplikowane niż filtrowanie grupowe, zostaną przytoczone w kolejnych rozdziałach. Warto zaznaczyć, żeby filtrowanie oparte na użytkownikach dawało konkretne wyniki, użytkowników musi być dużo, albo powinni mieć znaczną cześć wspólnych ocenionych filmów. W przypadku kiedy np. trzech użytkowników nie ma żadnych wspólnych treści, znalezienie podobieństw między nimi nie jest możliwe.

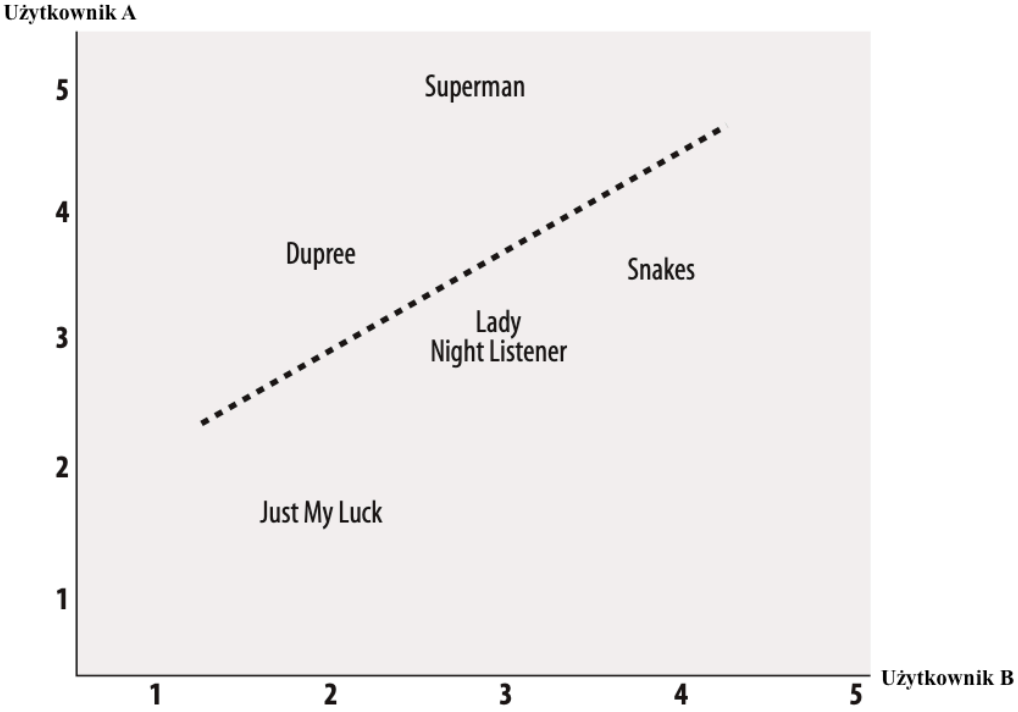
Gdy dysponujemy odpowiednio przygotowanymi danymi, należy określić, w jakim stopniu ludzie są podobni do siebie względem gustów. W tym celu, podobnie jak w przypadku rekomendacji opartej o zawartość, należy wyznaczyć miary podobieństwa pomiędzy wszystkimi użytkownikami[17]. Możemy skorzystać z wielu dostępnych miar tj. przedstawione w poprzednim rozdziale odległość Euklidesowa lub Cosinus między obiektami. Jeszcze jednym sposobem może być obliczenie miary korelacji Persona, czyli zbadać czy istnieje zależność liniowa pomiędzy ocenami pary użytkowników. Zastosowanie

[15] Z. Sun, N. Luo, *A new user-based collaborative filtering algorithm combining data-distribution - in Proceedings of the 2010 International Conference of Information Science and Management Engineering - Volume 02,* 2010

[16] P. Covington, J. Adams, E. Sargin, *Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16),* 2016

[17] T. Segaran, *Programing Collective Intelligence: Building Smart Web 2.0 Applications, O'Reilly 2007, s.29-46*

tej metody daje lepsze wyniki w sytuacji, gdy występuję tzw. *inflacja ocen* (3 gwiazdki użytkownika A nie zawsze są równe 3 gwiazdkom użytkownika B). Miara korelacji nie jest wrażliwa na takie zachowanie i pozwala na wychwycenie dużej wartości miary podobieństwa, nawet jeżeli występuje stała różnica między ocenami.



Rys. 4, Porównanie dwóch użytkowników na wykresie, mniejsza odległość od prostej oznacza większą korelacje,
źródło: Toby Segaran, Programing Collective Intelligence: Building Smart Web 2.0 Applications

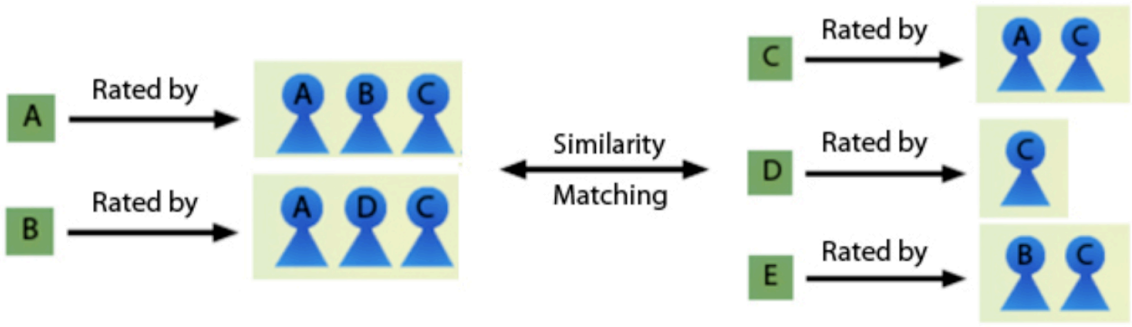
Mając obliczone miary podobieństwa dla wszystkich użytkowników, następnym krokiem jest odnalezienie sąsiedztwa (ang. neighborhood), czyli listę n innych osób o zbliżonym guście. Aby uniknąć sytuacji w której, z powodu małego zbioru użytkowników do sąsiedztwa zaliczane są niechciane elementy, należy dodać próg wejścia (ang. treshhold), który odrzuci użytkowników dopasowanych w niskim stopniu.

Dla każdego „sąsiada" zbieramy listę tytułów przez niego ocenionych, dodajemy je do siebie tworząc ranking z uwzględnieniem wag (np. osoby o większym dopasowaniu do użytkownika końcowego mają większe wagi lub tytuły często powtarzające się również wędrują wyżej). Istnieje wiele możliwości jak zbudować ten ranking na inne sposoby np. dodając karę za niskie oceny. Nie istnieje wyznaczony standard jak dokładnie należy postępować. Należy eksperymentować z różnymi metodami w celu poprawy wyników dla konkretnych danych którymi dysponujemy przy okazji danego projektu.

Z rankingu usuwamy pozycje, które odbiorca już widział, takie które mogą być nieodpowiednie dla jego wieku lub w inny sposób obraźliwe. Najczęściej zostawia się 20 lub 40 pierwszych wyników, wyciągając tym samym listę top-n rekomendacji otrzymanych metodą filtrowania grupowego bazującego na innych użytkownikach (ang. user based).

1.3.2. Oparte na pozycjach

Podejście oparte na pozycjach (ang. item based) również zostało zaproponowane przez badaczy z Uniwersytetu w Minnesocie w 2001 roku[18]. Bazując na fakcie, iż gust użytkowników jest stały lub zmienia się w bardzo powolnym tempie, zaproponowano rozwiązanie budujące sąsiedztwa obiektów na podstawie preferencji użytkowników[19].



Rys. 5, Schemat działania filtrowania grupowego opartego na pozycjach,
źródło: Asanov, D. A.. "Algorithms and Methods in Recommender Systems." (2011).

Chcąc uzyskać listę rekomendacji tym sposobem, należy postępować analogicznie jak w przypadku poprzednim, z tą różnicą aby w miejsce użytkowników wprowadzić pozycje i vice versa. Zazwyczaj w systemach internetowych liczba produktów jest mniejsza i bardziej stała od liczby użytkowników. Oznacza to, że rzadziej należy obliczać macierz podobieństw pomiędzy produktami, ponieważ częściej dochodzą nowi użytkownicy, niż nowe produkty. Dodatkowo czas przeznaczony na te obliczenia jest krótszy z powodu mniejszej macierzy pozycji niż użytkowników i co z tym idzie liczby koniecznych kombinacji. Eliminujemy również problem nowego użytkownika (ang. cold start), czyli sytuację w której nie znając preferencji odbiorcy nie jesteśmy w stanie porównać jego gustu do innych użytkowników.

1.3.3. Wady i zalety obu podejść

Zaletami filtrowania grupowego jest fakt, że jest proste w działaniu i daje widoczne rezultaty. Zaczynając nie musimy mieć specjalnej wiedzy o produkcie, interesujące są jedynie oceny społeczności. Podejście w odróżnieniu od filtrowania opartego na zawartości pozwala na odkrywanie nowych treści o których istnieniu użytkownik mógł wcześniej nie słyszeć.

Niestety taki silnik jest ciężko skalowany, co oznacza że im większa porcja danych na wejściu, tym dłużej należy czekać na rezultaty. W przypadku małych zbiorów nie jest to problemem, lecz dla dużych firm jest to strata mocy obliczeniowej, którą mógłby w tym czasie przeznaczyć na coś bardziej produktywnego. Dodatkowo podejście to wymaga dobrze przygotowanych danych, jest w dużym stopniu wrażliwe na szum i wybrakowanie w danych.

---

[18] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, *Item-based collaborative filtering recommendation algorithms - in Proceedings of the 10th international conference on World Wide Web,* 2001

[19] D. A. Asanov, *Algorithms and Methods in Recommender Systems*, 2011

1.4. Rozkład według wartości osobliwych

Rozkład SVD macierzy (ang. Singular Value Decomposition) nazywany rozkładem według wartości osobliwych to technika stosowana w analizie danych do redukcji wymiarów oraz odkrywania ukrytych w nich zależności. Rozkład SVD macierzy A to przedstawienie jej w postaci:

$$A = U \Sigma V^T,$$

gdzie U i V to macierze składowe zawierające przekształcenia z posiadanych przez nas danych o użytkownikach i ich ocenach, a Σ to macierz diagonalna. Aby uzyskać te macierze należy zastosować techniki faktoryzacji macierzy, czyli zmniejszyć liczbę wymiarów do dwóch, otrzymując trudne do interpretacji przez człowieka, ale jednak niosące w sobie ukrytą informację wartości.

Niestety aby wykorzystać SVD nie może wystąpić sytuacja w której dane są wybrakowane. Początkowo próbowano rozwiązać ten problem uzupełniając braki średnią lub medianą, lecz to wiązało się z olbrzymią liczbą dodatkowych danych i nie poprawiało wyników[20]. Jednak znając techniki wymnażania macierzy, możemy przedstawić wybrane wartości z jednej z nich jako operacje na drugich oraz próbując rozwiązać problem minimalizacji błędów. Stosując technikę stochastycznego obniżania gradientu (ang. stochastic gradient descent) otrzymamy macierz A z wyestymowanymi wartościami mogącymi służyć jako rekomendacje.

1.5. Rekurencyjne sieci neuronowe (RNN)

Zyskujące w ostatnich latach ogromną popularność techniki uczenia maszynowego mają również zastosowanie w systemach rekomendacji. Sieci neuronowe świetnie spisują się przy problemach związanych z rozpoznawaniem wzorów. Przed zastosowaniem w praktyce należy jednak zastanowić się, czy w istocie istnieje potrzeba tworzenia takiego systemu, czy jest to wynik panującej obecnie mody na ten rodzaj technologii. Proponowane do tej pory podejścia pokazały, że stosowanie tego typu rekomendacji lepiej radzi sobie z wybrakowanymi danymi oraz jest w stanie uzykiwać lepsze wyniki od klasycznych podejść[21].

1.5.1. Ograniczona maszyna Boltzmanna (RBM)

Konkurs Neflix'a, w którym uczestnicy mierzyli się aby jak najlepiej zminimalizować miarę RMSE w swoich systemach do rekomendacji zakończył się w 2009 roku. Większość topowych rozwiązań bazowała na SVD, podejściach hybrydowych lub właśnie na maszynach Boltzmanna (ang. Restricted Boltzmann Machines)[22]. To podejście spopularyzowane zostało

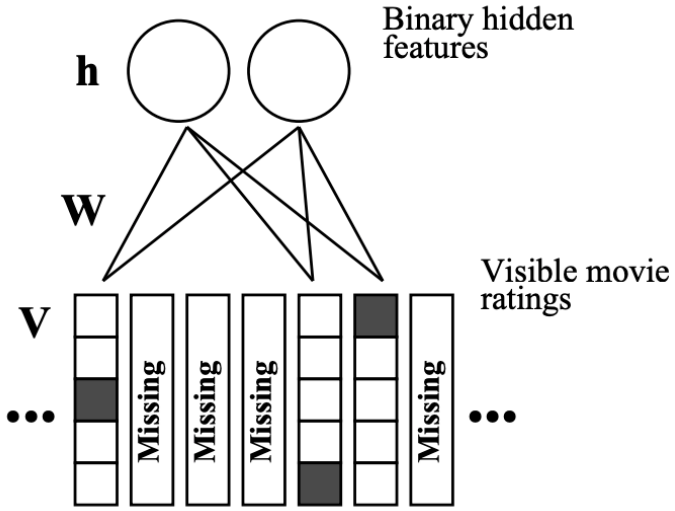
---

[20] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, *Application of Dimensionality Reduction in Recommender System – A Case Study.* ACM WebKDD Workshop, 2000

[21] X. Wang, Y. Wang, *Improving Content-based and Hybrid Music Recommendation using Deep Learning - in Proceedings of the 22nd ACM international conference on Multimedia (MM '14),* 2014

[22] *Netflix Prize Leaderboard,* https://www.netflixprize.com/leaderboard.html, [dostęp 06.06.2020]

przez naukowców z Uniwersytetu w Toronto w 2007 i zasada działania jest bardzo prosta[23]. Polega na użyciu sieci neuronowej z tylko jedną ukrytą warstwą i trenowaniu poprzez modyfikację wag i odchyleń/reszt przy zastosowaniu propagacji wstecznej (ang. backpropagation). Sieć jest „ograniczona", ponieważ może komunikować się wyłącznie z neuronami z innych warstw, co w aktualnie tworzonych modelach jest już standardowym podejściem.



Rys. 6, Schemat działania sieci neuronowej RBM,
źródło: Salakhutdinov, Ruslan & Mnih, Andriy & Hinton, Geoffrey. (2007). Restricted Boltzmann machines for collaborative filtering

Na wejściu sieć powinna otrzymać oceny użytkowników dla poszczególnych filmów, gdzie znów występuje problem niepełnych danych, ponieważ zdecydowana większość użytkowników nie widziała wszystkich możliwych produkcji. Właśnie w tym celu stosowana jest propagacja wsteczna, czyli po klasycznej iteracji następuje tzw. *backward pass*, czyli odwrócenie procesu i próba predykcji niekompletnych danych wejściowych. Otrzymane w ten sposób wartości należy zastosować w celu rekomendacji pozycji o najwyższych przewidywanych ocenach użytkownika.

1.6. Ewaluacja wyników systemów rekomendacji

Aby porównać ze sobą wyniki poszczególnych rozwiązań należy ustalić pewne miary mówiące o tym w jakim stopniu spełnione zostały założenia modelu. Trudno uzyskać informację o tym, czy w istocie użytkownikowi dana rekomendacja przypadła do gustu, bowiem jedyną osobą która może to określić, jest on sam.

W praktyce najlepiej badać różne modele i ich parametry stosując testy A/B, czyli dzielić użytkowników na dwie grupy i analizować które podejście sprawdza się lepiej i dlaczego. W środowisku naukowym jednak jest to utrudnione ze względu na brak działającej konkretnej usługi i stałej bazy odbiorców. Różnice pomiędzy rozwiązaniem działającym online oraz offline nazywa się problemem surogatki (ang. surrogate problem), model który

[23] R. Salakhutdinov, A. Mnih, G. Hinton, *Restricted Boltzmann machines for collaborative filtering. ACM International Conference Proceeding Series, 2007*

działa w teorii może dawać zupełnie inne wyniki na produkcji. Alternatywą dla testów A/B może być pytanie użytkowników wprost, czy rekomendacje były trafione, jednak to podejście nie jest zbyt eleganckie i może negatywnie wpłynąć na wrażenie użytkownika (ang. user experience). W tym podrozdziale przybliżone zostanie kilka miar ilościowych stosowanych do ewaluacji i porównywania ze sobą wybranych silników rekomendacji.

1.6.1. Dokładność

To jak predykcja różni się od realnych wartości określa dokładność (ang. accurancy). Najprościej może być mierzona przez średni błąd bezwzględny (ang. mean absolute error, MAE). Dla każdej przewidzianej wartości należy obliczyć jej bezwzględną różnice od wartości rzeczywistej i całość podzielić przez ilość ocen. Częściej jednak stosowana jest miara średniej kwadratowej błędów (ang. root mean square error, RMSE), która jest surowsza dla wartości odstających. Aby ją obliczyć, należy błąd każdej oceny podnieść do kwadratu i zsumować, podzielić na ilośc ocen jak w MAE oraz z tego obliczyć pierwiastek.

$$MAE = \sum_{i=1}^{n} \left| y_i - x_i \right|, \qquad RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}}$$

Miara RMSE została spopularyzowana przez już wcześniej wspomniany konkurs Netflix'a[24]. Nagroda główna w wysokości miliona dolarów miała przypaść drużynie, której uda się zmniejszyć średni błąd kwadratowy o 10%. Finalnie Netflix nie skorzystał ze zwycięskiego rozwiązania, ponieważ uznano że dokładność rekomendacji mierzona w ten sposób nie wpływa aż tak bardzo na wyniki w prawdziwym świecie. Najważniejsze jest to które filmy składają się na końcową listę top-n rekomendacji oraz to jak użytkownicy na nią reagują[25].

1.6.2. Trafienia

Jest kilka metryk skupiających się na badaniu top-n list. Najprostszą z nich jest wskaźnik trafień (ang. Hit Rate). Zasada działania jest prosta, dla całego zbioru testowego użytkowników należy wyznaczyć top-n listy, a następnie każdą pozycję którą w rzeczywistości ocenili - uznać jako trafienie. Finalnie wskaźnik trafień wyznacza się poprzez podzielenie wszystkich trafień przez liczbę użytkowników.

$$HitRate = \frac{hits}{users}$$

Niestety wadą tego podejścia jest fakt, że nie działamy na zbiorze testowym i treningowym oraz nie ma możliwości zastosowania walidacji krzyżowej (ang. cross validation). Wskaźnik trafień nie określa jak poszczególne pozycje różnią się od predykcji - jak w przypadku pozostałych miar, tylko jak trafna jest cała lista pozycji. Aby zapobiec ewaluacji na tym samym zbiorze na którym system był trenowany oraz sytuacji w której algorytm osiąga zawsze 100% trafień, rekomendując same pozycje najwyżej oceniane przez

---

[24] *Netflix Prize*, https://en.wikipedia.org/wiki/Netflix_Prize, [dostęp 07.06.2020]

[25] F. Kane, *Building Recommender Systems with Machine Learning and AI*, https://www.udemy.com/course/building-recommender-systems-with-machine-learning-and-ai/learn/lecture/11401866#overview, [dostęp 07.06.2020]

użytkowników, stosuję się podejście nazwane *leave-one-out cross validation*, czyli przy tworzeniu listy dla konkretnego użytkownika usuwamy jedną pozycję z jego danych treningowych, a następnie badamy czy algorytm zaproponuje dokładnie tą samą pozycję w fazie testowania. Podejście *leave-one-out* charakteryzuję się tym, że wskaźnik trafień jest analogicznie o wiele niższy od zwykłego *Hit Rate* oraz na małych zbiorach danych działa o wiele gorzej. Jednak ta miara daje dużo informacji o tym jak model sprawuje się na produkcji - z prawdziwymi użytkownikami, dlatego obok RMSE jest najczęściej stosowanym wskaźnikiem do ewaluacji działania silnika rekomendacji.

Użytkownicy częściej wybierają pozycję z przodu listy, dlatego istotna jest kolejność ich wyświetlania. Miarą która zwraca na to uwagę jest średni współczynnik trafień z rangami - ARHR (ang. average reciprocal hit rate). Działa tak jak zwykły hit rate, lecz każde trafienie bliżej początku jest lepiej nagradzane. Sprawdza się dobrze w systemach, gdzie listy wyświetlane są w kontenerach które użytkownik musi przewijać np. takich jak strona główna Netflix'a.

$$ARHR = \frac{\sum_{i=1}^{n} \frac{1}{rank_i}}{Users}$$

Przy małym zbiorze danych warto wprowadzić skumulowany wskaźnik trafień - cHR (ang. cumulative hit rate). Polega na wprowadzeniu progu dla którego trafienia są mierzone, tak aby rekomendacja słabych pozycji nie była wliczana do końcowego wskaźnika. Ostatnią miarą bazującą na trafieniach jest rHR (ang. rating hit rate), który zwyczajnie rozbija oceny na kategorie (np. od 1 do 5) i dla każdej z nich obliczany jest osobno. Pozwala to zaobserwować rozkład proponowanych rekomendacji i to jak algorytm sprawuję się w poszczególnych kategoriach.

1.6.3. Pokrycie

Dokładność predykcji to nie jedyna cecha jaką da się zmierzyć w modelu służącym do rekomendacji. Pokrycie (ang. coverage) to procent możliwych filmów jaki nasz system jest w stanie wybrać z całej bazy dostępnych filmów. Często miara pokrycia nie współgra z dokładnością, należy odnaleźć pomiędzy nimi idealny balans. Chcąc proponować coraz to lepsze pozycje, np. wprowadzając próg pod którym pozycje ze zbyt niską prognozowaną oceną będą odrzucane, narażamy się na zmniejszenie miary całkowitego pokrycia. Problem ten pogłębia się również, gdy skalujemy bazę danych zwiększając liczbę dostępnych pozycji. Każda nowa pozycja, nie będzie od razu rekomendowana i co z tym idzie, pokrycie będzie się zmniejszać.

1.6.4. Różnorodność

Miarę określającą szerokość tematyczną proponowanych treści nazywamy różnorodnością (ang. diversity). Mało różnorodny model po przesłuchaniu jednego albumu pewnego zespołu, rekomenduje twórczość jedynie tworzoną przez ten zespół. W kontraście do tego, model zbyt różnorodny zwraca praktycznie losowe utwory nie mające ze sobą wiele wspólnego. Aby obliczyć ten wskaźnik, należy znaleźć średnią miarę podobieństwa - S,

bazując na macierzach obliczanych w przypadku niektórych algorytmów. Miara szerokości jest przeciwnością średniej wartości podobieństwa, więc aby ją otrzymać należy podstawić do równiania:

$$Diversity = (1 - S)$$

1.6.5. Odkrywczość

Popularność pozycji znajdujących się w top-n listach nazywamy miarą odkrywczości modelu. W pewnych przypadkach nie chcemy, aby użytkownicy otrzymywali ciągle te same znane tytuły i aby czasem wpadli na coś dla nich nowego i świeżego. Jednak zbyt wysoka odkrywczość to znowu proponowanie losowych pozycji, użytkownik chciałby zobaczyć kilka pozycji które zna, aby stwierdzić że system działa poprawnie i jest pod niego spersonalizowany. Pozycje popularne w większości przypadków nie są popularne bez powodu, a dlatego, że część populacji uważa je za dobre. Celem systemów rekomendacji w pewnym sensie jest próba wyjścia z tej bańki popularnych tytułów i propozycja tych mniej znanych, niszowych tytułów, które również znajdują się w bazie, lecz na końcu tak zwanego „długiego ogona". Propozycja treści zbyt nieatrakcyjnych lub nie trafiających w niszę odbiorcy może skutkować negatywnym odbiorem modelu przez użytkownika.

Tak jak w poprzednim przypadku nie ma określonego przedziału w których te miary powinny się zawierać. Wiele zależy od danych którymi dysponujemy oraz jaki jest cel który chcemy osiągnąć. Najlepiej jest odnaleźć balans, który będzie dawał wyniki uznawane za najbardziej zadawalające.

1.6.6. Inne miary

Miara mierząca częstość zmiany rekomendowanych obiektów, czyli inaczej mówiąc odświeżania listy nazywana jest *churn.* Rekomendacje nie powinny być stale takie same, ponieważ jeżeli użytkownik widzi dany obiekt kolejny raz i nadal w niego nie kliknął, prawdopodobnie nie jest nim zainteresowany.

Wskaźnik *responsivnes* odpowiada na pytanie jak szybko nowe działanie użytkownika ma wpływ na zmianę proponowanych mu produktów. Z oczywistych względów należy dążyć do maksymalizacji tej miary, ale należy brać pod uwagę również skomplikowanie działania oraz koszta eksploatacyjne danego rozwiązania na produkcji.

2. Budowanie silnika rekomendacji

2.1. Zbiór danych

GroupLens jest grupą badawczą prowadzoną przez naukowców z Uniwersytetu w Minnesocie, która udostępnia precyzyjnie przygotowane dane z ocenami filmów. Ich celem jest rozpowszechnienie wiedzy na temat systemów rekomendacji pośród studentów i grup badawczo-rozwojowych[26].

Na ich stronie dostępne są różne warianty zbiorów danych, odpowiednie do nauki, prac badawczych wykorzystujących nowe algorytmy i obszerne syntetyczne dane (zawierające aż miliard ocen) otrzymane w rezultacie augmentacji dostępnych zbiorów[27]. Na stronie dostępne są również poprzednie wersje zbiorów danych z lat wcześniejszych.

W tej pracy zdecydowano się na skorzystanie z najświeższych danych datujących na październik 2018, wariant ten rekomendowany jest do działań akademickich. Dane zawierają ponad 100 tys. ocen na ponad 9 tys. filmach, wystawionych przez 610 użytkowników[28]. Taki zbiór pozwoli na wielokrotne testowanie algorytmów ze zmiennymi parametrami w celu ich ewaluacji, bez konieczności długotrwałego czekania na otrzymane wyniki.

Dane składają się z 4 tabeli - *links*, *movies*, *ratings* oraz *tags,* zapisanych w formacie *csv* (ang. comma-separated values)[29]. Zebrane zostały w latach 1996-2018, użytkownicy zostali wybrani losowo z jedynym warunkiem, że oddali przynajmniej 20 ocen. Nie zawierają informacji demograficznych, każdy użytkownik przedstawiony jest jedynie za pomocą unikalnego id, będącym jednocześnie kluczem tworzącym relacje pomiędzy tabelami. Format tekstu to UTF-8.

Zbiór danych nie był w żaden sposób przez autora modyfikowany, natomiast został zbadany pod kątem rozkładów i statystyk opisowych. Niektóre, bardziej ciekawe obserwacje zostały przedstawionej w poniższych podrozdziałach.

2.1.1 Oceny

Wszystkie oceny znajdują się w pliku *ratings.csv,* każdy wiersz reprezentuje jedną ocenę dla jednego filmu wraz z czasem wystawienia oceny *timestamp*, liczonym jako każda sekunda od północy 1 stycznia 1970 UTC. Oceny wystawione zostały 5-cio gwiazdkowej z możliwością wystawienia połowy gwiazdki (od 0,5 do 5,0). Wszystkich rekordów jest dokładnie 100 836, unikalnych użytkowników 610, użytkownik 414 ma największą liczę ocenionych pozycji - 2698. Najpopularniejszy film (Forest Gump, 1994) ma wystawionych 329 ocen ze średnią 4,16. Średnia ocena wszystkich filmów to 3,5. Rozkład cechuję się skośnością lewostronną widoczną na wykresie (Rys. 8).

---

[26] *About MovieLens, https://movielens.org/info/about,* [dostęp 09.06.2020]

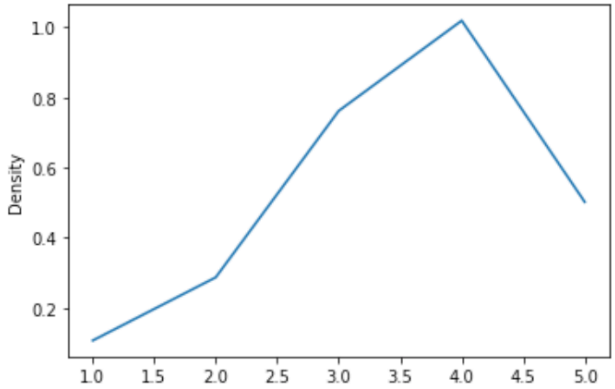
[27] *MovieLens datasets, https://grouplens.org/datasets/movielens/,* [dostęp 09.06.2020]

[28] F. M. Harper, J. A. Konstan, *The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS),* 2015

[29] *Comma-separated values, https://en.wikipedia.org/wiki/Comma-separated_values,* [dostęp 09.06.2020]

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

Rys. 7, Struktura danych pliku ratings.csv, źródło: własne



Rys. 8, Rozkład wszystkich ocen, źródło: własne

2.1.2. Tagi

Tagi w formie krótkiego komentarza do oceny o dowolnej treści, wprowadzonego przez użytkownika. Możliwe do wykorzystania np. w podejściu *Mise-en-Scene* omawianym w poprzedniej części pracy. Wyrażenia jedno lub dwu wyrazowe stanowią 91% tagów.

	userId	movieId	tag	timestamp
0	2	60756	funny	1445714994
1	2	60756	Highly quotable	1445714996
2	2	60756	will ferrell	1445714992
3	2	89774	Boxing story	1445715207
4	2	89774	MMA	1445715200

Rys. 9, Struktura danych pliku tags.csv, źródło: własne

	userId	movieId	tag	timestamp
1095	474	356	shrimp	1137375519
1096	474	356	Vietnam	1137375519
2782	533	356	bubba gump shrimp	1424753866
2783	533	356	lieutenant dan	1424753866
2784	533	356	stupid is as stupid does	1424753866
2890	567	356	bittersweet	1525287545
2891	567	356	emotional	1525287537
2892	567	356	heartwarming	1525287541
2893	567	356	touching	1525287539

Rys. 10, Tagi dla filmu Forest Gump (1994), źródło: własne

2.1.3. Filmy

Dane zawierające informację o 9 742 filmach, każdy z nich ma przynajmniej jedną ocenę. Oprócz tytułu i roku produkcji w nawiasie, podany jest gatunek do którego film należy.

	movieId	title	genres
312	354	Cobb (1994)	Drama
313	355	Flintstones, The (1994)	Children|Comedy|Fantasy
314	356	Forrest Gump (1994)	Comedy|Drama|Romance|War
315	357	Four Weddings and a Funeral (1994)	Comedy|Romance
316	358	Higher Learning (1995)	Drama

Rys. 11, Struktura danych pliku movies.csv, źródło: własne

Gatunki oddzielone są znakiem kreski pionowej, dzięki temu istnieje możliwość przekształcenia ich z użyciem *one-hot encoding*. Jest 20 możliwych kategorii gatunków, łącznie z brakiem zdefiniowanego gatunku oraz licząc IMAX jako odrębny gatunek. Dzięki zastosowaniu tej techniki istnieje możliwość np. znalezienia najpopularniejszych gatunków. W tym przypadku jest to *Romance* oraz *Children's*.

	movieId	title	(no genres listed)	Action	Adventure	Animation	Children	Comedy	Crime	Documentary	...	Film-Noir	Horror
312	354	Cobb (1994)	False	False	False	False	False	False	False	False	...	False	False
313	355	Flintstones, The (1994)	False	False	False	False	True	True	False	False	...	False	False
314	356	Forrest Gump (1994)	False	False	False	False	False	True	False	False	...	False	False
315	357	Four Weddings and a Funeral (1994)	False	False	False	False	False	True	False	False	...	False	False
316	358	Higher Learning (1995)	False	False	False	False	False	False	False	False	...	False	False

Rys. 12, Dane po zastosowaniu one-hot encoding, źródło: własne

2.1.4. Odwołania

Zbiór MovieLens powstał w oparciu o bazy danych serwisów imdb.com[30] oraz themoviedb.org[31]. W tabeli *links* znajdują się wartości reprezentujące id poszczególnych filmów w ww. portalach. Odnośniki mogą być przydatne, jeżeli potrzebne byłby dodatkowe informacje o produkcjach takie jak reżyser, aktorzy, ogólne przyjęcie publiczności itp.

	movieId	imdbId	tmdbId
0	1	114709	862.0
1	2	113497	8844.0
2	3	113228	15602.0
3	4	114885	31357.0
4	5	113041	11862.0

Rys. 13, Struktura danych pliku links.csv, źródło: własne

2.2. Biblioteka Surprise

Implementacja algorytmów przytoczonych w poprzedniej części byłaby bardzo żmudna i z pewnością przy ręcznej próbie odwzorowania tych algorytmów wkradłyby się błędy. Na szczęście Nicolas Hug, naukowiec z Uniwersytetu w Kolumbii jest autorem

[30] *Serwis IMDB, https://www.imdb.com/,* [dostęp 09.06.2020]

[31] *Serwis TMDB https://www.themoviedb.org/,* [dostęp 09.06.2020]

biblioteki w języku python, która ma w sobie zaimplementowane najpopularniejsze z nich oraz w łatwy sposób pozwala na dodawanie własnych algorytmów, modyfikowanie oraz mierzenie ich wydajności[32].

Biblioteka *SurPRISE* (od słów Simple Python RecommendatIon System Engine) jest narzędziem pozwalającym na budowanie i eksperymenty z systemami rekomendacji używając do tego konkretnych danych (ang. explicit). Obsługa danych domniemanych (ang. implicit) takich jak kliknięcia, czy *watch-time* nie jest zaimplementowana, ale można to zrobić na własną rękę. Wbudowane również są narzędzia do testowania krzyżowego, ewaluacji, analizy oraz porównywania poszczególnych podejść.

2.3. Kod źródłowy i wykorzystane narzędzia

Badanie przeprowadzono z wykorzystaniem języka python w wersji 3.7.5. Oprócz ww. biblioteki skorzystano z *numpy*[33] - biblioteki pomocnej przy operacjach matematycznych, *matplotlib*[34] - biblioteki do renderowania wykresów, *pandas*[35] - biblioteki służącej do obsługi danych w wygodnym formacie ramek danych oraz *tensorflow*[36] - biblioteki która początkowo służyła do lepszego zarządzania zasobami komputera podczas obliczeń na macierzach, a obecnie jest jedną z głównych wykorzystywanych przy projektach związanych z uczeniem maszynowym.

Do pracy dołączony jest kod źródłowy zawierający operacje ilustrujące przebieg badania. Kod oprócz wkładu własnego autora pracy, zawiera fragmenty zaczerpnięte z kursu *Building Recommender Systems with Machine Learning and AI* autorstwa Franka Kane'a, byłego pracownika serwisu Amazon[37].

2.4. Przebieg badania

2.4.1. Użytkownik benchmarkowy

Aby mieć odniesienie co do wartości, wyniki algorytmów przedstawiono również z algorytmem dobierającym pozycje losowo. Ziarno generatora ustawiono na 42 oraz dodatkowo wybrano użytkownika o id 42 tak, aby móc porównać generowane top-n listy z jego gustem. Po przyjrzeniu się jego ocenom, stwierdzono że w szczególności podobają mu się klasyczne produkcje amerykańskie, kino gangsterskie i wojenne, natomiast nie przepada za filmami opartymi na komiksach, slapstickowych komediach i satyrach.

Wyniki losowego podejścia zawarto w Tabeli 1, RMSE posłuży jako główna miara w porównywaniu poszczególnych podejść. Interpretując MAE możemy powiedzieć, w skali

---

[32] *SurpriseLib, A Python scikit for recommender systems,* http://surpriselib.com/, [dostęp 09.06.2020]

[33] *NumPy,* https://numpy.org/, [dostęp 10.06.2020]

[34] *MatPlotLib,* https://matplotlib.org/, [dostęp 10.06.2020]

[35] *Pandas,* https://pandas.pydata.org/, [dostęp 10.06.2020]

[36] *Tensorflow,* https://www.tensorflow.org/, [dostęp 10.06.2020]

[37] F. Kane, *Building Recommender Systems with Machine Learning and AI,* https://www.udemy.com/course/building-recommender-systems-with-machine-learning-and-ai/, [dostęp 10.06.2020]

pięcio-gwiazdkowej model popełnia błąd predykcji wielkości 1,14 gwiazdki. Chcąc interpretować Hit Rate, można powiedzieć, że w 0,66% przypadkach algorytm dobrze dobrał pozycję w liście rekomendacji. Pokrycie wskazuje, że model jest w stanie korzystać ze 100% zbioru, a odkrywczość jest na absurdalnie wysokim poziome z prostego powodu, czyli proponowania losowych pozycji. Przy losowym podejściu nie przedstawiono końcowej top-n listy dla użytkownika 42, ponieważ przedstawia losowe filmy ze zbioru nie mające ze sobą nic wspólnego.

Tabela 1 - Random

	RMSE	MAE	HR	cHR	ARHR	Coverage	Novelty
Random	1,4264	1,1408	0,66%	0,66%	0,38%	100%	852,8493

2.4.2. Content Based KNN

Korzystając z podejścia k-najbliższych sąsiadów bazującym na samej zawartości, otrzymano wyniki zdecydowanie lepsze od podejścia losowego. Podczas badania sprawdzono wszystkie kombinacje korzystające z cech takich jak gatunek, rok produkcji oraz danych *Mise-en-Scene* udostępnionych przez badaczy z Politechniki w Milanie[38]. Dodatkowo starano się wprowadzić alternatywne podejścia (np. dodawając wagi), lecz nie udało się pobić wyniku otrzymanego przez wariant mierzący macierz podobieństwa bazując na gatunku i roku produkcji. Ciekawym jest fakt, że pomimo dostatecznie zadawalających wyników, podejście *Mise-en-Scene* pogorszyło wynik pary gatunek-rok produkcji.

Tabela 2 - Content Based KNN

	RMSE	MAE
Genre	0,9280	0,7168
Year	0,9336	0,7224
MeS	1,0375	0,8062
Genre-Year	0,9055	0,6983
Genre-MeS	1,0281	0,7975
Year-MeS	1,0207	0,7939
Genre-Year-MeS	1,0133	0,7858
Genre-Year (alt)	0,9099	0,7032
Random	1,4264	1,1408

Zdecydowano się nie wyliczać list rekomendacji dla wszyskich wariantów, ze względu na złożoną ilośc operacji na sprzęcie i długi czas oczekiwania na wynik. Kolejne iteracje wyliczające powyższe miary zajmowały od 2 do 5 minut, w zależności od

[38] Y. Deldjoo, M. Elahi, P. Cremonesi, *Using Visual Features and Latent Factors for Movie Recommendation, ACM RecSys Workshop on New Trends in Content-based Recommender Systems (CBRecSys), ACM RecSys 2016, Massachusetts Institute of Technology (MIT)*, 2016, http://recsys.deib.polimi.it/mise-en-scene-visual/, [dostęp 11.06.2020]

wykorzystywanych technik. Chcąc wykorzystać system w praktyce należałoby skorzystać z o wiele większego zbioru danych, nie z 10 tys. pozycji filmowych jak w tym przypadku. Niestety to wiązałoby się z wyznaczeniem obszernej macierzy podobieństw, a czas oczekiwania na rezultat znacznie by się wydłużył.

Poniżej przedstawiona została końcowa lista dla podejścia o najniższym RMSE, trudno stwierdzić czy te rekomendacje pasują do gustu użytkownika 42. Można zauważyć, że wszystkie proponowane pozycje są bardzo stare, jednak przeglądając filmy ocenione przez tego użytkownika widać że nie oglądał ich wcale tak dużo. Pomimo tego, że podejście uwzględniające rok produkcji i gatunek okazało się najwydajniejsze pod względem predykcji ocen (najmniejsze RMSE), końcowa lista wydaję się być nie trafiona (Rys. 14). Lista wygenerowana bazując na samych gatunkach (Rys. 15) z kolei proponuje filmy dokumentalne, trudno dokładnie określić dokładność tych list nie znając osobiście tego użytkownika oraz tytułów mu proponowanych. Aby uzyskać wyniki satysfakcjonujące należałoby próbować z wieloma kombinacjami, a wyniki sprawdzać za pomocą testów A/B.

```
Johnny Eager (1942) 4.780539109225711
Grand Hotel (1932) 4.78039199748019
Divorcee, The (1930) 4.78039199748019
All This, and Heaven Too (1940) 4.780391
Wuthering Heights (1939) 4.7803919974801
Goodbye, Mr. Chips (1939) 4.780391997480
Dark Victory (1939) 4.780391997480187
Anna Karenina (1935) 4.780391997480187
Maidens in Uniform (MÃ¤dchen in Uniform)
Camille (1936) 4.780391997480187
```

Rys. 14, Lista top 10 rekomendacji dla użytkownika 42 (podejście genre-year), źródło: własne

```
Everest (1998) 4.413418273984495
Titanica (1992) 4.413418273984495
Alaska: Spirit of the Wild (1997) 4.413418273
Africa: The Serengeti (1994) 4.41341827398449
Michael Jordan to the Max (2000) 4.4134182739
Ghosts of the Abyss (2003) 4.413418273984495
Inside Job (2010) 4.333333333333334
The Jinx: The Life and Deaths of Robert Durst
Crumb (1994) 4.333333333333334
Maya Lin: A Strong Clear Vision (1994) 4.3333
```

Rys. 15, Lista top 10 rekomendacji dla użytkownika 42 (podejście genre), źródło: własne

2.4.3. Collaborative Filtering KNN

W przypadku filtrowania zbiorowego sprawdzono trzy metody wyliczana macierzy podobieństwa (MSD - czyli odległość euklidesowa, Cosinus oraz Pearson) na dwóch typach - opartych na użytkownikach i na pozycjach. Najlepszy wynik odniosła kombinacja MSD przy podejściu opartym na zawartości. Mierząc to jak często algorytm proponował pozycję wcześniej usuniętą, czyli stosując walidację *leave-one-out* - najlepiej sprawdziła się miara obliczana przy pomocy korelacji Pearsona. Próbując interpretować wynik można powiedzieć, że około co siódma lista zawierała w sobie pozycję wcześniej usuniętą, co jest wynikiem całkiem zadawalającym.

Tabela 3 - Collaborative Filtering KNN

Type	User based			Item based			Random
Sim	MSD	Cosine	Pearson	MSD	Cosine	Pearson	-
RMSE	0.9554	0,9787	0,9802	0.9149	0,9788	0,9755	1,4264
MAE	0.7330	0,7547	0,7558	0.7031	0,7610	0,7550	1,1408
HR	7,38%	5,08%	7,21%	12,62%	14,10%	15,41%	0,66%

Końcowe listy przedstawiono dla obu najlepszych podejść (Rys. 16 i 18). Pierwsza lista wydaję się zwracać pozycję o wiele lepiej dopasowane niż te bazujące na zawartości w które przedstawiono w poprzednim rozdziale. Widoczne są klasyczne filmy hollywoodzkie oraz komedie romantyczne. Z kolei podejście z największą miarą trafień, jest bardzo zbliżone z tą różnicą, że w miejsce komedii romantycznych wskoczyło kilka dramatów i filmów przygodowych. Trudno określić dokładnie, ale druga lista wydaję się do tej pory najlepiej trafioną, porównując z gustem i ocenami wystawionymi przez użytkownika 42.

```
Jurassic Park (1993) 3.2
Lord of the Rings: The Fellowship of the Ring, The (2001) 2.7
Minority Report (2002) 2.6
Eternal Sunshine of the Spotless Mind (2004) 2.0
Ed Wood (1994) 2.0
Rushmore (1998) 2.0
Memento (2000) 1.9
Lord of the Rings: The Return of the King, The (2003) 1.9
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981) 1.8
Bridget Jones's Diary (2001) 1.8
Amelie (Fabuleux destin d'AmÃ©lie Poulain, Le) (2001) 1.8
```

Rys. 16, Lista top 10 rekomendacji dla użytkownika 42 (item based - MSD),
źródło: własne

```
Jurassic Park (1993) 4.4
American Beauty (1999) 3.8
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981) 3.7
Animal House (1978) 3.6
One Flew Over the Cuckoo's Nest (1975) 3.5999999999999996
Army of Darkness (1993) 3.4
Fargo (1996) 3.4
Aliens (1986) 3.3000000000000003
Twelve Monkeys (a.k.a. 12 Monkeys) (1995) 3.3
Bill & Ted's Excellent Adventure (1989) 3.2
Amadeus (1984) 2.8
```

Rys. 17, Lista top 10 rekomendacji dla użytkownika 42 (item based - Pearson),
źródło: własne

2.4.4. SVD

Stosując technikę faktoryzacji macierzy oraz jej ulepszoną wersję (SVD++), otrzymano następujące wyniki (Tabela 4). Jak do tej pory są to najlepsze wyniki osiągnięte w badaniu, średni błąd absolutny został zmniejszony prawie o połowę porównując do wejściowego, losowego podejścia. Oznacza to że predykcja oceny różni się średnio o 0,66 gwiazdki. Wyniki widoczne w tabeli otrzymano za pomocą domyślnych parametrów algorytmów bazujących na SVD.

Tabela 4 - SVD

	RMSE	MAE	HR
SVD	0,8796	0,6743	2,95%
SVD++	0,8664	0,6641	2,46%
Random	1,4264	1,1408	0,66%

Końcowe listy proponowanych tytułów przedstawiono na grafikach (Rys. 18 i 19). Filmy tutaj prezentowane, zdecydowanie są bardziej popularne i uznawane przez środowisko jako warte uwagi. Jednocześnie przeglądając pozycje ocenione przez użytkownika 42, widać że prawdopodobnie trafiają w jego gust filmowy, a z pewnością lepiej niż listy prezentowane w poprzednich rozdziałach. Nie odzwierciedla tego miara trafień, jest o wiele niższa niż w przypadku metody grupowania zbiorowego. Prawdopodobnie jest to spowodowane tym, że SVD obiera na cel minimalizację błędów predykcji. Porównując samą jakość całych list rekomendacji, bardziej trafione wydają się listy powstałe po zastosowaniu faktoryzacji macierzy.

```
We recommend:
Fargo (1996) 4.8236591478413215
Great Escape, The (1963) 4.770910868354396
Trainspotting (1996) 4.725473019079763
Apocalypse Now (1979) 4.72263817618543
Lawrence of Arabia (1962) 4.646725065005703
Lost in Translation (2003) 4.642391213708452
Cinema Paradiso (Nuovo cinema Paradiso) (1989) 4.6397329800681595
Three Colors: Red (Trois couleurs: Rouge) (1994) 4.625488000014705
Seventh Seal, The (Sjunde inseglet, Det) (1957) 4.617490494586079
Serenity (2005) 4.602049352383305
```

Rys. 18, Lista top 10 rekomendacji dla użytkownika 42 (SVD), źródło: własne

```
We recommend:
Guess Who's Coming to Dinner (1967) 4.789032258803795
Great Escape, The (1963) 4.752691449601114
Blade Runner (1982) 4.742011310369239
Once Upon a Time in the West (C'era una volta il West) (1968) 4.729462148767498
Brazil (1985) 4.710647281858312
Hustler, The (1961) 4.691036853314538
The Martian (2015) 4.655538209314799
Good, the Bad and the Ugly, The (Buono, il brutto, il cattivo, Il) (1966) 4.6473281736454
86
Streetcar Named Desire, A (1951) 4.6458035094677275
Lost in Translation (2003) 4.641497944298121
```

Rys. 19, Lista top 10 rekomendacji dla użytkownika 42 (SVD++), źródło: własne

Powyższe wyniki otrzymano z parametrami domyślnymi, które można dowolnie modyfikować w celu uzyskania lepszych efektów. Aby uniknąć ręcznego wpisywania kolejnych wartości dla następnych iteracji zastosowano moduł *GridSearchCV* pozwalający na maksymalizację wyników, testując po kolei różne kombinacje podane przez użytkownika. Jako funkcję celu obrano RMSE i pośród kilku powtórzeń, wybrano te najlepsze (Tabela 5). Zmieniano parametry takie jak liczba liczba epok (*n_epochs*), współczynnik uczenia (*learning rate*) oraz liczba składowych (*n_factors*).

Tabela 5 - tuned SVD

epochs/LR/ factors	RMSE	MSA	HR	ARHR	Coverage	Diversity	Novelty
untuned	0,8802	0,6749	2,13%	0,64%	92,46%	0,0301	491,6
30/0,005/10	0,8749	0,6706	2,30%	0,80%	90,98%	0,0891	973,0
60/0,01/200	0,8753	0,6713	3,11%	1,13%	97,7%	0,1040	1015,4
60/0,01/300	0,8729	0,6700	3,11%	1,31%	97,87%	0,1262	1194,6
random	1,4264	1,1408	0,66%	0,15%	100%	0,0489	840,7

W najlepszym przykładzie udało się poprawić wynik dokładności o około 1%, współczynnik trafień również jest większy, ale nadal nie dorównuje temu otrzymanemu przy filtrowaniu zbiorowym. Dostrojone SVD ignoruje jedynie 2% wszystkich pozycji, filmy różnią się od siebie w dużym stopniu i są o wiele „świeższe" od tych z niedostrojonego SVD.

2.4.5. Deep learning

Badając jak rekurencyjne sieci neuronowe poradzą sobie z zagadnieniem rekomendacji sprawdzono podejście RMB. Wyniki z domyślnymi parametrami (20 epok, 50 neuronów ukrytych, współczynnik uczenia na poziomie 0.001) oraz paroma próbami ich dostrojenia przedstawiono w Tabeli 6.

Tabela 6 - RNN

epochs/hiddenDim/LR	RMSE	MSA
untuned (20/40/0.001)	1,2815	1,0873
20/40/0.2	1,2807	1,0866
20/20/0.1	1,2797	1,0857
random	1,4264	1,1408

Jak łatwo zauważyć, dostrojone wyniki nie różnią się znacząco od wartości domyślnych. Pomimo faktu, że wypadają nieco lepiej od losowych propozycji nie jest to wynik zadowalający. Powodem może być zbiór danych o niedostatecznie wielkiej ilości pozycji lub niewłaściwy dobór parametrów. W badaniu sprawdzono jedynie kilkanaście możliwości, a każde z nich zajmowało po parę godzin. Istnieje szansa, że podejście RMB osiągałoby wyniki porównywalne lub nawet przewyższające poprzedników, co udowodniono w publikacjach naukowych na ten temat. Jednak odpowiednie dobranie parametrów to proces długotrwały i należy odnaleźć odpowiedni balans pomiędzy końcowym wynikiem i pracą potrzebną aby je osiągnąć.

Filmy prezentowane na końcowej liście nie wydają się mieć ze sobą dużo wspólnego, niektóre z nich to pozycje znane i uznawane za obiektywnie dobre filmy. (Rys. 20). Współczynnik trafień na poziomie 0,15%, czyli najniższym jak do tej pory, gorszym nawet od rekomendacji podawanych losowo, jest sporym zaskoczeniem. Zastosowanie rekomendacji opartych na zawartości również otrzymało wynik poniżej losowości, jednak w tamtym

przypadku proponowane były produkcje skrajnie niszowe, a tutaj lista wydaję się jakościowo bardziej trafna.

```
We recommend:
Sin City (2005) 2.7637386
In Bruges (2008) 2.7631805
Big Short, The (2015) 2.7631671
Watchmen (2009) 2.7629402
Thank You for Smoking (2006) 2.7625976
LÃ©on: The Professional (a.k.a. The Professional) (LÃ©on) (1994) 2.762079
Prophet, A (Un ProphÃ¨te) (2009) 2.7618642
History of Violence, A (2005) 2.7615077
Shaun of the Dead (2004) 2.7615075
Drive (2011) 2.7614992
```

Rys. 20, Lista top 10 rekomendacji dla użytkownika 42 (RBM), źródło: własne

2.4.6. Porównanie wyników

W poniższej tabeli zestawiono najlepsze wyniki spośród badanych metod. Trudno porównać je ze sobą, bowiem z natury działania tych algorytmów każdy z nich celuje w co innego. W badaniu, większość podejść starała się minimalizować błąd predykcji i spośród nich najlepszy okazał się algorytm SVD++. W praktyce może okazać się, że miarą na której warto się skupić jest jednak współczynnik trafień, przy którym najlepiej sprawdzały się algorytmy filtrowania zbiorowego.

Podejście oparte na pozycjach też ma swoje zalety, możliwa jest rekomendacji dla tylko jednego użytkownika, lecz jak wykazało badanie, taki system należy bardzo dokładnie wymodelować, aby rekomendacje nie były zbyt zamknięte. Również sieci neuronowe mają swój potencjał, zastosowane tutaj RMB to tylko jeden z wielu schematów próbujących rozwiązać problem rekomendacji treści za pomocą tego narzędzia.

Tabela 7 - Porównanie najlepszych wyników

	RMSE	MAE	HR
Content	0,9055	0,6983	0,30%
CF User	0.9554	0.7330	7,38%
CF Item	0.9149	0.7031	15,41%
SVD	0,8664	0,6641	2,46%
RMB	1,2797	1,0857	0,15%
Random	1,4264	1,1408	0,66%

Podsumowanie

Wnioski

Sugestia odpowiednich treści nie jest zadaniem trywialnym. Jak można było zauważyć zastosowanie prostych, klasycznych algorytmów dawało wyniki lepsze od list losowych, ale nie do końca zaspokajało potrzebę dobrze trafionych rekomendacji. Z kolei techniki bardziej złożone, po ustaleniu odpowiednich parametrów i dostosowania zbioru danych, zwracały satysfakcjonujące rezultaty. Kosztem lepszych efektów, może być zanik przejrzystości otrzymanych wyników. Stosowanie sztucznych sieci neuronowych z warstwami ukrytymi, czy faktoryzacja macierzy na przestrzeń dwuwymiarową pozwalają na odkrycie zależności i wzorów nie możliwych do dostrzeżenia przez człowieka, zjawisko to nazywane jest czarną skrzynką (ang. black box) i konserwatywne organizacje (takie jak np. banki, urzędy) mogą być nie chętne do wdrażania takich rozwiązań w swoich systemach.

Nie wykluczona jest możliwość, że wyniki badania są stronnicze. Analiza wykonana została na stosunkowo małym zbiorze danych, przygotowanym specjalnie do celów badawczych. Nie ma gwarancji, że modele tak przygotowane i uzyskujące wysokie wartości w miarach ewaluacji, sprawdziłyby tak samo w realnym zastosowaniu. Aby uzyskać najbardziej wiarygodne i dające dużo informacji o przedmiocie badanym wyniki, należałoby zastosować testy A/B. Jednak sporządzenie takich testów, które dają prawdziwe, nie przekłamane wyniki, również zadaniem trywialnym nie jest. Bardzo łatwo popełnić błąd, który negatywnie rzutuje na efekt badania.

Praktyka pokazuje, że najczęściej stosowane są metody hybrydowe, czyli korzystające z dwóch lub więcej rozwiązań w celu stworzenia rekomendacji. Pracownicy firmy Netflix, zapytani o to które z algorytmów są przez nich aktualnie stosowane, twierdzą że korzystają ze wszystkich. Nie wiadomo na ile jest to próba zachowania tajemnicy przedsiębiorstwa, a na ile ciągła chęć postępu, poprzez stosowanie różnych hybryd algorytmów ich ewaluacji na „żywym organizmie". Mieszanie ze sobą różnych technik, pozwala też na wyeliminowanie wad niektórych z nich. Dla przykładu stosowanie filtrowania zbiorowego jest bardzo dobrym pomysłem, ale przy nowym użytkowniku nie jest w stanie zaproponować nic ciekawego. Zastosowanie w takiej sytuacji filtrowania opartego na zawartości, rozwiązuje ten problem i takie hybrydowe daje efekt synergiczny.

Systemy rekomendacji stają się coraz bardziej popularne i w najbliższym czasie, możemy doczekać się w sytuacji w której nie stosowanie takiego rozwiązania będzie się wiązało ze stratą zainteresowania użytkownika daną aplikacją. Stale rosnąca ilość treści w internecie oraz społeczeństwo kapitalistyczne w którym żyjemy jeszcze bardziej napędza popyt na modele pozwalające na łatwiejszy dostęp do treści i produktów, które uwielbiamy.

Możliwe kontynuacje badania

W pracy ukazano jedynie metody najbardziej znane i cenione, wskazano kilka ciekawych, eksperymentalnych podejść, które w przyszłości, po dopracowaniu mogą okazać zwracać jeszcze lepsze rezultaty. Oprócz tego istnieje szereg innych podejść do problemu

rekomendacji, filtrowania pozycji niechcianych lub rankingu przedstawianych wyników. Branża rozwija się z roku na rok i powstają nowe metody, jeszcze bardziej innowacyjne i bazujące na świeżych technologiach.

Z pewnością można stwierdzić, że przy bardziej złożonych metodach występuje mnogość możliwych konfiguracji parametrów. W pracy zastosowano techniki pozwalające na łatwiejsze ich przeszukiwanie w celu znalezienia optymalnego rozwiązania, jednak nie jest potwierdzone, że takowe odnaleziono. Długotrwała modyfikacja i testowanie parametrów ww. metod może skutkować poprawą wyników.

Badanie przeprowadzono na danych filmowych udostępnianych bezpłatnie w internecie. Istnieje możliwość jego powtórzenia, tym razem na danych zebranych przez realny system, przykładowo kierowany do polskich odbiorców, z danymi zawierającymi więcej rodzimych pozycji. Porównanie wyników obu prac mogłoby wykazać na różnice kulturowe pomiędzy odbiorcami z różnych części świata. Równie dobrze podobne badanie, można bardzo łatwo powielić z użyciem danych innego rodzaju np. produktów sklepu internetowego lub pozycji muzycznych.

Spis Rysunków

Spis Tabel

Bibliografia

1. About MovieLens, https://movielens.org/info/about, [dostęp 09.06.2020]
2. Asanov D.A., Algorithms and Methods in Recommender Systems, 2011
3. Candillier L., Jack K., Fessant F., Meyer F., State of the Art Recommender System, 2009
4. Comma-separated values, https://en.wikipedia.org/wiki/Comma-separated_values, [dostęp 09.06.2020]
5. Content-based Filtering Advantages & Disadvantages, https://developers.google.com/machine-learning/recommendation/content-based/summary, [dostęp 31.05.2020]
6. Content-based Filtering, https://developers.google.com/machine-learning/recommendation/content-based/basics, [dostęp 25.05.2020]
7. Covington P., Adams J., Sargin E., Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems (RecSys ’16), 2016
8. Deldjoo Y., Elahi M., Cremonesi P., Bakhshandegan-Moghaddam F., Caielli A. L. E., How to Combine Visual Features with Tags to Improve the Movie Recommendation Accuracy, Springer – Proceedings of the 17h International conference on Electronic Commerce and Web Technologies, 2016
9. Deldjoo Y., Elahi M., Cremonesi P., Using Visual Features and Latent Factors for Movie Recommendation, ACM RecSys Workshop on New Trends in Content-based Recommender Systems (CBRecSys), ACM RecSys 2016, Massachusetts Institute of Technology (MIT), 2016, http://recsys.deib.polimi.it/mise-en-scene-visual/, [dostęp 11.06.2020]
10. Deldjoo Y., Garzotto F., Elahi M., Piazzolla P., Cremonesi P., Recommending Movies Based on Mise-en-Scene Design, 2016
11. Harper F. M., Konstan J. A., The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS), 2015
12. How Netflix's AI Saves It $1 Billion Every Year, https://www.fool.com/investing/2016/06/19/how-netflixs-ai-saves-it-1-billion-every- year.aspx, [dostęp 14.01.2020]
13. Internet live stats, https://www.internetlivestats.com/one-second/, [dostęp 16.06.2020]
14. Introduction to recommender systems, https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada, [dostęp18.06.2020]
15. Kane F., Building Recommender Systems with Machine Learning and AI, https://www.udemy.com/course/building-recommender-systems- with-machine-learning-and-ai/, [dostęp 10.06.2020]
16. MatPlotLib, https://matplotlib.org/, [dostęp 10.06.2020]
17. MovieLens datasets, https://grouplens.org/datasets/movielens/, [dostęp 09.06.2020]
18. Netflix Prize Leaderboard, https://www.netflixprize.com/leaderboard.html, [dostęp 06.06.2020]
19. Netflix Prize, https://en.wikipedia.org/wiki/Netflix_Prize, [dostęp 07.06.2020]
20. New Epsilon research indicates 80% of consumers are more likely to make a purchase when brands offer personalized experiences, https:// us.epsilon.com/pressroom/new-

epsilon-research-indicates-80-of-consumers-are-more-likely-to-make-a-purchase-when-brands-offer- personalized-experiences, [dostęp 16.06.2020]
21. NumPy, https://numpy.org/, [dostęp 10.06.2020]
22. One-hot encoding, https://en.wikipedia.org/wiki/One-hot, [dostęp 25.05.2020]
23. Pandas, https://pandas.pydata.org/, [dostęp 10.06.2020]
24. Recommender system, https://en.wikipedia.org/wiki/Recommender_system, [dostęp 18.06.2020]
25. Salakhutdinov R., Mnih A., Hinton G., Restricted Boltzmann machines for collaborative filtering. ACM International Conference Proceeding Series, 2007
26. Sarwar B., Karypis G., Konstan J., Reidl J., Item-based collaborative filtering recommendation algorithms - in Proceedings of the 10th international conference on World Wide Web, 2001
27. Sarwar B.M., Karypis G., Konstan J.A., Riedl J., Application of Dimensionality Reduction in Recommender System – A Case Study. ACMWebKDD Workshop, 2000
28. Segaran T., Programing Collective Intelligence: Building Smart Web 2.0 Applications, O'Reilly 2007, s.29-46
29. Serwis IMDB, https://www.imdb.com/, [dostęp 09.06.2020]
30. Serwis TMDB https://www.themoviedb.org/, [dostęp 09.06.2020]
31. Sun Z., Luo N., A new user-based collaborative filtering algorithm combining data-distribution - in Proceedings of the 2010 InternationalConference of Information Science and Management Engineering - Volume 02, 2010
32. SurpriseLib, A Python scikit for recommender systems, http://surpriselib.com/, [dostęp 09.06.2020]
33. Tensorflow, https://www.tensorflow.org/, [dostęp 10.06.2020]
34. The Netflix Recommender System, https://dl.acm.org/doi/pdf/10.1145/2843948, [dostęp 14.01.2020]
35. Wang X., Wang Y., Improving Content-based and Hybrid Music Recommendation using Deep Learning - in Proceedings of the 22nd ACM international conference on Multimedia (MM ’14), 2014
36. YouTube Facts, Figures and Statistics – 2020, https://merchdope.com/youtube-stats/, [dostęp 16.06.2020]