|  |  |
| --- | --- |
| **ZUT_2** | WI_1 |

kierunek studiów: Informatyka

specjalność: Systemy komputerowe zorientowane na człowieka

Praca dyplomowa magisterska

WYKORZYSTANIE UCZENIA GŁĘBOKIEGO W HYBRYDOWYCH SYSTEMACH REKOMENDACYJNYCH

THE USE OF DEEP LEARNING IN HYBRID RECOMMENDER SYSTEMS

Aleksander Rubis

nr albumu: 48105

Opiekun:

dr hab. inż. Piotr Sulikowski

Katedra Inżynierii Systemów Informacyjnych

Szczecin 2023

**Abstrakt**

Niniejsza praca ma na celu omówienie zagadnienia systemów rekomendacyjnych, a w szczególności systemów hybrydowych oraz zbadanie wykorzystania w nich głębokiego uczenia maszynowego. W pracy zostały przedstawione główne typy systemów rekomendacyjnych tj. filtrowanie kolaboracyjne, filtrowanie oparte na treści oraz podejście hybrydowe, a następnie korzystając ze zbioru danych MovieLens, przeprowadzone zostało porównanie działania różnych algorytmów rekomendacyjnych w tym systemu hybrydowego łączącego algorytmy NeuCF i Doc2Vec.

Słowa kluczowe: system rekomendacyjny, wspólna filtracja, filtracja oparta na treści, hybrydowy system rekomendacyjny, uczenie głębokie, uczenie maszynowe, neural collaborative filtering, doc2vec

**Abstract**

The purpose of this work is to discuss the issue of recommendation systems, in particular hybrid systems, and to examine the use of deep machine learning in them. The main types of recommendation systems, such as collaborative filtering, content-based filtering, and hybrid approach are presented, and with the use of the MovieLens dataset, a comparison of the performance of different recommendation algorithms is conducted including hybrid system utilizing NeuCF and Doc2Vec algorithms.

Keywords: recommender system, collaborative filtering, content-based filtering, hybrid recommender system, deep learning, machine learning, neural collaborative filtering, doc2vec

**Spis treści**

[**Wstęp** 3](#_Toc135326904)

[**1.1.** **Uczenie Maszynowe** 6](#_Toc135326905)

[**1.1.1.** **Nadzorowane uczenie maszynowe** 6](#_Toc135326906)

[**1.1.2.** **Nienadzorowane uczenie maszynowe** 6](#_Toc135326907)

[**1.1.4.** **Znaczenie uczenia maszynowego** 7](#_Toc135326908)

[**1.2.** **Sieci neuronowe** 7](#_Toc135326909)

[**1.3.** **Uczenie głębokie** 12](#_Toc135326910)

[**2.** **Systemy rekomendacyjne** 14](#_Toc135326911)

[**2.1.** **Filtracja kolaboracyjna** 19](#_Toc135326912)

[**2.2.** **Filtracja oparta na treści** 22](#_Toc135326913)

[**2.3.** **Inne podejścia** 24](#_Toc135326914)

[**2.3.1.** **Systemy oparte na społeczności** 24](#_Toc135326915)

[**2.3.2.** **Systemy demograficzne** 24](#_Toc135326916)

[**2.3.3.** **Systemy oparte na wiedzy** 24](#_Toc135326917)

[**2.3.4.** **Systemy uwzględniające kontekst** 24](#_Toc135326918)

[**2.4.** **Podejście hybrydowe** 25](#_Toc135326919)

[**2.6.** **Metody ewaluacji systemów rekomendacyjnych** 26](#_Toc135326920)

[**2.7.** **Powtarzalność wyników w badaniach nad systemami rekomendacyjnymi** 29](#_Toc135326921)

[**3.** **Opracowanie systemów rekomendacyjnych** 31](#_Toc135326922)

[**3.1.** **Zbiór MovieLens, badania eksploracyjne i edycja zbioru danych** 31](#_Toc135326923)

[**3.2.** **Referencyjne systemy rekomendacyjne** 36](#_Toc135326924)

[**3.3.** **System rekomendacyjny oparty na treści** 37](#_Toc135326925)

[**3.4.** **System rekomendacyjny filtracji kolaboracyjnej** 42](#_Toc135326926)

[**3.5.** **Hybrydowy system** 43](#_Toc135326927)

[**4.** **Analiza działania opracowanych systemów rekomendacyjnych** 44](#_Toc135326928)

[**5.** **Podsumowanie** 51](#_Toc135326929)

**Wstęp**

W obecnych czasach obserwuje się gwałtowny rozwój technologii informacyjnych, a co za tym idzie wzrost ilości informacji dostępnych dla użytkownika. Pomimo tego wzrostu, zdolności kognitywne człowieka nie zwiększyły się. Opisywane już latach 70. zjawisko „przeciążenia informacją” (ang. „information overload”) staje się coraz większym problemem. Oznacza ono sytuację, w której pod wpływem natłoku informacji zdolności analizowania przekazu i podejmowania decyzji zostają zakłócone. W takich warunkach uzyskanie dostępu do treści, które rzeczywiście byłyby dla użytkownika użyteczne staje się utrudnione bądź nawet niemożliwe. Codziennie, w serwisie społecznościowym Twitter publikowanych jest 500 milionów „tweetów”, a na platformie YouTube jest w tym samym czasie zamieszczane ponad 3 mln nowych materiałów wideo o łącznej długości prawie 700.000 godzin czyli niemal 80 lat nagrań [1][2]. Wskazuje się również na związek przeciążenia informacją z rozprzestrzenianiem się tzw. „fake newsów” oraz działaniami propagandowymi [3][4].

Istnieje więc potrzeba posiadania rozwiązania algorytmicznego będącego w stanie skutecznie filtrować dostępną zawartość i dostarczać użytkownikowi najbardziej odpowiednich dla niego w danym momencie treści. Odpowiedź na te problemy stanowią systemy rekomendacyjne (SR), które odgrywają coraz większą rolę w znajdowaniu przez użytkowników preferowanych przez nich pozycji spośród ogromnej ilości treści [5]. W publikacji „Information Overload and Usage of Recommendations” autorzy pokazują, że w przeprowadzonym badaniu, w którym uczestnicy mieli za zadanie dokonać wyboru laptopa w środowisku przeciążenia informacją dokonywali oni trafniejszych i pewniejszych wyborów, gdy mieli możliwość skorzystania z systemów rekomendacyjnych [6].

Systemy te generują spersonalizowane listy przedmiotów, zazwyczaj posortowane od najbardziej do najmniej odpowiadającego użytkownikowi. Używają do tego różnych typów informacji z wielu heterogonicznych źródeł, takich jak ocenione wcześniej przedmioty/pozycje, kliknięte adresy URL, złożone zamówienia, napisane komentarze, czy dane demograficzne [5]. Możliwość dostarczenia przez SR odpowiadających użytkownikowi treści z dużą dokładnością i wydajnością sprawiła, że zagadnienie systemów rekomendacyjnych stało się obiektem zainteresowań społeczności naukowej jak i przemysłu informatycznego [7].

Systemy rekomendacyjne odgrywają ważną rolę w najpopularniejszych serwisach internetowych takich jak: YouTube, Google, Amazon, Netflix, Tripadvisor, czy IMDb. Wymienione platformy jak i wiele innych stale opracowują i wdrażają coraz bardziej zaawansowane systemy SR jako część usług które świadczą swoim użytkownikom. Przykładowo Netflix, serwis strumieniujący filmy i seriale, przyznał nagrodę 1 miliona dolarów zespołowi, któremu w zorganizowanym w 2009 roku konkursie „Netflix Prize” udało się opracować SR który osiągnął dokładność predykcji wyższą o 10.5% od systemu używanego przez platformę Netflix [5][8].

SR dzielą się głównie na trzy kategorie: systemy oparte na treści, systemy filtrowania kolaboracyjnego i systemy hybrydowe. Systemy oparte na treści rekomendują pozycje podobne do tych, które użytkownik polubił wcześniej bądź wszedł z nimi w interakcję. Systemy filtrowania kolaboracyjnego generują rekomendacje na podstawie preferencji użytkowników, którzy mają podobne gusta z użytkownikiem docelowym. Systemy hybrydowe łączą co najmniej 2 systemy rekomendacji w systematyczny sposób i zwykle posiadają zalety każdego ze składowych systemów.

Od pewnego czasu uczenie głębokie - UG (ang. „deep learning”) wywołuje znaczne zainteresowanie w wielu dziedzinach badawczych, takich jak rozpoznawanie obrazów czy przetwarzanie języka naturalnego zarówno ze względu na swoją doskonałą wydajność, jak i zdolność uczenia się reprezentacji cech od zera. W ostatnich latach UG rozkwitać zaczyna również w takich obszarach, jak wyszukiwanie informacji i systemy rekomendacyjne [13].

Celem niniejszej pracy magisterskiej jest omówienie zagadnienia systemów rekomendacyjnych, ich wariantu hybrydowego, a także zastosowania w nich metod uczenia głębokiego. Przedstawione zostaną najpopularniejsze algorytmy rekomendacji proponowane przez autorów i wykorzystywane w przemyśle, omówione zostaną ich wady i zalety. W części praktycznej pracy porównane zostanie działanie podstawowych typów algorytmów rekomendacyjnych.

Praca składa się z 5 rozdziałów:

W rozdziale pierwszym omówione zostaną zagadnienia uczenia maszynowego, sieci neuronowych i uczenia głębokiego, przedstawione zostaną mechanizmy ich działania, podstawowe pojęcia, wady i zalety tego rozwiązania oraz jego zastosowania.

W rozdziale drugim szczegółowo omówione zostaną systemy rekomendacyjne. Zostaną przedstawione rodzaje SR, ich wady i zalety, aspekty zastosowania każdego z rozwiązań w konkretnych przypadkach oraz metody oceniania SR. W tym rozdziale omówione zostanie również wykorzystanie uczenia głębokiego w SR.

W rozdziale trzecim przeprowadzone zostaną badania eksploracyjne zbioru danych MovieLens, który zostanie wykorzystany do trenowania i testowania algorytmów rekomendacyjnych. Następnie opisane zostaną szczegóły implementacji wybranych algorytmów rekomendacyjnych.

W rozdziale czwartym znajdzie się analiza działania zaimplementowanych algorytmów, porównane zostaną miary oceny działania algorytmów.

Rozdział piąty stanowić będzie podsumowanie pracy.

1. **Uczenie maszynowe i uczenie głębokie**
   1. **Uczenie Maszynowe**

Uczenie maszynowe to poddziedzina sztucznej inteligencji, która zajmuje się badaniem i budowaniem algorytmów, które poprawiają swoją skuteczność automatycznie, poprzez zdobywanie doświadczenia na zbiorze przykładów. Przykłady te mogą pochodzić z natury, być uzyskane i skatalogowane ręcznie przez ludzi lub też wygenerowane przez inny algorytm [10].

Uczenie maszynowe można również zdefiniować, jako proces rozwiązywania praktycznego problemu poprzez zgromadzenie zbioru danych, a następnie algorytmicznego zbudowania modelu statystycznego na podstawie tego zbioru. Uzyskany w ten sposób model, może być następnie wykorzystany do rozwiązywanie założonych problemów [10].

Można je podzielić na trzy najważniejsze rodzaje, a mianowicie nadzorowane uczenie maszynowe, nienadzorowane uczenie maszynowe i uczenie ze wzmocnieniem [10].

* + 1. **Nadzorowane uczenie maszynowe**

W uczeniu nadzorowanym zbiór danych jest zbiorem oznaczonych przykładów . Każdy element spośród nazywany jest wektorem cech. Wektor cech to wektor, w którym każdy wymiar j = 1,…,D zawiera wartość, która w jakiś sposób opisuje przykład. Wartość ta nazywana jest cechą i jest oznaczona jako . Na przykład, jeśli każdy przykład w danej kolekcji reprezentuje osobę, pierwsza cecha, , może zawierać wzrost w centymetrach, druga cecha, , może zawierać wagę w kilogramach, może zawierać płeć i tak dalej. Dla wszystkich przykładów w zbiorze danych, cecha na pozycji w wektorze cech zawsze zawiera ten sam rodzaj informacji. Etykieta może być elementem należącym do skończonego zbioru klas , liczbą rzeczywistą, albo bardziej złożoną strukturą, taką jak wektor, macierz, drzewo lub graf. Klasę można rozumieć jako kategorię, do której należy przykład. Celem nadzorowanego uczenia jest wykorzystanie zbioru danych do stworzenia modelu który na wejściu przyjmuje wektor cech, a na wyjściu określa etykietę dla danego przykładu [10].

* + 1. **Nienadzorowane uczenie maszynowe**

W uczeniu nienadzorowanym zbiór danych jest zbiorem nieoznaczonych przykładów . Ponownie, jest wektorem cech, a celem algorytmu jest stworzyć model, który pobiera wektor cech jako dane wejściowe i przekształca go w inny wektor albo do wartości, która może być użyta do rozwiązania problemu. Na przykład, w analizie skupień (ang. data clustering) model zwraca identyfikator klastra dla każdego wektora cech w zbiorze danych. W przypadku redukcji wymiarowości (ang. „dimensionality reduction”) wynik modelu jest wektorem cech, który ma mniej cech niż wejściowy .

* + 1. **Uczenie ze wzmocnieniem**

Uczenie ze wzmocnieniem to szkolenie modeli uczenia maszynowego podejmowania sekwencji decyzji. Algorytm uczy się osiągać cel w niepewnym, potencjalnie złożonym środowisku. W uczeniu się ze wzmocnieniem sztuczna inteligencja staje w obliczu sytuacji przypominającej grę. Komputer wykorzystuje metodę prób i błędów, aby znaleźć rozwiązanie problemu. Aby maszyna zrobiła to, czego chce programista, sztuczna inteligencja otrzymuje nagrody lub kary za akcje, które wykonuje. Jego celem jest maksymalizacja całkowitej nagrody. Chociaż projektant ustala politykę nagradzania – czyli zasady gry – nie daje modelowi żadnych wskazówek ani sugestii, jak rozwiązać grę. To od modelu zależy, jak wykonać zadanie, aby zmaksymalizować nagrodę, zaczynając od całkowicie losowych prób, a kończąc na wyrafinowanej taktyce [10].

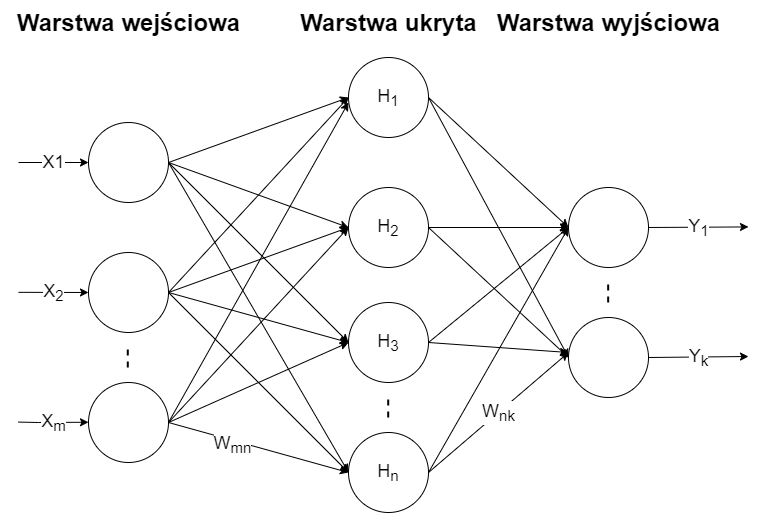
* + 1. **Znaczenie uczenia maszynowego**

Algorytmy uczenia maszynowego są wykorzystywane w wielu różnych zastosowaniach, takich jak finanse, medycyna, rozpoznawanie mowy, filtrowanie wiadomości e-mail i widzenie komputerowe, gdzie opracowanie konwencjonalnych algorytmów do wykonywania potrzebnych zadań jest trudne lub niemożliwe [10].

* 1. **Sieci neuronowe**

Sztuczne sieci neuronowe (SSN) są metodą przetwarzania informacji inspirowaną działaniem ludzkiego mózgu. Wykorzystywane są do zadań takich jak regresja, grupowanie, klasyfikacja i rozpoznawanie wzorców. Zainspirowane ludzkim mózgiem SSN są w stanie uczyć się i uogólniać wiedzę na podstawie zdobytego doświadczenia. SSN uczą się na przykładach i wychwytują subtelne funkcjonalne związki w zbiorze danych, nawet jeśli leżące u ich podstaw relacje są nieznane lub trudne do opisania [10].

SSN jest to nadzorowany bądź nienadzorowany system uczenia się zbudowany z dużej liczby prostych jednostek – neuronów (analogicznie do neuronów biologicznych występujących w mózgu) nazywanych również perceptronami. Każde połączenie wychodzące może przekazywać sygnał do innego neuronu. Neuron odbierający może przetwarzać sygnały, a następnie przesyłać je dalej, do podłączonych do niego neuronów. Neurony mogą mieć przypisany im stan, zazwyczaj reprezentowany przez liczby rzeczywiste, zwykle od 0 do 1. Neurony i synapsy mogą również posiadać przypisane wagi (w), które odzwierciedlają jak istotne są te połączenia dla wyniku końcowego. W większości sieci neuronowych, neurony zorganizowane są w warstwy. Różne warstwy mogą wykonywać różne rodzaje przekształceń na swoich wejściach. Sygnały zazwyczaj przechodzą od warstwy pierwszej (wejściowej) do ostatniej (wyjściowej). Płytka sieć ma tylko trzy warstwy neuronów: warstwę wejściową, pojedynczą warstwę ukrytą i warstwę wyjściową. Tymczasem w głębokiej sieci neuronowej (ang. „deep neural network”) wykorzystać można dwie lub więcej warstw ukrytych. Dzięki temu sieci głębokie są dokładniejsze w rozwiązywaniu złożonych problemów. Poniższy rysunek przedstawia prostą sieć neuronową [10].

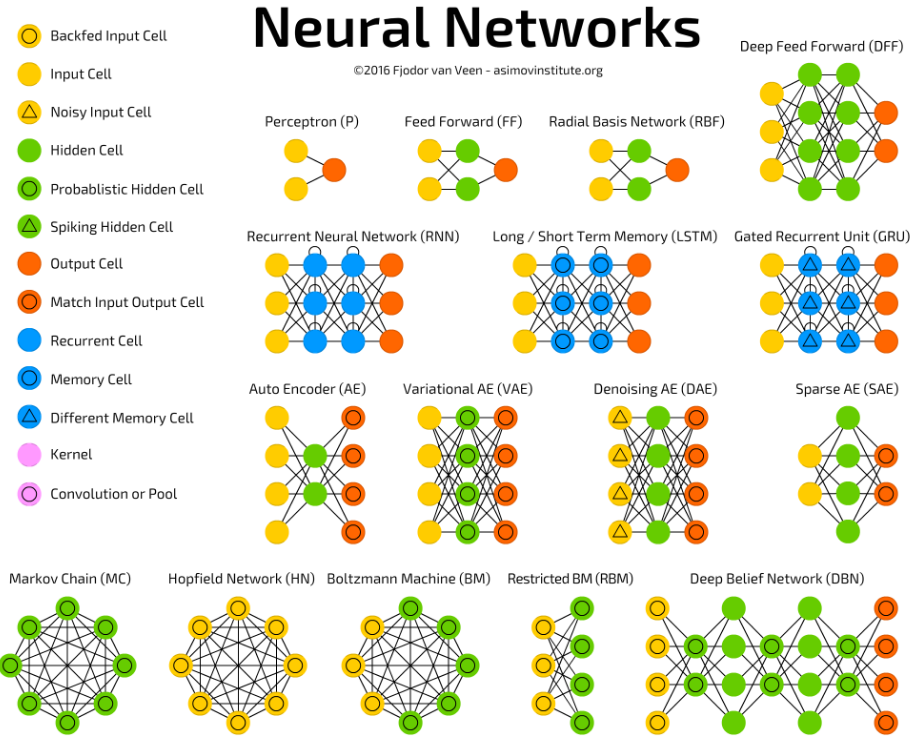


Rys. Przykładowa sieć neuronowa z warstwą wejściową, wyjściową i jedną warstwą ukrytą. Źródło: Opracowanie własne.

Jednymi z głównych typów sztucznych sieci neuronowych są sieci neuronowe jednokierunkowe (ang. „feedforward networks) tj. niewykorzystujące sprzężenia zwrotnego oraz sieci ze sprzężeniem zwrotnym ( ang. „feedback networks”). Sieć neuronowa typu feedforward to sieć, która nie jest rekurencyjna. Neurony w takiej sieci są połączone tylko z neuronami w następnej warstwie i nie stanowią grafu z cyklami. W tego typu sieci neuronowej sygnały wędrują tylko w jednym kierunku, tj. w kierunku warstwy wyjściowej. [11]

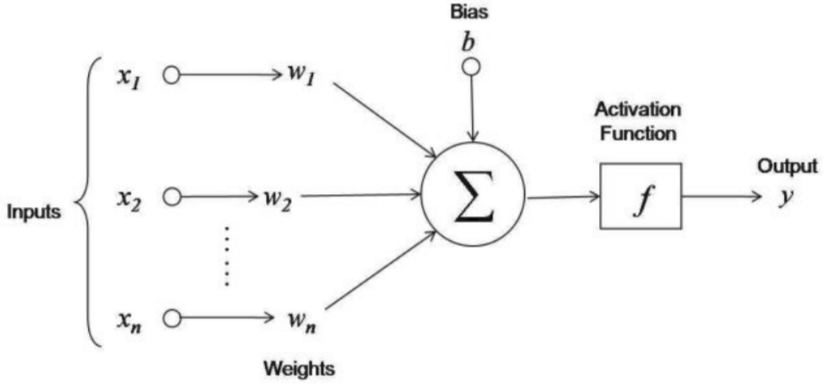
Sieci neuronowe ze sprzężeniem zwrotnym zawierają cykle. Sygnały przemieszczają się w obu kierunkach, wprowadzając pętle w sieci. Cykle sprzężenia zwrotnego mogą powodować zmianę zachowania sieci w czasie na podstawie jej danych wejściowych. Sieć neuronowa ze sprzężeniem zwrotnym, znana jest również jako rekurencyjna sieć neuronowa. [11]

Sieci neuronowe są gałęzią uczenia maszynowego która rozwija się w bardzo szybkim tempie. Rocznie publikowane są na świecie tysiące prac naukowych poruszających to zagadnienie. Istnieje wiele różnych rodzajów i podrodzajów SSN realizujących określone zadania i mogących być łączone w jeszcze bardziej skomplikowane modele. Poniższy rysunek przedstawia schematy wybranych modeli SSN.



Rys. Przykłady sieci neuronowych. Źródło: https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

* + 1. **Proces uczenia się**



Rys. Aktywacja neuronu. Źródło [12]

Na samym początku procesu uczenia zainicjowane zostają wagi, a wejścia są wprowadzane do sieci jako pierwsza aktywacja. Następnie każdy neuron oblicza sumę ważonych aktywacji wszystkich neuronów w poprzedniej warstwie z którymi jest połączony, a następnie dodawana jest wartość biasu [10]. Po tej funkcji (), wykonywana zostaje funkcja aktywacji , w wyniku czego otrzymywane jest wyjście, będące aktywacją neuronów następnej warstwy. Proces ten wykonują wszystkie neurony we wszystkich warstwach aż do warstwy wyjściowej, będącej wynikiem sieci.

Gdzie to wyjście neuronu *i* warstwy , natomiast to liczba neuronów w poprzedniej warstwie. jest wagą połączenia między neuronem w warstwie a neuronemw warstwie . to funkcja aktywacji.jest biasem neuronu w warstwie .

* + 1. **Funkcja aktywacji:**

Funkcja aktywacji jest równaniem matematycznym, które z sumy ważonej wejść i tzw. „biasu”, określa wyjście neuronu. Funkcja ta dyktuje, czy neuron może zostać odpalony czy nie.

Istnieje wiele funkcji aktywacji, jednymi z najbardziej popularnych są [3][5]:

* Funkcja skokowa (ang. „threshold function”). Jest nieciągłą funkcją przyjmującą wartość 1 dla argumentów niedodatnich oraz wartość 0 dla pozostałych.
* Funkcja sigmoidalna jest nieliniową funkcją aktywacji używaną głównie w sieciach jednokierunkowych. Jest funkcją rzeczywistą, różniczkowalną i ograniczoną od 0 do 1. Można ją opisać wzorem:
* ReLu oznacza rektyfikowaną jednostkę liniową (ang. „rectified linear unit”). Funkcja ta jest przedziałową funkcją liniową zwracającą wartość wejściową jeśli jest to wartość dodatnia lub 0 jeśli wartość jest negatywna. Jest najczęściej używaną funkcją aktywacji w sieciach neuronowych. Można ją opisać wzorem:
  + 1. **Funkcja kosztu:**

Jest to funkcja mierząca efektywność modelu uczenia maszynowego dla określonych danych. Funkcja kosztu kwantyfikuje błąd między wartościami przewidywanymi a wartościami oczekiwanymi i przedstawia go w postaci pojedynczej liczby rzeczywistej. W zależności od problemu funkcję kosztu można formować na wiele różnych sposobów. Celem funkcji kosztów może być:

* Maksymalizacja - wtedy wartość, jaką zwraca, nazywana jest nagrodą. Celem jest znalezienie wartości parametrów modelu, dla których zwracana liczba jest jak największa.
* Minimalizacja - wtedy zwracana wartość nazywana jest zwykle kosztem, stratą lub błędem. Celem jest znalezienie wartości parametrów modelu, dla których funkcja kosztu zwraca możliwie najmniejszą liczbę.

Wykorzystując funkcję kosztu w optymalizacji sieci neuronowej celem staje się zazwyczaj minimalizacja wartości błędu modelu, z tego też powodu funkcja ta bywa nazywana funkcją straty (ang. „loss function”) lub stratą (ang. „loss”).

Istnieje wiele różnych funkcji kosztu, a wybór konkretnej funkcji zależy od tego czy model ma za zadanie rozwiązywać problem regresji bądź klasyfikacji. Jednymi z najczęściej wykorzystywanych funkcji kosztu są entropia krzyżowa (ang. „Cross Entropy”), średni błąd absolutny (MAE – ang. „mean absolute error”), błąd średniokwadratowy (MSE – ang. „mean squared error”) i pierwiastek z błędu średniokwadratowego (RMSE - root mean squared terror)

Każda z tych funkcji znajduje zastosowanie w określonym rodzaju problemu. Entropia krzyżowa jest używana w problemach klasyfikacji, podczas gdy MAE, MSE i RMSE są używane do problemów regresji.

Posiadając narzędzie jakim jest funkcja kosztu, celem staje się znalezienie takiego zestawu wag, który zminimalizuje wartość błędu *E* w całym zestawie uczącym. Wpływ indywidualnej wagi na funkcję kosztu może zostać określony poprzez propagację wsteczną.

* + 1. **Propagacja wsteczna:**

W celu uzyskania optymalnego zestawu wag wykonuje się wsteczną propagację (ang. „backward propagation” lub „backpropagation”), poruszając się niejako do tyłu, z predykcji sieci do neuronów które wygenerowały tą predykcję, i zmieniając wagi. Wsteczna propagacja jest metodą obliczania pierwszej pochodnej funkcji błędu w odniesieniu do każdej wagi sieci w celu znalezienia wag, które sprowadzą funkcję kosztu do minimum. Odbywa się to za pomocą metody gradientowej (ang. „gradient descent” lub GD) która polega na iteracyjnym wykonywaniu małych kroków w kierunku przeciwnym do gradientu (lub przybliżonego gradientu) funkcji w bieżącym punkcie, ponieważ jest to kierunek najbardziej stromego spadku wartości. Analogicznie, krok w kierunku gradientu prowadzi do lokalnego maksimum tej funkcji, a procedura ta nazywana jest wówczas wznoszeniem gradientowym.

Algorytm propagacji wstecznej można podzielić na następujące kroki:

* Przejście do przodu ( od warstwy wejściowej do warstwy wyjściowej)
* Propagacja wsteczna do warstwy wyjściowej
* Propagacja wsteczna do warstw ukrytych
* Zaktualizowanie parametrów sieci

Algorytm zostaje zatrzymany kiedy wartość funkcji kosztu zostanie wystarczająco zmniejszona.

W klasycznym GD parametry są aktualizowane po przejściu przez cały zestaw treningowy

propagacji do przodu i do tyłu ale zwykle nie jest to praktyczne gdy ma się do czynienia z dużymi zestawami danych. Zazwyczaj partia (z ang. „batch”) danych jest przetwarzana w jednym dużym przejściu do przodu, a następnie na zagregowanym wyniku wykonywana jest propagacja wsteczna. Wielkość partii to ważny hiperparametr (parametr sieci ustawiany przed procesem uczenia, wpływający na wartości parametrów podczas uczenia)zmieniany w celu uzyskania najlepszych wyników. Przeprowadzenie całości uczenia przez proces wstecznej propagacji jest nazywany epoką. Stochastyczny spadek gradientu ( z ang. „Stochastic Gradient Descent”) (SGD) wykonuje GD z tylko jednym przykładem z zestawu na krok aby uniknąć redundancji. Proces wykonywania GD na partii danych na krok nazywany jest „Mini-Batch Gradient Descent”. Mini-BatchGD wykonuje przybliżenie gradientu z partii danych. W związku z tym, dzięki zaletom zarówno GD, jak i SGD, Mini-BatchGD pozwala na otrzymanie dokładniej oszacowanych parametrów w mniejszej liczbie kroków i znacznie szybciej [9][10].

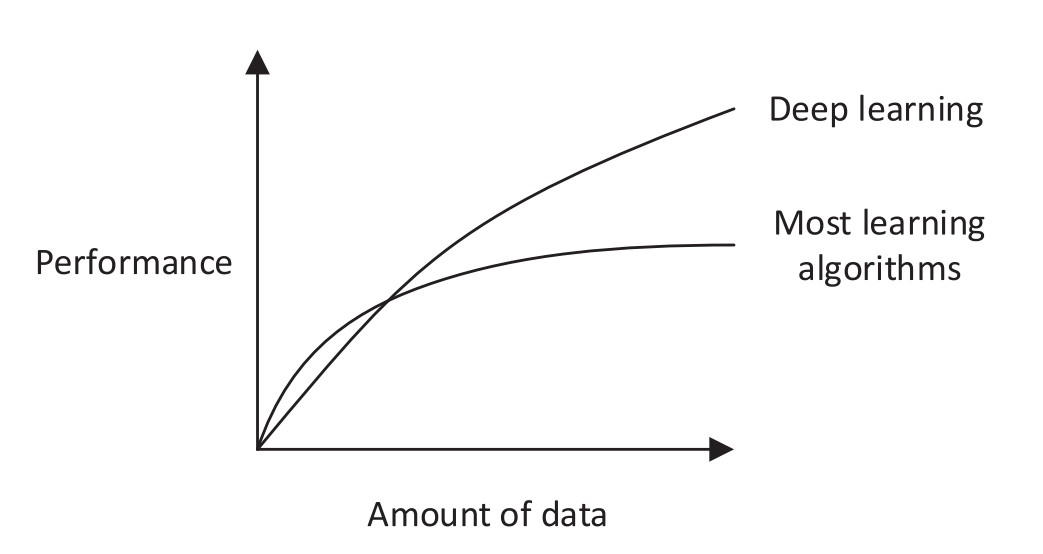
* + 1. **Przetrenowanie i niedouczenie**

Gdy sieć neuronowa dobrze sobie radzi z przykładami ze zbioru treningowego, ale nie potrafi wystarczająco zgeneralizować zdobytą wiedzę i odpowiednio sklasyfikować inne, nieznane jej wcześniej przykłady, mówi się, że sieć jest przeuczona lub też nadmiernie dopasowana. Odznacza się ona wtedy niskim obciążeniem (biasem, czyli różnicą między wartością przewidywaną i rzeczywistą) i wysoką wariancją ( wrażliwością na małe wahania w zbiorze testowym). Można temu zapobiec, ponownie trenując tę samą sieć z innymi początkowymi wartościami wag, sprawdzaniem wartości błędu po każdej iteracji, zatrzymywaniem procesu uczenia, gdy sieć zaczyna się nadmiernie dopasować do danych lub przez dodanie tzw. „dropout’u” (hiperparemetru, losowo zatrzymującego proces uczenia dla części neuronów w każdej iteracji). Z drugiej strony, sieć neuronowa może być również niedouczona czy też niedopasowana, co oznacza, że nie zwraca prawidłowych wyników nawet dla zbioru treningowego. Charakteryzuje się wtedy wysokim obciążeniem i niską wariancją. Niedopasowania można uniknąć, dodając więcej próbek treningowych [5].

* 1. **Uczenie głębokie**

Uczenie głębokie - UG (ang. „deep learning”) jest podzbiorem technik uczenia maszynowego opartych na sztucznych sieciach neuronowych. Techniki uczenia głębokiego poczyniły duże postępy w rozwiązywaniu problemów, które przez lata uważane były przez badaczy i praktyków uczenia maszynowego za trudne bądź nierozwiązywalne. Okazało się, że UG bardzo dobrze radzi sobie z odkrywaniem zawiłych struktur w danych wielowymiarowych, dzięki czemu znajduje zastosowanie w wielu dziedzinach nauki, biznesu i administracji. Oprócz bicia rekordów w rozpoznawaniu obrazów i mowy pobiło inne techniki uczenia maszynowego w przewidywaniu aktywności potencjalnych cząsteczek leków, analizie danych z akceleratorów cząstek, symulacji biologicznych układów neuronowych i przewidywaniu wpływu mutacji w niekodującym DNA na ekspresję genów i choroby. Co więcej, głębokie uczenie przyniosło niezwykle obiecujące wyniki w przypadku różnych zadań związanych z rozumieniem języka naturalnego, w szczególności klasyfikacji tematów, analizy nastrojów, odpowiadania na pytania i tłumaczenia języka [14].

Uczenie głębokie wykorzystuje wiele warstw do stopniowego wydobywania cech wyższego rzędu z danych wejściowych; każdy poziom uczy się przekształcać swoje dane wejściowe w nieco bardziej abstrakcyjną i złożoną reprezentację. Przykładowo w aplikacji do rozpoznawania obrazów, surowe dane wejściowe mogą być macierzą pikseli; pierwsza warstwa reprezentacyjna może abstrahować piksele i rozpoznawać krawędzie; druga warstwa może łączyć i kodować układy krawędzi; trzecia warstwa może kodować nos i oczy; a czwarta warstwa może rozpoznać, że obraz zawiera twarz. Co ważne, proces głębokiego uczenia może samodzielnie „nauczyć się”, które cechy optymalnie umieścić w którym poziomie. Nie eliminuje to potrzeby ręcznego dostrajania; na przykład zróżnicowanie liczby i wielkości warstw może zapewnić różne stopnie abstrakcji [10].



Rys. Porównanie wydajności modelów wykorzystujących uczenie głębokie z większością innych modeli uczenia maszynowego. Wraz ze wzrostem dostępnych danych głębokie sieci neuronowe zwiększają wydajność podczas gdy inne algorytmy osiągają pewien pułap i nie są w stanie skorzystać z większej ilości danych uczących.

Źródło: „Deep learning in business analytics and operations research: Models, applications and managerial implications” Mathias Kraus, Stefan Feuerriegel, Asil Oztekinb, 2019

1. **Systemy rekomendacyjne**

Systemy rekomendacyjne, lub systemy rekomendacji (czasami zastępuje się słowo "system" synonimem takim jak platforma lub silnik), są podklasą systemów filtrowania informacji, które dostarczają sugestii dotyczących pozycji, które są najbardziej odpowiednie dla konkretnego użytkownika. Zazwyczaj sugestie odnoszą się do różnych procesów decyzyjnych, takich jak jaki produkt kupić, jakiej muzyki słuchać, lub jakie wiadomości online czytać. Systemy rekomendacji są szczególnie przydatne, gdy użytkownik musi wybrać pozycję z potencjalnie przytłaczającej liczby pozycji, które może zaoferować usługa [5][10].

Początków badań nad tym zagadnieniem doszukiwać się można w 1979 roku, kiedy to Elaine Rich stworzyła nieświadomie pierwsze systemy rekomendacyjne Grundy. Szukała sposobu na polecenie użytkownikowi książki, która mogłaby mu się spodobać. Jej pomysł polegał na stworzeniu systemu, który zadawałby użytkownikowi określone pytania i przypisywałby mu stereotypy w zależności od jego odpowiedzi. W zależności od stereotypu użytkownika, otrzymywałby on rekomendację książki, która mogłaby mu się spodobać [14][15].

Pierwsza faktyczna wzmianka o systemie rekomendującym pojawiła się w raporcie technicznym w 1990 roku autorstwa Jussi Karlgrena z Uniwersytetu Kolumbii jako "cyfrowa półka na książki", a następnie została zaimplementowana na dużą skalę i dopracowana w raportach technicznych i pracach publikowanych od 1994 roku przez Jussi Karlgrena oraz grupę badawczą kierowaną przez Pattie Maes, Willa Hill i Paulę Resnick, których praca nad GroupLens została nagrodzona w 2010 roku nagrodą ACM Software Systems Award. W 1992 Goldberg i in. zaproponowali system Tapestry który uważany jest za pierwszy system wspólnej filtracji korzystający z ocen użytkowników [14][15].

W 2003 roku Montaner przedstawił pierwszy przegląd systemów rekomendacyjnych z perspektywy agenta inteligentnego. Adomavicius w 2005 zapewnił nowy, alternatywny przegląd systemów rekomendacyjnych. W 2004 roku Herlocker dostarcza dodatkowego przeglądu technik oceny dla systemów rekomendacyjnych. Beel i in. omówili problemy ewaluacji systemów, dostarczyli również przeglądów literatury na temat dostępnych systemów rekomendacyjnych dla artykułów badawczych i istniejących wyzwań [15].

Już przed 2005 rokiem branża systemów rekomendacyjnych zdominowała została przez techniki wspólnej filtracji, takie jak wspólna filtracja typu użytkownik- użytkownik i przedmiot-przedmiot oraz faktoryzacja macierzy. W 2009 roku, po zakończeniu konkursu Netflix Prize, którego zwycięzcą został algorytm zespołu BellKor's Pragmatic Chaos w dużym stopniu wykorzystującym dekompozycję SVD (z ang. Singular Value Decomposition) modele faktoryzacji macierzy były badane bardzo intensywnie. Z powodu szybkiego rozwoju badań i zastosowań komercyjnych w dziedzinie systemów rekomendacyjnych, społeczność naukowa postanowiła zorganizować pierwszą konferencję ACM Recommender Systems w 2007 roku. Obecnie, ACM RecSys stała się jedną z najważniejszych corocznych konferencji naukowych, które skupiają się na badaniach nad systemami rekomendacyjnymi [14].

Od 2016 roku modele rekomendacyjne oparte na głębokich sieciach neuronowych pojawiły się zarówno w środowisku akademickim, jak i komercyjnym. W przypadku systemów rekomendacyjnych wykorzystywanych w przemyśle, modele Wide&Deep i DeepFM zostały z sukcesem wdrożone do poprawy rekomendacji aplikacjach Google Play oraz Huawei App Market. Modele YouTubeDNN i correct-sfx zostały wykorzystane do zwiększenia dokładności rekomendacji wideo. Modele DIN i DIEN zostały zaproponowane do modelowania sekwencyjnych informacji, takich jak zainteresowania użytkowników z mechanizmem uwagi. Wang i in. zaproponowali modele DCN i DCN-V2 do automatycznego i efektywnego uczenia się ograniczonej liczby predykcyjnych interakcji cech. W środowisku akademickim naukowcy również zaproponowali ważne algorytmy rekomendacyjne oparte na sieciach neuronowych, takie jak FNN, PNN, NeuralCF, NFM, CVAE [14].

W najprostszej formie, rekomendacje są dostarczane w postaci listy z rankingiem produktów. Aby utworzyć taki ranking, systemy rekomendacyjne starają się przewidzieć najlepsze produkty lub usługi dla danego użytkownika, opierając się na jego preferencjach. Aby zrealizować to zadanie, systemy rekomendacyjne zbierają informacje o upodobaniach użytkowników, które są wyrażone jawnie, na przykład poprzez oceny produktów, lub wywnioskowane przez interpretację działań użytkownika podczas interakcji z systemem. Przykładowo, system rekomendacyjny może uznać zakup określonego przedmiotu za objaw pewnych konkretnych upodobań bądź cech użytkownika.

Rozwój systemów rekomendacyjnych wynikał z dość prostego spostrzeżenia: ludzie często polegają na rekomendacjach innych w podejmowaniu rutynowych, codziennych decyzji. Na przykład, gdy wybierają książkę do przeczytania, często zwracają się do swoich znajomych; pracodawcy polegają na listach rekomendacyjnych podczas rekrutacji, a w przypadku wyboru filmu do obejrzenia, osoby często czytają i polegają na recenzjach filmowych autorstwa krytyków filmowych lub pytają swoich przyjaciół o rekomendacje. W celu naśladowania tego zachowania, pierwsze systemy rekomendacyjne stosowały algorytmy w celu wykorzystania rekomendacji produkowanych przez społeczność użytkowników i dostarczania tych rekomendacji "aktywnemu" lub "docelowemu" użytkownikowi szukającemu sugestii. Rekomendacje dotyczyły przedmiotów, które podobni użytkownicy lub ci o podobnych gustach lubili. Takie podejście nazywa się filtrowaniem kolaboracyjnym lub systemem wspólnej filtracji i opiera się ono na założeniu, że jeśli dany użytkownik zgodził się w przeszłości z pewnymi użytkownikami, to inne rekomendacje pochodzące od tych podobnych użytkowników powinny być istotne i interesujące dla danego użytkownika [5][7].

W pracy „Evaluating Collaborative Filtering Recommender Systems”, która stała się klasycznym punktem odniesieniem w tej dziedzinie [5]. Herlocker i in. wymieniają 11 popularnych zadań stawianych przed systemem rekomendacyjnym. Są to [16]:

* Znalezienie odpowiadających przedmiotów: Przedstawienie użytkownikowi rankingu przedmiotów, które prawdopodobnie uznałby za najbardziej istotne, zazwyczaj posortowanych od najbardziej preferowanego przedmiotu do najmniej. Jest to zadanie najczęściej stawiane przed systemami rekomendacyjnymi.
* Znalezienie wszystkich odpowiadających przedmiotów: Zarekomendowanie wszystkich elementów, które mogą zaspokoić potrzeby użytkownika, w przypadkach których nie wystarczy znaleźć tylko kilka z nich. Jest to szczególnie prawdziwe, gdy liczba elementów jest stosunkowo niewielka lub gdy SR ma znaczenie kluczowe, takie jak w zastosowaniach medycznych lub finansowych. W tych sytuacjach, użytkownik zyskuje możliwość dokładnego przeanalizowania wszystkich możliwości.
* Adnotacja w kontekście: W zależności od preferencji użytkownika, w istniejącym kontekście, na przykład w liście przedmiotów, należy wybrać niektóre z nich. Przykładowo, system rekomendacji telewizyjnej może opatrzyć adnotacjami programy telewizyjne wyświetlane w elektronicznym przewodniku po programie (EPG), które warto obejrzeć.
* Rekomendacja sekwencji: Zamiast skupiać się na generowaniu pojedynczych rekomendacji, tworzy się całą sekwencję kolejnych rekomendacji przedmiotów. Typowe przykłady obejmują rekomendowanie kolejnej atrakcji do odwiedzenia czy serialu telewizyjnego do obejrzenia.
* Rekomendacja zestawu: Przedstawienie użytkownikowi grupy przedmiotów które pasują do siebie. Przykładowo w sklepie internetowym, przy zakupie komputera, zaproponowanie użytkownikowi myszki i klawiatury.
* Tylko przeglądanie: Użytkownik może jedynie przeglądać katalog dostępnych przedmiotów bez intencji dokonania wyboru konkretnego z nich. Zadaniem systemu rekomendacyjnego może być tutaj nakierowanie na przedmioty które będą miały szanse trafić w preferencje użytkownika
* Znalezienie SR godnego zaufania: Czasami użytkownicy mogą nie ufać działaniu systemu rekomendacyjnego i w związku z czym chcieć testować jego działanie co system powinien im umożliwiać
* Budowanie profilu: Możliwość wprowadzenia przez użytkownika informacji do systemu rekomendacyjnego na temat tego, co lubi a czego nie. Jest to fundamentalne zadanie, które jest niezbędne do zapewnienia spersonalizowanych rekomendacji. Jeśli system nie ma konkretnej wiedzy na temat danego użytkownika, to może jedynie dostarczyć rekomendacje, które zostałyby dostarczone dla "uśrednionego" użytkownika.
* Wyrażanie siebie: Niektórym użytkownikom może nie zależeć na uzyskaniu rekomendacji, a na wyrażeniu swoich opinii i preferencji poprzez ocenianie przedmiotów. Umożliwienie im tego, może prowadzić do większej chęci do dalszej interakcji z serwisem oraz wprowadzania do niego dodatkowych danych.
* Pomaganie innym: Niektórzy użytkownicy są skłonni do zapewniania pomocy innym. Poprzez np. ocenianie przedmiotów czy dodawanie komentarzy czują się oni przydatni dla społeczności i spełnieni. Podobnie jak wyrażanie siebie to zadanie działa pozytywnie na działanie serwisu, może też być wyjątkowo przydatne w przypadku SR używanego przez użytkowników rzadko.
* Wywieranie wpływu: Motywem niektórych osób korzystających z danego SR może być chęć wpłynięcia na innych. Zdarzyć się może nawet, że złośliwi użytkownicy mogą chcieć wykorzystać sposób działania danego SR do wypromowania lub ukrycia konkretnego przedmiotu.

Wziąwszy pod uwagę, że przed systemami rekomendacyjnymi stawiać można tak wiele zróżnicowanych zadań, konieczne jest wykorzystanie wszelkich dostępnych źródeł danych i zastosowanie technik rekomendacji najlepiej pasujących do danego problemu. Dane używane w SR dzielą się na 3 kategorie: użytkownicy, przedmioty oraz interakcje między nimi. Wszystkie systemy rekomendacyjne, aby być w stanie uzyskać preferencje użytkowników i wygenerować dla nich rekomendacje, zbierają informacje o ich interakcjach z przedmiotami. Niektóre z SR gromadzą również dane o użytkownikach, takie jak ich wiek, zainteresowania czy płeć, podczas gdy inne gromadzą dane przedmiotów, takie jak gatunek filmu czy cena produktu [5][10].

Przedmioty są obiektami, które są rekomendowane. Ilość dostępnych danych dotyczących przedmiotów zależy od złożoności ich pozyskania. Na przykład w systemie rekomendacyjnym wykorzystującym multimedia, obiekty takie jak obrazy, wymagają specjalnych algorytmów analizy obrazów, aby wyciągnąć ich cechy z dostępnych treści. Przedmioty będące obiektami tekstowymi takie jak wiadomości czy recenzje napisane przez użytkowników będą wymagać algorytmów przetwarzania tekstu [5][10].

Użytkownicy systemu rekomendacyjnego, jak wspomniano powyżej, mogą mieć bardzo zróżnicowane cele i cechy. Aby personalizować rekomendacje oraz interakcję człowiek-komputer, systemy rekomendacyjne wykorzystują wiele informacji o użytkownikach. Informacje te mogą być ustrukturyzowane na różne sposoby. Dane użytkownika stanowią tzw. model użytkownika. Model użytkownika profiluje użytkownika, czyli koduje jego preferencje i potrzeby. Stosowano różne podejścia do modelowania użytkowników, a system rekomendacyjny może być traktowany jako narzędzie generujące rekomendacje poprzez budowanie i wykorzystywanie modeli użytkowników. Wybór informacji, które mają być uwzględnione w modelowaniu, zależy od techniki rekomendacji, dostępności danych i wysiłków potrzebnych do ich zbierania i ekstrakcji wiedzy [5][10].

Na przykład, w filtrowaniu kolaboracyjnym użytkownik jest modelowany jako prosty zestaw jego ocen dla określonych przedmiotów. W kontekstowych systemach rekomendacyjnych model użytkownika uwzględnia informacje kontekstowe, aby rekomendować przedmioty użytkownikom w określonych sytuacjach kontekstowych ( np. tylko o określonej porze dnia). W systemach rekomendacyjnych opartych na treści użytkownicy mogą być modelowani na podstawie cech przedmiotów z którymi weszli w interakcję wcześniej [5][10].

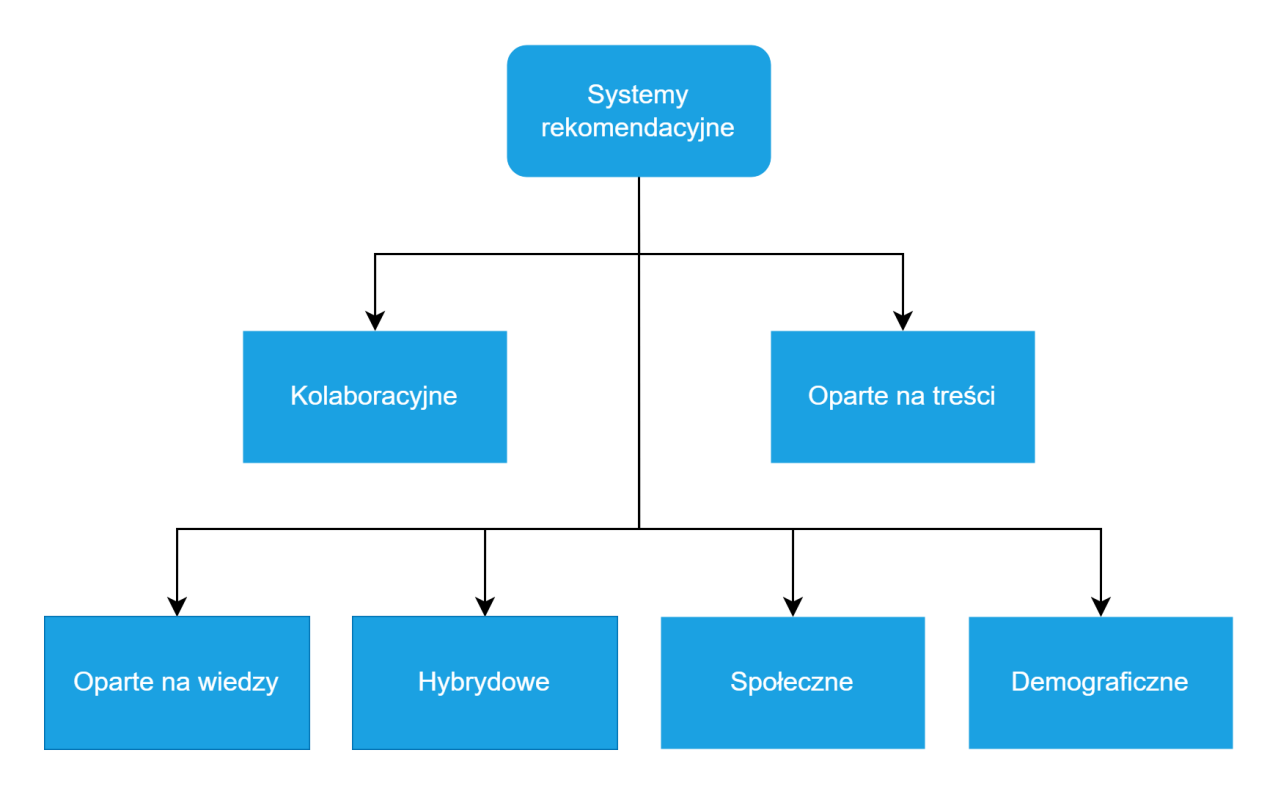
Informacje o interakcjach między użytkownikami i przedmiotami są gromadzone przez systemy rekomendacyjne pod postacią logów. Logi te składać się mogą z różnego rodzaju informacji. Przykładowo może to być odniesienie do produktu który użytkownik zakupił, ciąg zdarzeń który doprowadził to tego zakupu (np. przeglądanie, klikanie, dodanie do koszyka), ocena wystawiona dla przedmiotu lub dowolna inna informacja o interakcji, jej czasie czy miejsca zaistnienia. Prawdopodobnie najważniejszym rodzajem informacji o interakcjach między użytkownikami i przedmiotami są informacje zawrotne (ang. feedback) zebrane od użytkowników. Informacje te mogą być wyrażone jako jawne i bezpośrednie (ang. explicit feedback) lub jako ukryte i domniemane (ang. implicit feedback) [5][10].

Informacje zwrotne jawne są tą informacje świadomie wprowadzane do systemu przez użytkownika. Są one uważane za bardziej dokładne od informacji ukrytych i skutkujące generowaniem dobrze dopasowanych rekomendacji. Ich wadą jest jednak to, że wymagają większego zaangażowania użytkownika w proces rekomendacji co skutkuje tym, że zazwyczaj ich liczba jest niewielka. Najpopularniejszym ich rodzajem są oceny wystawiane przedmiotom. Oceny te mogą przyjmować różne formy [5][10]:

* Oceny numeryczne: na przykład oceny filmów od 1-5 w zbiorze MovieLens
* Oceny kategoryczne: użytkownik musi wybrać spośród listy opinii (przykładowo: „bardzo się zgadzam”, „zgadzam się”, ”oceniam neutralnie”, „nie zgadzam się” itd.)
* Oceny binarne: użytkownik musi wybrać spośród 2 opcji (przykładowo „lubię” lub „nie lubię”)
* Inne, na przykład ocena pod postacią recenzji produktu bądź komentarza na jego temat

Informacje zwrotne niejawne, lub domniemane odnoszą się do preferencji użytkownika do przedmiotu wydedukowanych przez SR z zaobserwowanych interakcji. Nie jest tutaj wymagany udział użytkownika w zbieraniu informacji zwrotnej, w przeciwieństwie do jawnych informacji zwrotnych. System automatycznie śledzi preferencje użytkowników, monitorując wykonywane działania, takie jak zobaczone przez nich przedmioty, gdzie użytkownicy klikali, jakie przedmioty zakupili lub jak długo przebywali na stronie internetowej. Aby uzyskać poprawne dane do rekomendacji, należy znaleźć odpowiednie akcje do monitorowania na podstawie dziedziny w której działać ma system rekomendacyjny i postawionych przed nim zadań. Inną zaletą domniemanej informacji zwrotnej jest to, że minimalizują one problemy związane z tzw. "zimnym startem", które występują, dopóki przedmiot nie zostanie oceniony wystarczająco, aby mógł zostać użyty do rekomendacji [5][10].

Burke w pracy „Hybrid Web Recommender Systems” zaproponował taksonomię systemów rekomendacyjnych, która stała się klasycznym sposobem rozróżniania poszczególnych typów rekomendacji [5][17]. Na poniższym schemacie przedstawiono wspomnianą taksonomię:



Rys. 5 Taksonomia systemów rekomendacyjnych. Źródło: Opracowanie własne na podstawie [17].

* 1. **Filtracja kolaboracyjna**

Filtracja kolaboracyjna (FK), czyli filtrowanie oparte na współpracy (ang. collaborative filtering) to jedna z metod generowania rekomendacji w systemach rekomendacyjnych, która do prognozowania nieznanych preferencji danego użytkownika wykorzystuje znane preferencje (oceny) grupy innych użytkowników. Jej główne założenie polega na tym, że osoby, które podobnie oceniały pewne przedmioty w przeszłości, są skłonne do podobnych ocen innych przedmiotów w przyszłości. Dla każdego użytkownika rekomendacje są tworzone indywidualnie, ale wykorzystywane informacje są zbierane od wielu użytkowników [18].

Filtracja kolaboracyjna jest najczęściej stosowaną techniką w systemach rekomendacyjnych. Podzielić ją można na filtrację opartą na pamięci (ang. memory-based collaborative filtering) i filtrację opartą na modelu (ang. model-based collaborative filtering).

* + 1. **Filtracja oparta na pamięci**

W filtracji opartej na pamięci dane opisujące preferencje użytkowników (ich oceny) są wykorzystywane do obliczania podobieństwa pomiędzy użytkownikami lub przedmiotami. Wartość podobieństwa może być następnie użyta do dokonania predykcji. Ten rodzaj filtracji cechuje wykorzystanie całego dostępnego zbioru dodanych, który musi zostać wczytany do pamięci w celu dokonania rekomendacji. Zaletami tego rozwiązania są łatwość implementacji ( wymagane jest jedynie obliczenie podobieństw między użytkownikami/przedmiotami), dobra skalowalność oraz możliwość uczenia systemu online - „w locie”. Wprowadzanie nowych danych jest więc proste. Istnieje też możliwość wytłumaczenia użytkownikowi dlaczego zarekomendowane zostały mu dane przedmioty. Jedną z wad tego rodzaju rozwiązań jest jednak niedokładność predykcji w momencie gdy istnieje niewiele bądź zero przedmiotów wspólnych dla użytkowników czyli w przypadku rzadkości danych.

Przykładem algorytmu filtracji opartej na pamięci jest algorytm k najbliższych sąsiadów (lub algorytm k-nn z ang. k nearest neighbours).

Często wykorzystywanymi miarami podobieństwa są podobieństwo kosinusowe mierzące podobieństwo między dwoma n wymiarowymi wektorami na podstawie kąta między nimi oraz współczynnik korelacji Pearsona określający stopień zależności liniowej między zmiennymi losowymi. Wyrażone zostały poniższymi wzorami:

,

gdzie:

- reprezentują dwóch użytkowników bądź dwa przedmioty,

– wektory ocen użytkowników lub przedmiotów,

- długość euklidesowa wektora zdefiniowana jako pierwiastek kwadratowy iloczynu skalarnego wektora ze samym sobą,

- zbiory wszystkich przedmiotów ocenionych przez odpowiednio: użytkownika ; użytkownika ; obydwu użytkowników,

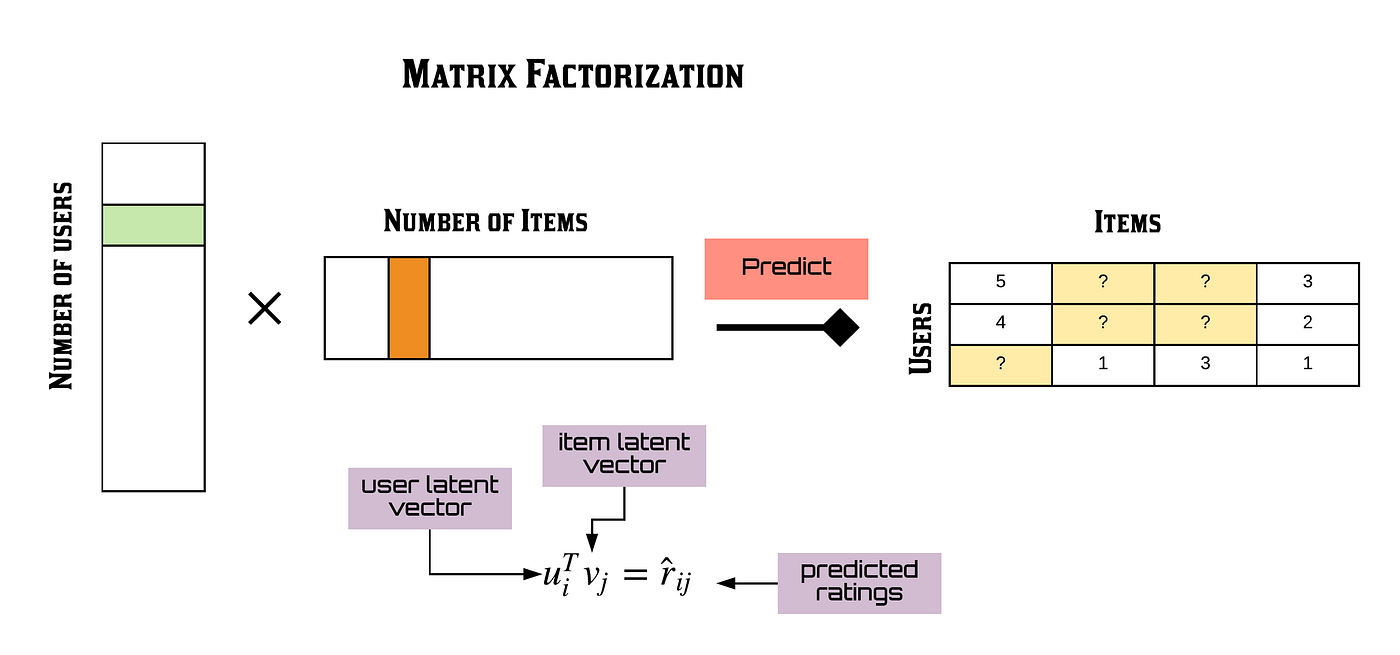
- ocena wystawiona przez użytkownika przedmiotowi ,

- średnia wszystkich ocen użytkownika .

* + 1. **Filtracja oparta na modelu**

W przeciwieństwie do filtracji opartej na pamięci, na podstawie dostępnych danych wytrenowany zostaje model opisujące cechy użytkowników i przedmiotów i dopiero ten model zostaje wykorzystany do utworzenia rekomendacji. Proces tworzenia modelu przeprowadzany jest korzystając z technik uczenia maszynowego takich jak klasyfikacja, analiza skupień, faktoryzacja macierzy czy sieci neuronowe. Algorytmy zaliczane do tego typu filtracji opracowane zostały w celu wyeliminowania wad, którymi obarczone były wcześniej wykorzystywane algorytmy filtracji opartej na pamięci takie jak wrażliwość na rzadkość danych czy słaba skalowalność. Stwierdzono, że cechuje się ono również wyższą precyzją. Wadami tego podejścia są wyższe wymagania zasobów takich jak czas i pamięć potrzebne do wytrenowania modelu [5][18].

W podejściu tym najczęściej stosowaną techniką jest redukcja wymiarów. Techniki takie jak dekompozycja SVD(ang. Singular Value Decomposition) czy analiza głównych składowych(ang. principal component analysis, PCA), które określane są jako modele cech ukrytych(ang. latent factor models) bazują na założeniu, że istnieje nieznana niskowymiarowa reprezentacja użytkowników i przedmiotów w której można precyzyjnie zamodelować ich podobieństwo. Przykładowo, w serwisie do oceniania filmów użytkownik zamiast posiadać 10000 pozycji które tworzyłyby jego model mógłby zostać opisany za pomocą wektora 50 cech, gdzie każda cecha była by liniową kombinacją filmów. Cecha taka reprezentować by mogła „filmy animowane produkcji Disneya z lat 90.” lub „filmy komediowe ze zwrotem akcji” itp. Rekomendacja bazująca na cechach ukrytych jest bardziej skuteczna niż bazowanie na samych podobieństwach między przedmiotami – użytkownik mógł nie oglądać filmu „Matrix” ale może znać filmy podobne do niego na podstawie cech ukrytych podobnych dla tych filmów [5].

Cechy ukryte, swoja nazwę zawdzięczają faktowi, że są one obecne w zbiorze danych ( np. macierzy interakcji między użytkownikami i przedmiotami) ale nie zostaną poznane dopóki nie zastosowane zostaną techniki redukcji wymiarowości. Jedną z najpopularniejszych z tych technik jest faktoryzacja macierzy. Opracowanych zostało wiele różnych algorytmów tego rodzaju, takich jak SVD++ czy NMF(nieujemna faktoryzacja macierzy ang. non-negative matrix factorization) [5]. 

Rys. 6 Mechanizm działania faktoryzacji macierzy.

Źródło: https://towardsdatascience.com/recsys-series-part-4-the-7-variants-of-matrix-factorization-for-collaborative-filtering-368754e4fab5

Pierwszym algorytmem wykorzystującym faktoryzację był zaprezentowany przez Simona Funka Funk MF lub Funk SVD, który to dokonywał rozkładu macierzy ocen użytkownicy-przedmioty na dwie macierze o niższych wymiarach gdzie w pierwszej macierzy każdy wiersz reprezentował użytkownika, a każda kolumna drugiej macierzy przedmiot jako wektor cech ukrytych.

Przewidywane oceny mogą być wyrażone jako:

gdzie:   
 jest obliczoną macierzą ocen użytkownicy-przedmioty,

zawiera cechy ukryte użytkowników,

zawiera cechy ukryte przedmiotów.

Dla pary użytkownika u i przedmiotu i, ocenę obliczyć można w następujący sposób:

Możliwe jest uzyskanie wyższej jakości predykcji poprzez zmianę liczby cech ukrytych modelu jednak wybranie zbyt dużej ich liczby może skutkować przeuczeniem i pogorszeniem predykcji. Funk MF minimalizuje poniższą funkcję kosztu:

gdzie jest normą Frobeniusa, a pozostałe normy macierzowe mogą być również normami Frobeniusa bądź innymi w zależności od danego problemu [20][5][8].

Powstało wiele innych algorytmów faktoryzacji macierzy, rozwijających Funk MF i osiągających lepsze rezultaty. Przykładami mogą być SVD++, Asymmetric SVD bądź Weighted MF. Najczęściej wykorzystywanymi algorytmami do minimalizacji funkcji kosztu są SGD oraz WALS ( Weighted Alternating Least Squares).

Obecnie, największą popularnością cieszą się rozwiązania wykorzystujące sieci neuronowe i uczenie głębokie i zaproponowanych zostało wiele modeli tego typu. Jednym z bardziej znanych rozwiązań z tej kategorii jest Neural Collaborative Filtering łączący w sobie elementy tradycyjnej faktoryzacji macierzy jak i uczenia głębokiego. Model ten zostanie omówiony dokładniej w części praktycznej pracy.

* 1. **Filtracja oparta na treści**

Filtracja oparta na zawartości, znana również jako filtracja oparta na treści (ang. Content-based filtering), jest jednym z najczęściej stosowanych podejść do personalizacji rekomendacji w systemach rekomendacyjnych. Modele filtracji kolaboracyjnej często borykają się z problemem rzadkości danych (ang. data sparsity problem), a także z problem zimnego startu (ang.cold-start). Dla użytkowników, którzy nieposiadaną wcześniejszych interakcji, algorytmy kolaboracyjne oparte na modelu nie są w stanie utworzyć rekomendacji. Filtracja oparta na treści pomaga rozwiązać wspomniane problemy.

Podejście to polega na analizie zawartości przedmiotów, a następnie rekomendowaniu użytkownikom produktów, które są podobne do tych, co do których wykazywali wcześniej zainteresowanie ignorując przy tym oceny innych użytkowników. Innymi słowy, filtracja oparta na zawartości polega na dopasowywaniu użytkowników do produktów, które odpowiadają ich preferencjom i gustom na podstawie cech i atrybutów produktów.

Na przykład, jeśli użytkownik oglądał wiele filmów o tematyce science-fiction, system rekomendacyjny może polecić mu kolejny film z tego samego gatunku, ponieważ jest to produkt, który jest podobny do tych, które użytkownik polubił wcześniej. Ważną cechą filtracji opartej na zawartości jest to, że nie wymaga ona dużych ilości danych na temat zachowań użytkowników, jak w przypadku filtracji kolaboracyjnej. W związku z tym, filtracja oparta na zawartości może być szczególnie przydatna w przypadku nowych użytkowników, którzy nie mieli jeszcze okazji ocenić wiele produktów lub dla nowych produktów, które mają małą liczbę ocen. Dodatkową zaletą tego podejścia jest to, że umożliwia ono dla użytkownika wgląd w swój mechanizm działania i wytłumaczenie dlaczego otrzymał dane rekomendacje. Jedną z istotniejszych jego wad jest to, że rekomendowane przedmioty są zazwyczaj podobnego typu i blisko ze sobą powiązane tematycznie jednak nie będące interesujące dla użytkownika (np. dla filmu „Matrix” zarekomendowany może zostać nikomu nieznany film o robotach z lat 40.) co określane jest jako efekt bąbla (ang. bubble effect)[5][18].

Jednym z najprostszych i popularnych algorytmów wykorzystywanych w filtracji opartej na treści jest TF-IDF (Term Frequency-Inverse Document Frequency). Jest to metoda statystyczna pozwalająca na zidentyfikowanie najważniejszych słów w dokumencie po wytrenowaniu na korpusie takich dokumentów. Tworzy ona rzadką reprezentację wektorową dokumentów, która może zostać następnie wykorzystana do obliczenia podobieństwa (np. kosinusowego) pomiędzy dokumentami. Wykorzystując TF-IDF, wektor opisujący użytkownika otrzymywany jest poprzez obliczenie średniej ważonej wektorów przedmiotów ocenionych przez użytkownika, a następnie na podstawie podobieństwa otrzymanego wektora i wszystkich wektorów przedmiotów nie poznanych przez użytkownika wybierane jest n najlepszych.

Metodę tą można zapisać następująco:

gdzie:  
 jest całkowitą liczbą wyrazów w dokumencie ,

jest częstością wyrazu w dokumencie,

jest całkowitą liczbą dokumentów w korpusie

* 1. **Inne podejścia**
     1. **Systemy oparte na społeczności**

Ta metoda generowania rekomendacji bazuje na preferencjach sieci znajomych danego użytkownika. Technikę tą można opisać przysłowiem „pokaż mi swoich przyjaciół, a powiem ci kim jesteś”. Wykazano, że ludzie są bardziej skłonni do zasugerowania się rekomendacjami pochodzącymi od swoich znajomych niż od nieznanych np. na stronie internetowej. Obserwacja ta, w połączeniu z ogromną popularnością serwisów społecznościowych skłoniła wielu badaczy do tworzenia systemów opartych na sieci społeczności. Tego typu modele gromadzą informację o relacjach społecznych użytkowników i preferencjach znajomych aktywnego użytkownika by utworzyć dla niego rekomendacje będące odzwierciedleniem upodobań całego kręgu znajomych [5].

* + 1. **Systemy demograficzne**

Ten typ systemów rekomenduje przedmioty na podstawie danych demograficznych w profilu użytkownika. Zakłada się w tym przypadku, że dla różnych nisz demograficznych powinno się wygenerować różne rekomendacje. Wiele serwisów implementuje proste techniki personalizacji wykorzystujące demografię. Przykładowo, użytkownicy mogą zostać przekierowani do osobnej witryny na podstawie języka lub kraju. Rekomendacje mogą też zostać utworzone bazując na wieku bądź płci użytkownika [5].

* + 1. **Systemy oparte na wiedzy**

Systemy oparte na wiedzy (ang. knowledge-based) rekomendują przedmioty wykorzystując jawną i specyficzną dla danej domeny wiedzę o tym jak cechy przedmiotów spełniają wymagania i preferencje użytkowników. Systemy te są często stosowane w przypadku gdy inne podejścia, takie jak filtracja kolaboracyjna lub filtracja oparta na treści nie mogłyby by zostać użyte. Zaletą systemów knowledge-based jest nieposiadanie przez nich problemu zimnego startu. Wadą jest natomiast potencjalne ograniczenie ilości pozyskanej wiedzy przez konieczność jawnego jej zdefiniowania. Systemy oparte na wiedzy są z powodzeniem stosowane w dziedzinach gdzie przedmioty są wybierane rzadko lub jednorazowo np. w serwisie sprzedającym nieruchomości lub w salonie motoryzacyjnym. Systemy korzystające z ocen użytkowników często zawodzą w tego typu przypadkach z powodu małej ilości ocen [5].

* + 1. **Systemy uwzględniające kontekst**

Preferencje użytkownika systemu rekomendacyjnego mogą się zmieniać w zależności od kontekstu (pora roku, pora dnia, nastrój użytkownika, lokalizacja, powód skorzystania z danego serwisu itp.). Przykładowo, to jakie filmy wybrałby użytkownik może zależeć od pory dnia, od tego czy użytkownik jest sam bądź z towarzystwem lub z jakiej okazji film ma zostać obejrzany. Innym przykładem takiego podejścia może być system rekomendacji kierunków wakacyjnych, rekomendujący inne atrakcje w zależności od pory roku ( zimą wyjazd na narty, latem wyjazd nad morze). Wyróżnia się zasadniczo trzy rodzaje tego rodzaju systemów. W podejściu Contextual Pre-filtering dane kontekstowe służą do wyboru przedmiotów z których właściwy algorytm rekomendacyjny wybierze najodpowiedniejsze dla użytkownika. Contextual Post-Filtering filtruje przedmioty wybrane przez właściwy algorytm rekomendacyjny i wybiera te, które pasują do kontekstu. Contextual Modeling jest podejściem bardziej zaawansowanym i zakłada jawne wykorzystanie danych kontekstowych w systemie rekomendacyjnym [5][21].

* 1. **Podejście hybrydowe**

System hybrydowy stanowi kombinacje dwóch lub więcej wymienionych powyżej algorytmów. Hybrydowy system łączący techniki A i B pozwala wykorzystać zalety techniki A, jednocześnie niwelując wady techniki B. Na przykład, w metodzie filtracji kolaboracyjnej występują problemy związane z dodawaniem nowych przedmiotów, nie posiadających ocen (problem zimnego startu). Ta sytuacja nie będzie jednak stanowić problemu dla filtracji opartej na zawartości w której wykorzystane zostaną jedynie cechy opisujące przedmioty. W opisany sposób możliwe jest stworzenie systemu cechującego się wysoką jakością predykcji w różnych scenariuszach. Ponadto, pojawienie się technik uczenia głębokiego znacznie ułatwiło budowanie hybrydowych systemów rekomendacyjnych i osiąganie jeszcze lepszych rezultatów.

Robin Burke w pracy „Hybrid Web Recommender Systems” wymienia 7 technik hybrydyzacji [17]:

* Ważenie: wyniki poszczególnych algorytmów rekomendacyjnych są łączone numerycznie.
* Przełączanie: system wybiera spośród składników rekomendacji i wykorzystuje najbardziej odpowiedni do zaistniałej sytuacji.
* Mieszanie: rekomendacje z różnych systemów rekomendacyjnych są prezentowane razem.
* Kombinacja cech: cechy pochodzące z różnych źródeł wiedzy są łączone razem i przekazywane do jednego algorytmu rekomendacji.
* Wzbogacanie cech: jedna technika rekomendacji jest używana do obliczenia cechy lub zestawu cech, które następnie są wykorzystywane w kolejnej technice.
* Technika kaskadowa: system rekomendacyjny działa na zasadzie szeregu etapów, w których systemy rekomendacyjne w każdym kolejnym etapie stopniowo polepszają wynik rekomendacji.
* Technika meta-poziomowa: jedna technika rekomendacji jest stosowana i tworzy pewien rodzaj modelu, który jest następnie wejściem dla kolejnej techniki.
  1. **Zastosowanie uczenia głębokiego w systemach rekomendacyjnych**

W ostatnich latach, uczenie głębokie zrewolucjonizowało sposoby tworzenia architektur systemów rekomendacyjnych i pozwoliło na polepszenie ich działania. Ostatnie postępy w rozwoju SR opartych na głębokim uczeniu zdobyły znaczną uwagę, pokonując przeszkody konwencjonalnych modeli i osiągając wysoką jakość rekomendacji. Uczenie głębokie jest w stanie skutecznie uchwycić nieliniowe i nietrywialnie relacje między użytkownikiem a przedmiotem, umożliwiając kodowanie bardziej złożonych abstrakcji pod postacią wielowymiarowych ukrytych cech. Ponadto, umożliwia łatwo wychwycić relacje istniejące pomiędzy różnymi dostępnymi źródłami danych takimi jak dane kontekstowe czy wizualne, co jest bardzo trudne do osiągnięcia w przypadku standardowych algorytmów uczenia maszynowego [22].

Wiele firm zaimplementowało uczenie głębokie w oferowanych usługach. Covington i in. zaprezentowali SR oparty na głębokich sieciach neuronowych dla rekomendacji treści wideo w serwisie YouTube. Cheng i in. zaproponowali system Wide&Deep rekomendujący aplikacje w usłudze Google Play. Serwisy takie jak Ebay, Twitter czy Spotify również wdrożyli systemy oparte na uczeniu głębokim. W testach online wymienione modele głębokie wykazały znaczącą poprawę jakości predykcji w porównaniu do tradycyjnych modeli [22].

Liczba publikacji na temat metod rekomendacyjnych opartych na głębokim uczeniu wzrosła w ostatnich latach wykładniczo, co jest dowodem na nieuniknioną wszechobecność głębokiego uczenia w badaniach nad SR. Międzynarodowa konferencja poświęcona systemom rekomendacyjnym, RecSys, zaczęła od roku 2016 organizować regularne warsztaty na temat głębokiego uczenia dla systemów rekomendacyjnych [22].

* 1. **Metody ewaluacji systemów rekomendacyjnych**

Ewaluacja jest niezbędna do ocenienia efektywności algorytmów rekomendacji. Przeprowadzać ją można online: badać działanie SR na podstawie reakcji użytkowników i metryk z nimi związanych, lub offline: badać działanie systemu na podstawie stabilnego zbioru danych historycznych wykorzystując do tego różnego rodzaju metryki statystyczne. Metryki te dzielą się na: metryki dokładności predykcyjnej, metryki dokładności klasyfikacyjnej, metryki dokładności rankingowej i metryki nie mierzące dokładności [5][23].

Metryki dokładności predykcyjnej lub predykcji ocen odnoszą się do tego, jak oszacowania ocen są bliskie w stosunku do rzeczywistych ocen użytkowników. Ten rodzaj metryki jest powszechnie stosowany do ewaluacji ocen niebinarnych. Najważniejszymi z tego rodzaju metryk są średni błąd absolutny (MAE – ang. „mean absolute error”), błąd średniokwadratowy (MSE – ang. „mean squared error”) i pierwiastek z błędu średniokwadratowego (RMSE - root mean squared error) [5][23].

Metryki te można zapisać następująco:

gdzie:

– prawdziwa ocena przedmiotu wystawiona przez użytkownika ,

– przewidywana ocena przedmiotu wystawiona przez użytkownika ,

– zbiór par dla których istnieje ocena

Metryki dokładności klasyfikacyjnej mierzą liczbę poprawnych i niepoprawnych klasyfikacji przedmiotów dokonanych przez system rekomendacyjny. Metryki te ignorują wartości ocen i pozycje rankingowe rekomendowanych przedmiotów i mierzą poprawność ich zaklasyfikowania (na przykład, czy przedmiot zainteresował użytkownika czy też nie) [23]. Przedstawiona poniżej macierz błędów jest wykorzystywana do oceny błędu klasyfikacji binarnej:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Dopasowany do użytkownika | Niedopasowany | Suma |
| Zarekomendowano | *TP*  (prawdziwie pozytywny) | *FP*  (fałszywie pozytywny) | *TP + FP* |
| Nie zarekomendowano | *FN*  (fałszywie negatywny) | *TN*  (fałszywie negatywny) | *FN + TN* |
| Suma | *TP + FN* | *FP + TN* | *n* |

Tabela 1 Macierz błędów rekomendacji

Najpopularniejszymi metrykami bazującymi na powyższej macierzy są [5]:

* dokładność (ang. accuracy) określająca stosunek pozytywnych klasyfikacji do wszystkich:
* precyzja (ang. precision) określa ile z przedmiotów zaklasyfikowanych pozytywnie jest prawdziwie pozytywna:
* czułość (ang. recall) określa ile przedmiotów prawdziwie pozytywnych zostało wykrytych:
* swoistość (ang. specificity) określa ile przedmiotów prawdziwie negatywnych zostało wykrytych:

Ponadto, istnieje wiele innych, bardziej rozbudowanych metryk dokładności klasyfikacji. Z ważniejszych wymienić można MAP (Mean Average Precision), MAR (Mean Average Recall), krzywą ROC (Receiver Operating Characteristic) czy pole pod krzywą ROC – AUC [5].

Metryki dokładności rankingowej pozwalają ocenić zdolność algorytmu do zarekomendowania listy przedmiotów w kolejności odpowiadającej rzeczywistym preferencjom użytkownika. Opierają się one wyłącznie na kolejności rekomendowanej listy przedmiotów i są niezależne od wartości ocen obliczonych przez algorytm [5][23].

Jedną z popularnych metryk tego rodzaju jest MRR – średni odwrotny ranking (ang. mean reciprocal rank) będąca średnią rangą pierwszego dopasowanego przedmiotu we wszystkich zbiorach rekomendacji. Metryka ta mierzy, jak szybko użytkownik znajduje element, który jest do niego najbardziej dopasowany. Im wcześniej ten element zostanie znaleziony w rankingu, tym wyższa wartość MRR. MRR opisać można następującym wzorem [5][23]:

gdzie:

- ranga (pozycja) pierwszego przedmiotu w zbiorze rekomendacji który użytkownik faktycznie wybrał,

– użytkownik należący do zbioru użytkowników

W metryce MRR dopasowanie przedmiotu do użytkownika określane jest binarnie (dopasowanie lub brak dopasowania). Dodatkowo uwzględnione zostaje jedynie pierwsze wystąpienie dopasowanego przedmiotu.

Metryka nDCG (ang. Normalized Discounted Cumulative Gain), której nazwę przetłumaczyć można jako „Znormalizowany zdyskontowany zysk kumulacyjny” bierze pod uwagę nie tylko to, czy dany element został umieszczony wysoko w rankingu, ale również jego rzeczywistą wartość dla użytkownika. Metryka ta bada, jak dobrze elementy rankingu pasują do zainteresowań użytkownika. Zakładając, że każdy użytkownik u osiąga zysk g dzięki otrzymaniu rekomendacji przedmiotu i, średni DCG dla zbioru przedmiotów J można zapisać następującym wzorem [5][23]:

gdzie:

- przedmiot na pozycji w zbiorze ,

– podstawa logarytmu, zazwyczaj stosowaną wartością jest 2

Metryka nDCG jest znormalizowaną wersją DCG i dana jest wzorem:

Gdzie jest wynikiem DCG policzonym dla idealnego zbioru rekomendacji tzn. posortowanego od przedmiotu z najwyższą wartością dopasowania (np. wysokością oceny filmu) do przedmiotu z najniższą [5][23].

Istnieją również inne miary pozwalające ocenić działanie systemu rekomendacyjnego. Zaliczyć do nich można m.in. [5]:

* Zasięg (ang. coverage), reprezentujący procent przedmiotów, które model był w stanie zarekomendować z całego ich zbioru.
* Personalizację, pokazującą czy model rekomenduje wiele podobnych przedmiotów różnym użytkownikom. Wyznacza się ją, obliczając niepodobieństwa (1 – podobieństwo kosinusowe) pomiędzy listami przedmiotów zarekomendowanych dla każdego z użytkowników, a następnie licząc średnią z otrzymanych wartości.
* Podobieństwo wewnątrz-listowe (ang. intra-list similarity), będące średnim podobieństwem kosinusowym pomiędzy wszystkimi przedmiotami w liście rekomendacji. Do obliczenia podobieństwa między przedmiotami używa się ich cech.
* Miary biznesowe oraz mierzące odbiór systemu przez użytkowników, np.: testy A/B, współczynnik klikalności, współczynnik konwersji, prywatność, zaufanie do systemu, jak zaskakujące są rekomendacje itp.
  1. **Powtarzalność wyników w badaniach nad systemami rekomendacyjnymi**

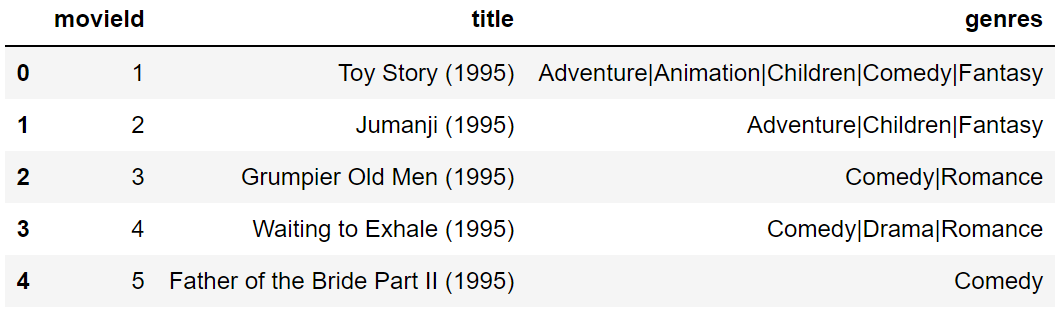
Systemy rekomendacyjne są trudne do oceny offline, niektórzy badacze twierdzą, że fakt ten spowodował nawet kryzys powtarzalności wyników w publikacjach dotyczących SR. Temat powtarzalności wyników jest powtarzającym się problemem w niektórych pracach z dziedziny uczenia maszynowego. Praca z 2019 roku przebadała pewną liczbę wybranych publikacji wykorzystujących głębokie uczenie lub sieci neuronowe do problemu rekomendacji, opublikowanych na czołowych konferencjach (SIGIR, KDD, WWW, RecSys, IJCAI). Wyniki pokazały, że średnio mniej niż 40% artykułów mogło być odtworzone przez autorów badania, a w niektórych konferencjach było to tylko 14%. W nowszej pracy sprawdzającej ten sam zestaw modeli uzyskano bardzo różne jakościowo wyniki, a metody wykorzystujące sieci neuronowe i deep learning osiągnęły jedne z najlepszych rezultatów [24][25].

Problem ten widać również na przykładzie popularnej techniki Neural Collaborative Filtering (NeuCF), która została zaimplementowana w rozdziale trzecim niniejszej pracy dyplomowej. W publikacji [26], autorzy stwierdzają, że wykorzystanie jedynie faktoryzacji macierzy jako algorytmu rekomendacji nie daje zadowalających wyników i proponują zastosowanie sieci neuronowej łączącej w sobie faktoryzację macierzy i perceptron wielowarstwowy tworzący sieć głęboką. Dzięki temu, model jest w stanie odkryć zależności nieliniowe w zbiorze danych i uzyskać lepsze wyniki. W pracy „Neural Collaborative Filtering vs. Matrix Factorization Revisited” [27] autorzy podważają te wyniki i stwierdzają, że metody oparte na faktoryzacji macierzy osiągają znacznie lepsze wyniki niż podejście korzystające z sieci neuronowych oraz że rozwiązania uczenia głębokiego są zbyt kosztowne obliczeniowo. W 2021 w pracy „Rethinking Neural vs. Matrix-Factorization Collaborative Filtering: the Theoretical Perspectives” [28] autorzy nie zgadzają się z wnioskami z pracy [27] i stwierdzają, że model NeuCF proponowany w [26] dorównuje standardowej metodzie wykorzystującej iloczyn skalarny. Dodatkowo zauważają, że te dwa podejścia cechuje pewien kompromis zalet i wad w zależności od specyficznego zastosowania.

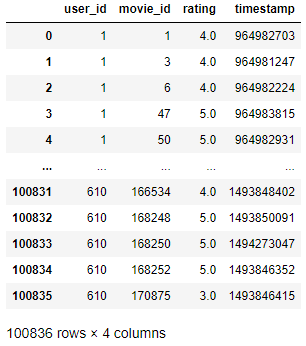
1. **Opracowanie systemów rekomendacyjnych**
   1. **Zbiór MovieLens, badania eksploracyjne i edycja zbioru danych**

Do wytrenowania i porównania zaimplementowanych algorytmów wykorzystany został zbiór danych MovieLens udostępniany przez GroupLens. Zbiór ten stał się standardem w dziedzinie systemów rekomendacyjnych i wykorzystany został w dużej ilości publikacji w tej dziedzinie [5].

W skład użytego zbioru ml-latest-small wchodzi 100.836 ocen wystawionych przez 610 użytkowników dla 9742 filmów w latach 1996-2018. Każdy z użytkowników ocenił minimum 20 filmów. Użytkownicy zostali wybrani losowo. Dodatkowo, w zbiorze danych zawartych zostało 3683 tagów opisujących filmy. Oceny zapisane są w pliku ratings.csv w formacie (numer użytkownika, numer filmu, ocena). Oceny są wyrażone w skali 0,5-5,0, wzrastając co 0,5. Poniżej przedstawiono przykładowe dane ze zbioru ml-latest-small:

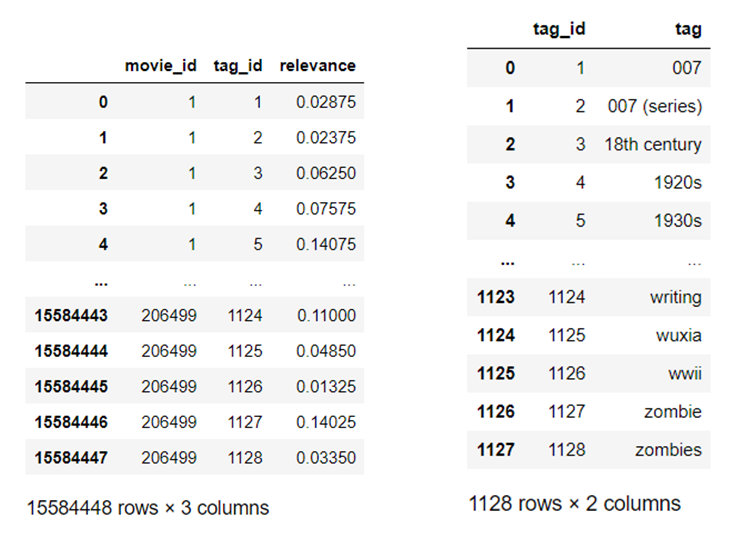


Rys. Struktura pliku movies.csv.



Rys. Struktura pliku ratings.csv.

Dodatkowo w pracy wykorzystano zbiór danych MovieLens Tag Genome zawierający listę tagów opisujących filmy z miarą relewancji, czyli tego jak trafny jest tag do danego filmu. Tagi są to pojedyncze słowa lub krótkie wyrażenia opisujące filmy (np. „atmospheric”, „thought-provoking”, „realistic”). Poniżej przedstawiono przykładowe dane ze zbioru Tag Genome:

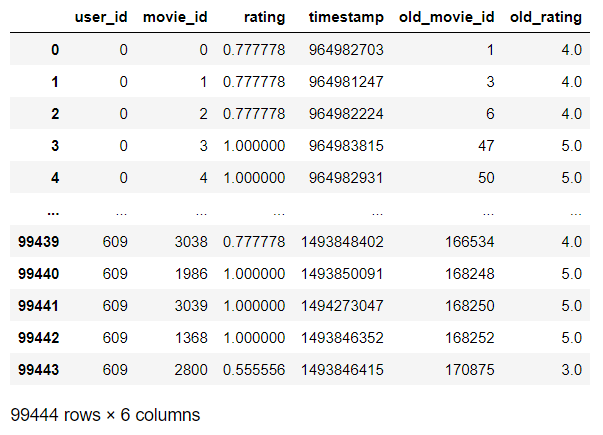


Rys. Zawartość genome-scores oraz genome-tags.

.

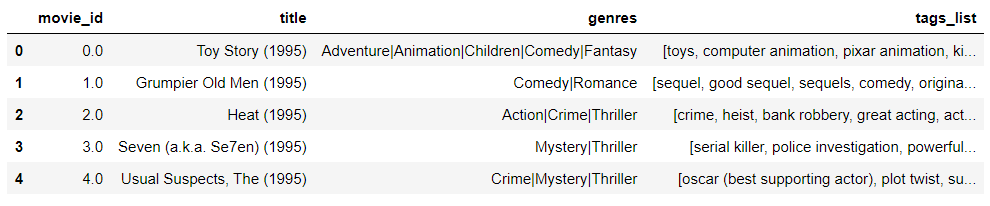
W celu ujednolicenia danych zawartych w powyższych plikach dokonano ich modyfikacji. W pliku ratings.csv nie są obecne wszystkie filmy obecne w pliku genome-scores.csv oraz w pliku movies.csv. Filmy różniące te zbiory danych zostały usunięte, w efekcie czego uzyskany został spójny zbiór danych składający się z 610 użytkowników, 8651 filmów (usunięto 1073 filmy) oraz 99444 ocen.

Kolejnym krokiem w edycji danych było przeskalowanie wartości ocen z pierwotnego zakresu od (0.5, 5.0) do (0.0, 1.0) czyniąc je łatwiejszymi do nauki przez zaimplementowane modele uczenia maszynowego. Następnie zastosowano zmianę numeracji filmów i użytkowników na spójną i zaczynającą się od 0. W efekcie tych działań, zbiór ocen wyglądał następująco:



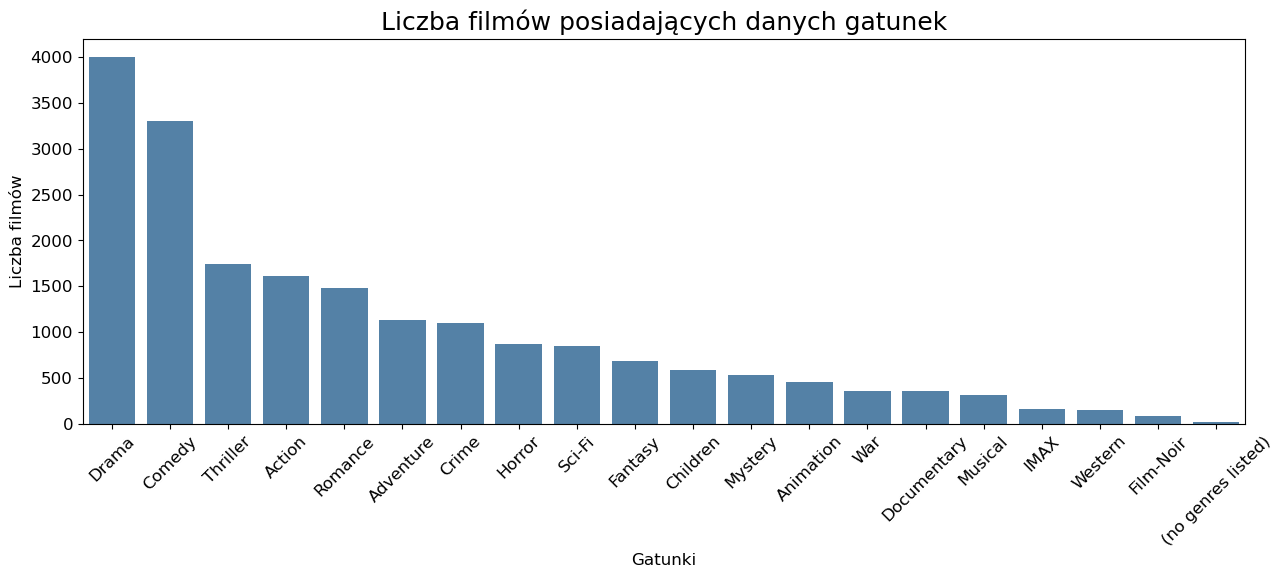
Rys. Zbiór ocen po przeskalowaniu wartości ocen oraz zmianie indeksacji użytkowników i filmów.

W kolejnym kroku zmodyfikowany został zbiór filmów z pliku movies.csv. Dla każdego z filmów dodanych zostało 50 tagów najlepiej je opisujących (według wartości kolumny „relevance”). Dane te, wykorzystane zostaną w implementacji modelu filtracji opartej na zawartości – Content Based. Po opisanej edycji, zbiór filmów prezentował się następująco:



Rys. Zbiór filmów z pliku movies.csv po edytowaniu.

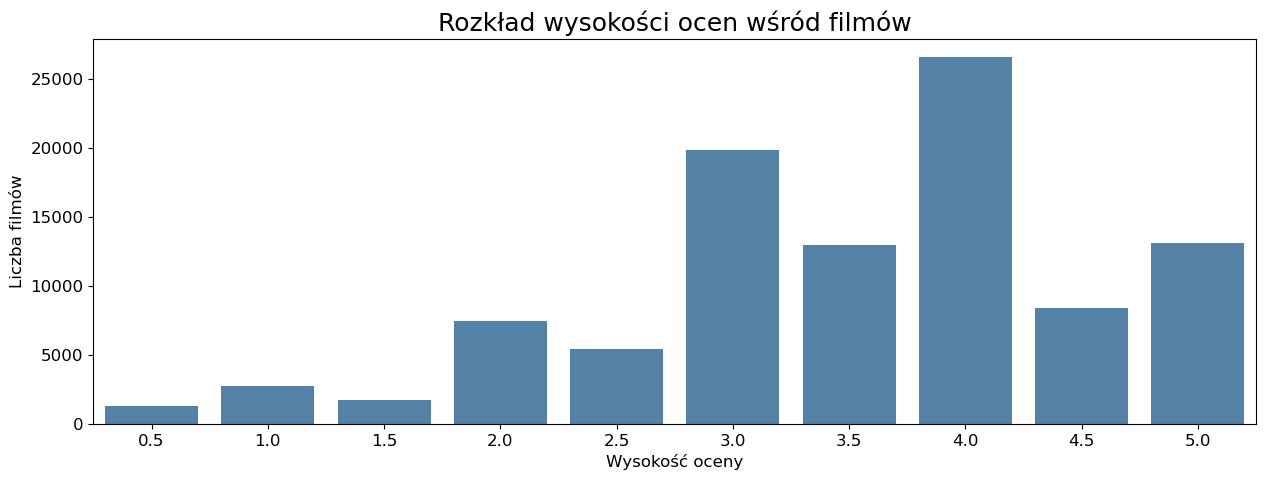
Każdy z filmów z powyższego zbioru oprócz kolumny zawierającej listę tagów posiada również kolumnę “genres”, w której znajduje się lista gatunków filmowych do których zaliczyć można film. W zbiorze filmów występuje 20 unikalnych gatunków filmowych: Romance', 'Adventure', 'Action', 'War', 'Documentary', 'Comedy', 'Children', 'Western', 'Film-Noir', 'Fantasy', 'Thriller', 'Musical', 'Sci-Fi', 'Drama', 'Mystery', 'Horror', 'Animation', 'IMAX', 'Crime', '(no genres listed)'. Na poniższym wykresie przedstawione zostały ilości filmów klasyfikujących się do danych gatunków filmowych:



Wykres Liczba filmów klasyfikujących się do danych gatunków filmowych.

Z powyższego wykresu odczytać można, że zdecydowana większość filmów zaliczona jest do kategorii „drama” oraz „comedy”, widać również, że istnieją filmy bez przypisanych im gatunków („no genres listed”).

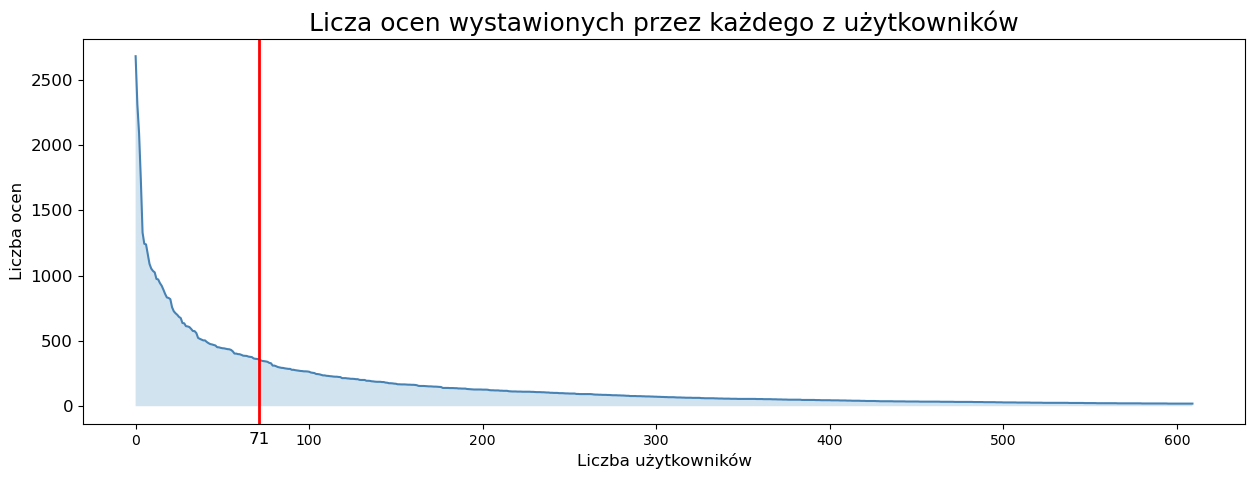
Następnym krokiem w badaniu eksploracyjnym zbioru danych MovieLens było przyjrzenie się danym zawartym w pliku ratings.csv. Poniższy wykres przedstawia rozkład wysokości ocen wśród filmów w zbiorze:



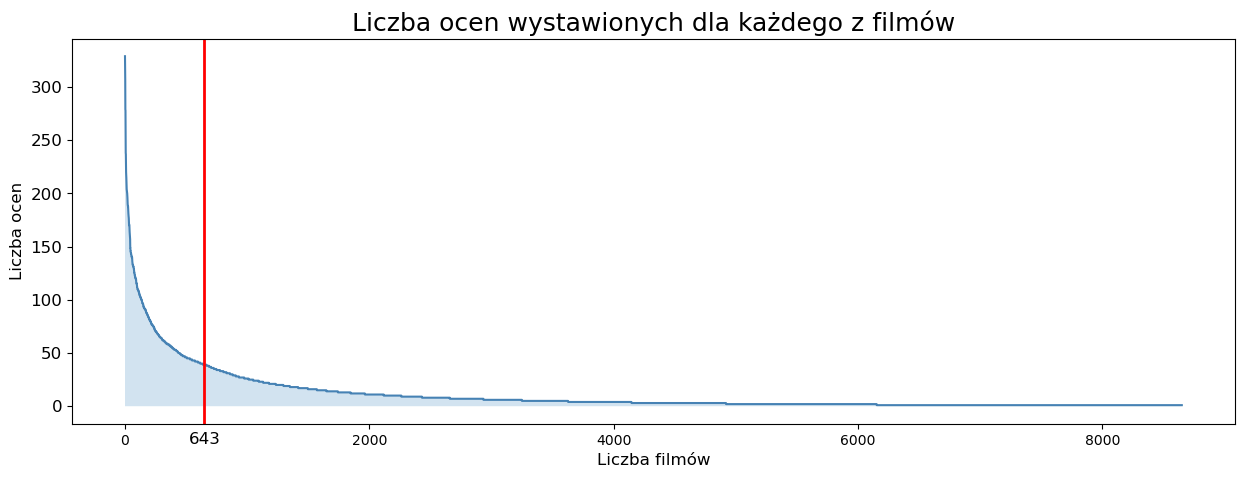
Wykres Rozkład wysokości ocen wśród filmów.

Z wykresu odczytać można, że w zbiorze ocen dominują oceny 3.0, 3.5 i 4.0, a najmniej jest ocen w wysokości 0.5. Rozkład wysokości ocen w niewielkim stopniu przypomina rozkład normalny, jest jednak w pewnym stopniu przesunięty w stronę wyższych ocen.

Poniższe dwa wykresy przedstawiają odpowiednio rozkład ocen wśród użytkowników oraz filmów:



Wykres Rozkład ocen wśród użytkowników. Czerwona linia wyznacza połowę liczby wystawionych ocen.

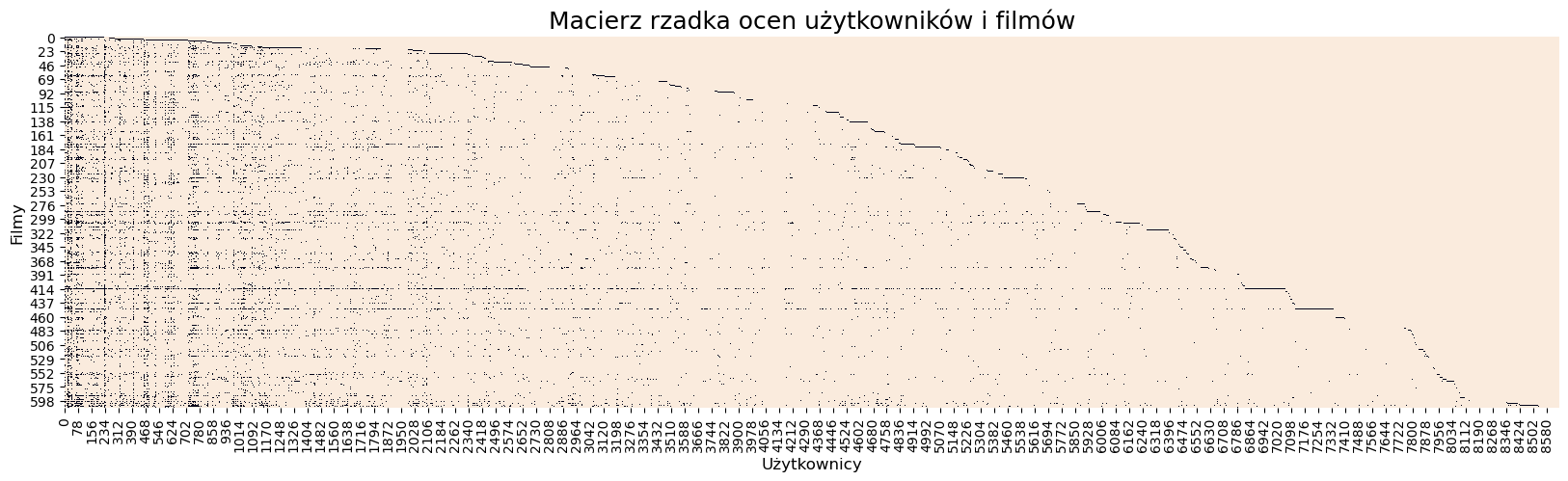


Wykres Rozkład ocen wśród filmów. Czerwona linia wyznacza połowę liczby wystawionych ocen.

Z pierwszego wykresu wynika, że 71 użytkowników wystawiło połowę obecnych w zbiorze ocen podczas gdy druga połowa wystawiona została przez ok. 540 użytkowników. Sygnalizuje to, że dane o ocenach są bardzo rzadkie i mogą pojawić się problemy podczas generowania rekomendacji dla większości użytkowników posiadających niską ilość ocen na podstawie których można by stworzyć profil użytkownika. Drugi wykres przedstawia rozkład ilości ocen wśród filmów i sytuacja jest tutaj podobna jak w przypadku użytkowników. Spośród 100836 ocen wystawionych dla 9742 filmów, 643 filmy, tj. ok. 6,6% wszystkich filmów posiada 50% wszystkich ocen. Sytuację taką określa się jako występowanie długiego ogona [5].

Problemem długiego ogona w systemach rekomendacyjnych określa się nierównomierny rozkład popularności przedmiotów, gdzie niewielka liczba przedmiotów cieszy się dużą popularnością (zwana „głową” rozkładu), podczas gdy większość przedmiotów ma niewielką ilość ocen lub interakcji (stanowią one „długi ogon”). Oznacza to, że większość przedmiotów często nie jest rekomendowana użytkownikom, co prowadzi do niedostatecznego uwzględnienia ich zainteresowań. Ponieważ w danych treningowych występuje wiele obserwacji popularnych przedmiotów, dla systemu rekomendacyjnego nie jest trudno nauczyć się precyzyjnego przewidywania tych przedmiotów. W zbiorze danych filmowych najpopularniejsze filmy to tzw. „blockbuster'y” i klasyki. Filmy te są już dobrze znane większości użytkowników, a rekomendacje dotyczące nich mogą nie oferować spersonalizowanego doświadczenia ani pomagać użytkownikom odkrywać nowe, wartościowe dla nich filmy.

Jako że w analizowanym zbiorze obecnych jest 610 użytkowników oraz 9742 filmów istnieje 5.942.620 możliwych par użytkownik – film. Zbiór ocen zawiera natomiast jedynie ok. 100.000 takich par. Jest więc to zbiór bardzo rzadki. Poniższy wykres przedstawia przybliżony wygląd macierzy użytkowników i filmów:



Rys. Macierz rzadka ocen użytkowników i filmów.

* 1. **Referencyjne systemy rekomendacyjne**

W celu oceny działania proponowanych modeli systemów rekomendacyjnych niezbędne jest zaimplementowanie modeli bazowych (ang. „baseline models”) lub referencyjnych, ponieważ umożliwia to porównanie wyników i wydajności nowego modelu z prostymi, ale dobrze działającymi metodami. Modele bazowe to proste modele, które mogą służyć jako punkt odniesienia do oceny nowego modelu. Porównanie wyników między nowym modelem systemu rekomendacyjnego a modelem bazowymi pozwala ocenić, czy nowy model osiąga lepsze rezultaty w zakresie rekomendacji. W tym celu, w ramach pracy zaimplementowane zostały 3 systemy bazowe:

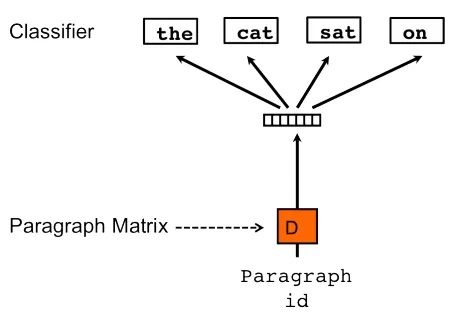
* Najbardziej oczywistym i najprostszym sposobem na wygenerowanie listy rekomendacji dla użytkownika jest wygenerowanie tej listy w sposób losowy. Wiąże się to z faktem całkowitego niedopasowania utworzonych rekomendacji do użytkownika. Porównywanie nowego modelu do modelu losowego jest jednak bardzo pożyteczne i pozwala ocenić, czy nowy model jest w stanie zaoferować jakikolwiek wzrost personalizacji i ogólnej jakości rekomendacji. Jeśli nowy model nie wykazuje poprawy w stosunku do losowych rekomendacji lub nawet sprawuje się gorzej oznacza to, że należy go skorygować, bądź wybrać inny model.
* Prostym rozwiązaniem na wygenerowanie rekomendacji jest model bazujący na popularności filmów. Dla każdego użytkownika generuje on identyczną listę najbardziej popularnych w zbiorze filmów pod względem ilości ocen. Rekomendacja taka z założenia nie będzie spersonalizowana ale wciąż może być bardzo dokładna z powodu przedstawionego w poprzednim podrozdziale rozkładu długiego ogona ocen.
* Ostatnim bazowym modelem wykorzystanym w pracy jest model SVD (Singular Value Decomposition) przedstawiony w rozdziale 2. SVD zaimplementowane zostało z wykorzystaniem biblioteki Surprise, Jest to podstawowy model kolaboracyjny przekształcający interakcje między użytkownikami i przedmiotami do przestrzeni ukrytej i następnie odtwarzający brakujące oceny filmów w rzadkiej macierzy.
  1. **System rekomendacyjny oparty na treści**

Model oparty na treści do działania wymaga posiadania reprezentacji przedmiotów (filmów) na podstawie ich zawartości. W przypadku wykorzystanego w pracy zbioru danych MovieLens, dostępne są dane tagów przypisanych do każdego z filmów. Zbiór tagów Tag Genome jest bardzo obszerny i pozwala opisać każdy z filmów. Do znalezienia wektorowych reprezentacji filmów wykorzystany został algorytm Doc2Vec.

Algorytm Doc2Vec, znany również jako Paragraph Vector, jest metodą reprezentacji tekstów, która rozszerza popularny algorytm Word2Vec na poziom całych dokumentów. Doc2Vec umożliwia wygenerowanie gęstych wektorów reprezentujących dokumenty lub sekwencje tekstowe, które zachowują semantyczną i kontekstową informację. Podstawowym założeniem algorytmu Doc2Vec jest, że każdy dokument w korpusie jest reprezentowany przez unikalny wektor. Algorytm uczy się mapować dokumenty do przestrzeni n-wymiarowej, gdzie odległość między wektorami dokumentów odzwierciedla podobieństwo semantyczne lub tematyczne między nimi. Doc2Vec wykorzystuje architekturę sieci neuronowej do generowania wektorów dokumentów. Istnieją dwie główne wersje algorytmu Doc2Vec: DBOW (Distributed Bag of Words) i DM (Distributed Memory) [29].

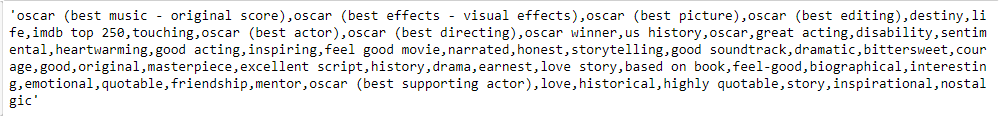
W przypadku DBOW, algorytm ignoruje kolejność słów wewnątrz dokumentu i stara się przewidzieć słowa na podstawie kontekstu całego dokumentu. Modele DBOW są bardziej zrównoleglone i mogą być szybsze w uczeniu, ale nie przechwytują informacji o kolejności słów [29]. Wariant DBOW wykorzystany został w pracy.

W przypadku DM, algorytm uwzględnia zarówno kontekst dokumentu, jak i poszczególne słowa wewnątrz dokumentu. Algorytm stara się przewidzieć następne słowo wewnątrz dokumentu na podstawie kontekstu poprzednich słów i samego dokumentu. Modele DM są bardziej skomplikowane obliczeniowo, ale przechwytują informacje o kolejności słów, co może prowadzić do lepszej jakości reprezentacji dokumentów [29].

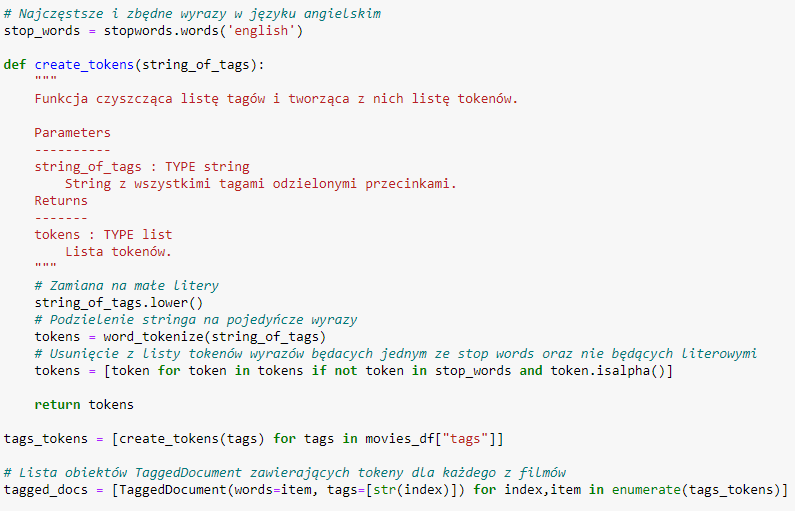
****

Rys. Wizualizacja techniki PV-DBOW z pracy Doc2Vec. Dla każdego ID paragrafu ( dokumentu ) model jest uczony przewidywania wystąpienia słowa (tagu filmu). Źródło: „Distributed Representations of Sentences and Documents” [29]

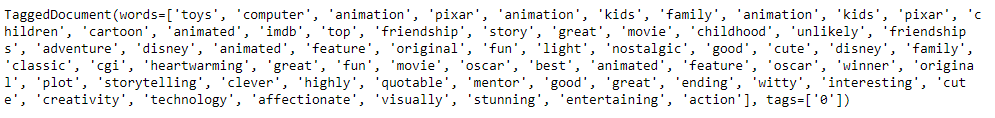
Algorytm Doc2Vec zaimplementowany został z wykorzystaniem biblioteki Gensim. Tagi filmów występują w zbiorze w różnych formatach, zawierają spacje, znaki interpunkcyjne, itp. Może to zakłócać uczenie się algorytmu Doc2Vec więc konieczne było oczyszczenie listy tagów:



Rys. Przykładowa lista tagów przed oczyszczeniem.

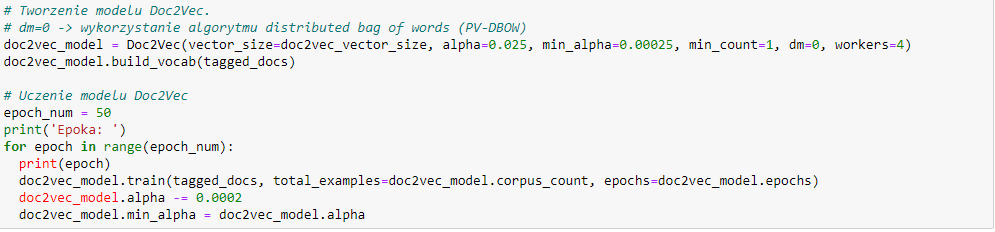


Rys. Oczyszczanie tagów ze znaków innych niż litery oraz ze słów przystankowych jak "the", "a", "an", a następnie konwersja listy tagów na obiekty TaggedDocument.



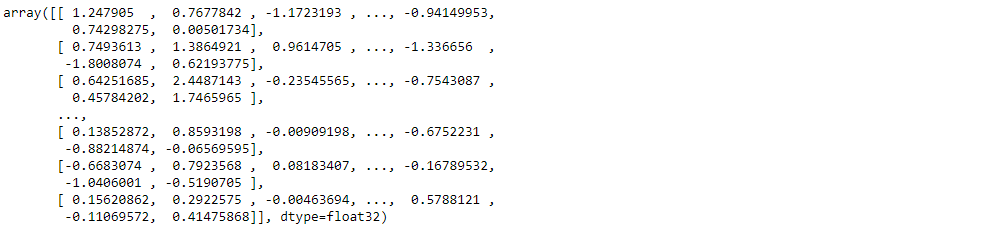
Rys. Przykładowy obiekt TaggedDocument z przetworzonymi tagami.

Po przygotowaniu tagów, przystąpiono do uczenie modelu Doc2Vec:



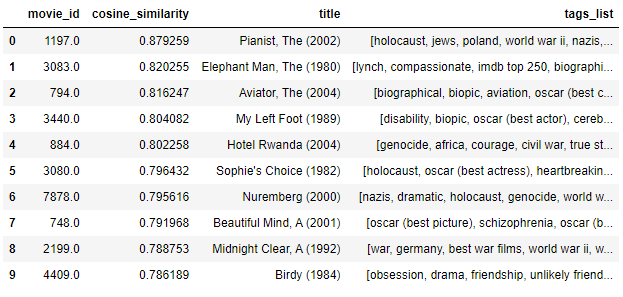
Rys. Uczenie modelu Doc2Vec przez 50 epok, rozmiar wektorów które mają reprezentować filmy ustalono na 20.

Po zakończeniu uczenia modelu, uzyskano wektory opisujące zawartość filmów:



Rys. Wektorowe reprezentacje filmów.

Posiadając powyższe wektory reprezentujące każdy z filmów możliwe jest obliczenie podobieństwa cosinusowego między wektorami i znalezienie w ten sposób filmów najbardziej podobnych do zadanego.



Rys. Lista filmów najbardziej podobnych do filmu „Schindler's List (1993)”.

Jak widać, filmy z uzyskanej listy rzeczywiście są podobne w tematyce do podanego filmu. W celu zwizualizowania otoczenia podanego filmu wykorzystano bibliotekę WordCloud:



Rys. Chmura słów opisująca otoczenie filmu „Schindler's List (1993)”.

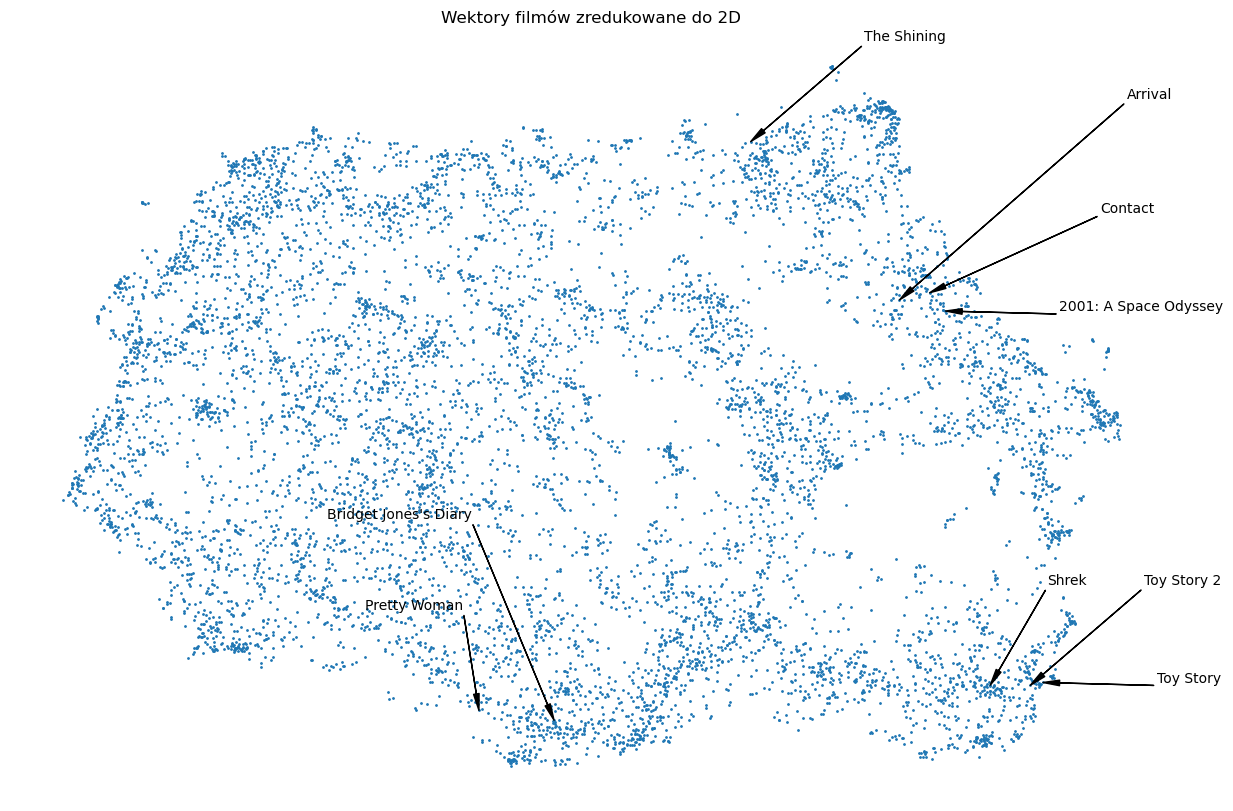


Rys. Lista filmów najbardziej podobnych do filmu Lord of the Rings: The Fellowship of the Ring, The (2001).



Rys. Chmura słów opisująca otoczenie filmu Lord of the Rings: The Fellowship of the Ring, The (2001).

Korzystając z biblioteki UMAP zredukowano wymiarowość wektorów filmów w celu wizualizacji:



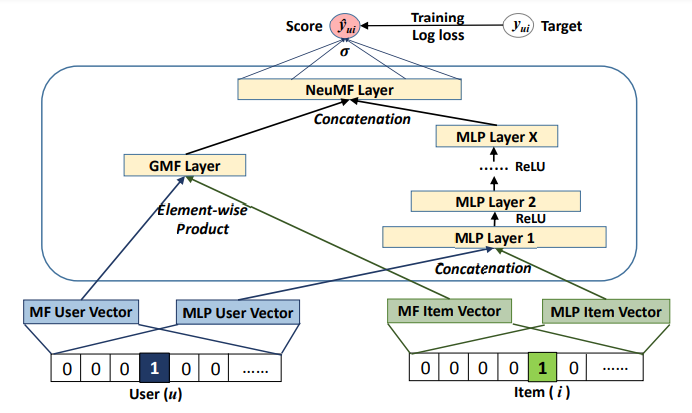
Rys. 20 wymiarowa przestrzeń zredukowana do 2 wymiarów.

Analizując powyższy wykres, wyciągnąć można wniosek, że model Doc2Vec znalazł bardzo dobre reprezentacje filmów w przestrzeni 20 wymiarów. Na wykresie widać, że filmy od których oczekiwałoby się żeby znalazły się blisko siebie, rzeczywiście są blisko. Filmy animowane znalazły się blisko siebie, przy czym „Toy Story 2” znajduje się bliżej „Toy Story” niż filmu „Shrek”.

W celu generowania rekomendacji dla użytkowników z wykorzystaniem wektorów uzyskanych przez algorytm Doc2Vec konieczne jest utworzenie profili użytkowników. Wykonane to zostało poprzez wybranie filmów obejrzanych przez użytkownika i ocenionych przez niego powyżej arbitralnie wybranej wysokości oceny ( zastosowano ocenę 3.0) i uśrednienie wektorów tych filmów. Tak otrzymany wektor stanowi reprezentację profilu użytkownika. Obliczając podobieństwo cosinusowe wszystkich wektorów filmów do wektora profilu użytkownika, a następnie wybierając K filmów z najwyższym podobieństwem otrzymać można listę rekomendowanych filmów.

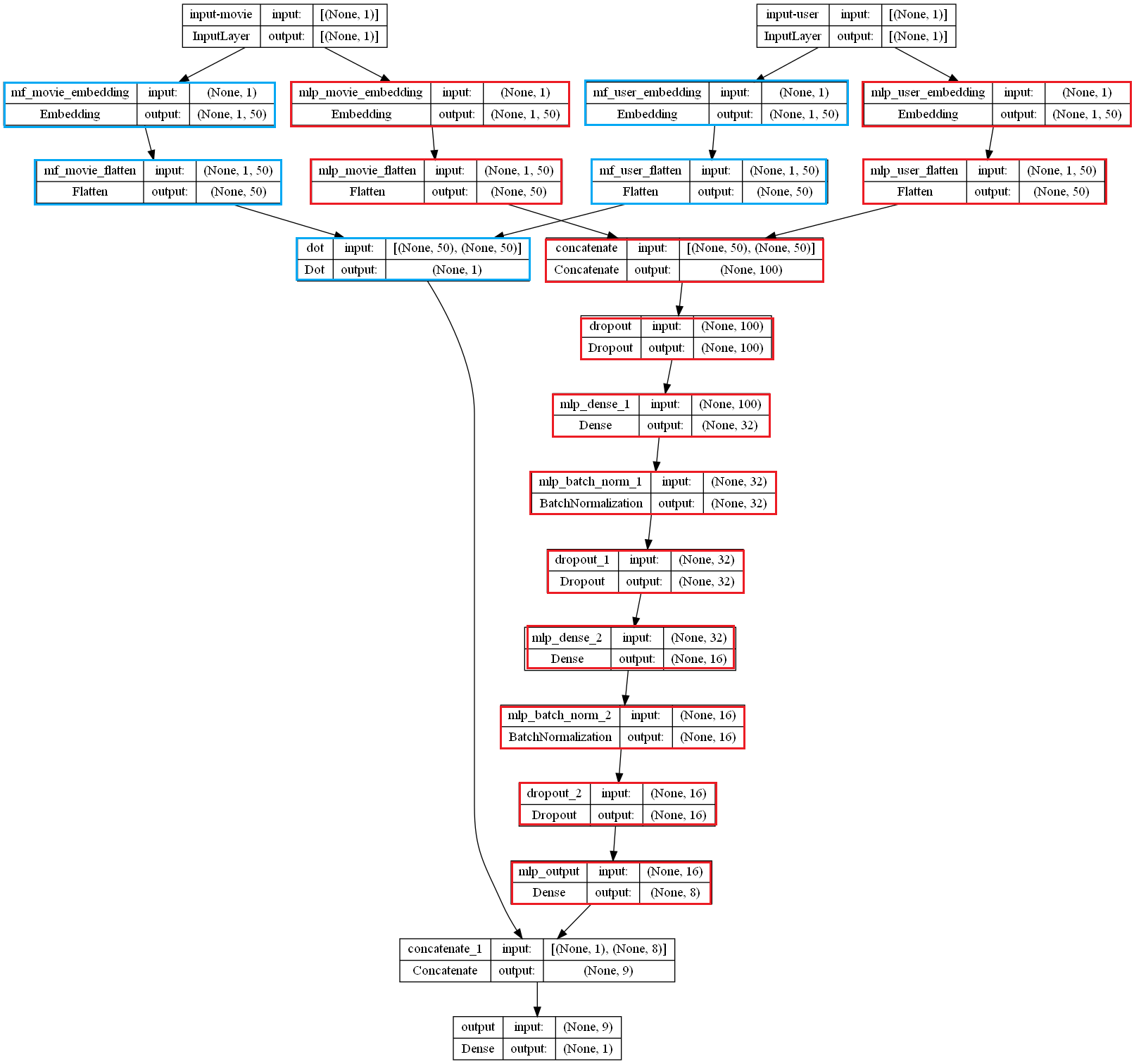
* 1. **System rekomendacyjny filtracji kolaboracyjnej**

Jako system filtracji kolaboracyjnej wybrany został model Neural Collaborative Filtering przedstawiony przez He et al. [26]. Neural Collaborative Filtering łączy nieliniowy model głębokiej sieci neuronowej (MLP) oraz faktoryzację macierzy. Model ten został zaprojektowany w celu wykorzystania przestrzeni cech ukrytych nie tylko do tradycyjnego filtrowania kolaboratywnego zrealizowanego poprzez faktoryzację macierzy, ale także do w pełni połączonej głębokiej sieci neuronowej. Dodatkowa część ma na celu uchwycenie wzorców i cech, które mogą umknąć faktoryzacji macierzy [26]. Model NCF składa się z 2 części GMF (Generalised Matrix Factorisation) oraz MLP (Multi Layer Perceptron).



Rys. Schemat modelu Neural Collaborative Filtering. Źródło: Neural Collaborative Filtering, He et al. [26].

Schemat zaimplementowanego modelu:

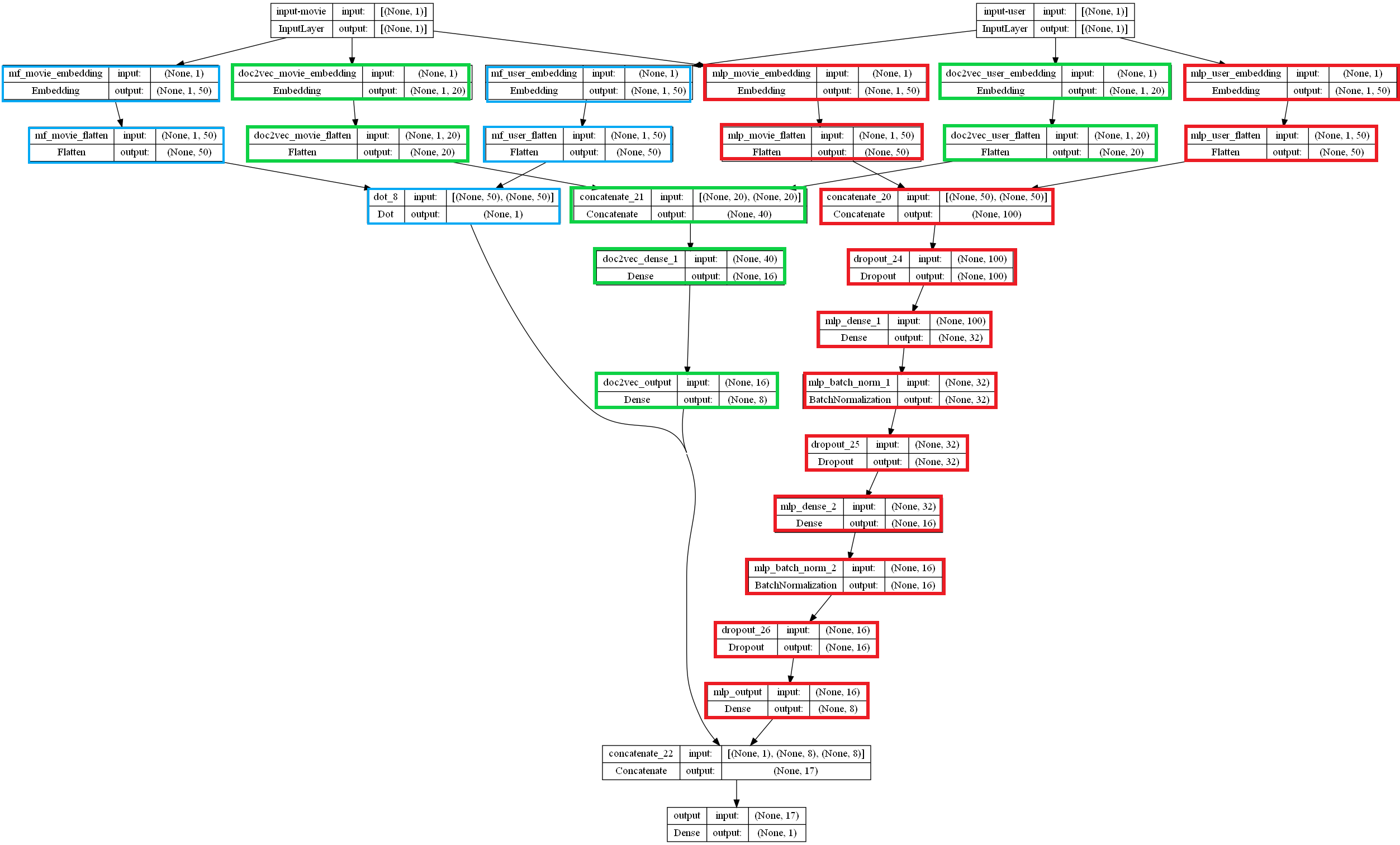


Rys. Schemat modelu NCF. Na czerwono zaznaczone zostały warstwy części MLP, na niebiesko części GMF.

Model NeuCF był uczony przez 10 epok z wykorzystaniem optymalizatora Adam i funkcji kosztu MAE – średniego błędu bezwzględnego.

* 1. **Hybrydowy system**

Model hybrydowy opracowany został poprzez dodanie wektorów filmów i użytkowników uzyskanych z modelu Doc2Vec do sieci modelu Neural Collaborative Filltering. Spodziewanym rezultatem takiego połączenia jest polepszenie predykcji ocen przez model dzięki uzyskaniu dodatkowych informacji ( informacji o zawartości danego filmu oraz profilu danego użytkownika).



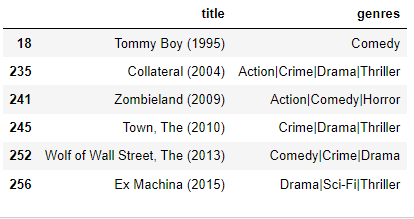
Rys. Schemat modelu hybrydowego. Na czerwono zaznaczone zostały warstwy części MLP, na niebiesko części GMF, na zielono część Doc2Vec.

Model Hybrydowy był uczony przez 10 epok z wykorzystaniem optymalizatora Adam i funkcji kosztu MAE – średniego błędu bezwzględnego.

1. **Analiza działania opracowanych systemów rekomendacyjnych**

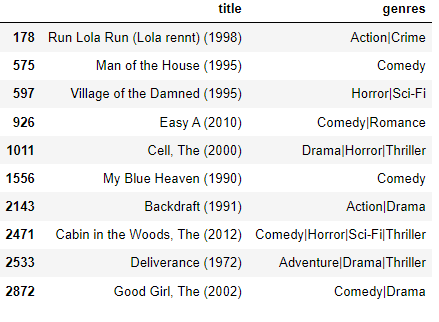
Ewaluacja działania modeli została przeprowadzona poprzez wygenerowanie list rekomendacji dla wszystkich użytkowników przez każdy z modeli. Następnie dla wszystkich modeli obliczone zostały metryki mierzące ich działanie. Dodatkowo, korzystając z każdego z modeli, wygenerowane zostały rekomendacje dla przykładowych użytkowników i zostały one ze sobą porównane empirycznie i przeanalizowane.

Filmy obejrzane przez użytkownika user\_id = 1:



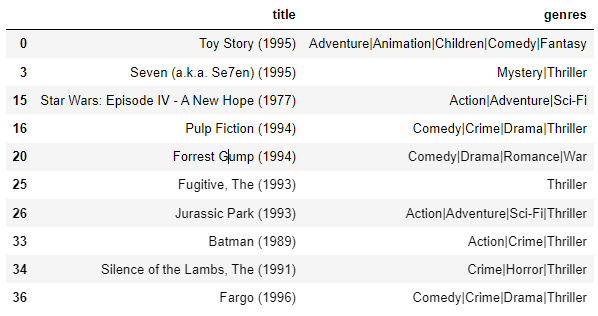
Rys. Filmy obejrzane przez użytkownika user\_id = 1.

Rekomendacje losowe dla użytkownika user\_id = 1:



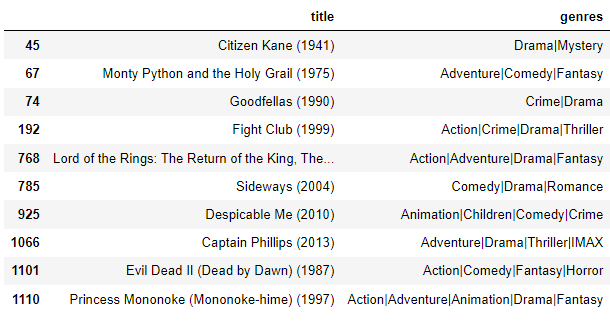
Rys. Filmy zarekomendowane losowo dla użytkownika user\_id = 1.

Rekomendacje popularnościowe dla użytkownika user\_id = 1:



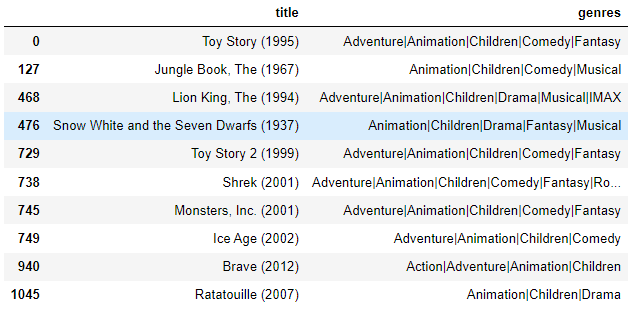
Rys. Filmy zarekomendowane popularnościowo dla użytkownika user\_id = 1.

Rekomendacje SVD dla użytkownika user\_id = 1:



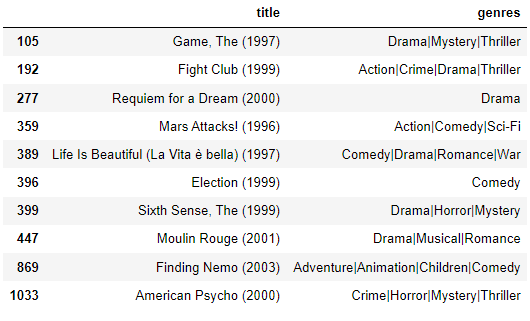
Rys. Rekomendacje SVD dla użytkownika user\_id = 1.

Rekomendacje Doc2Vec dla użytkownika user\_id = 1:



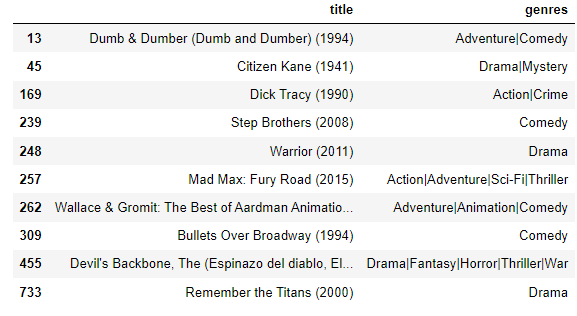
Rys. Rekomendacje Doc2Vec dla użytkownika user\_id = 1.

Rekomendacje NCF dla użytkownika user\_id = 1:



Rys. Rekomendacje NCF dla użytkownika user\_id = 1.

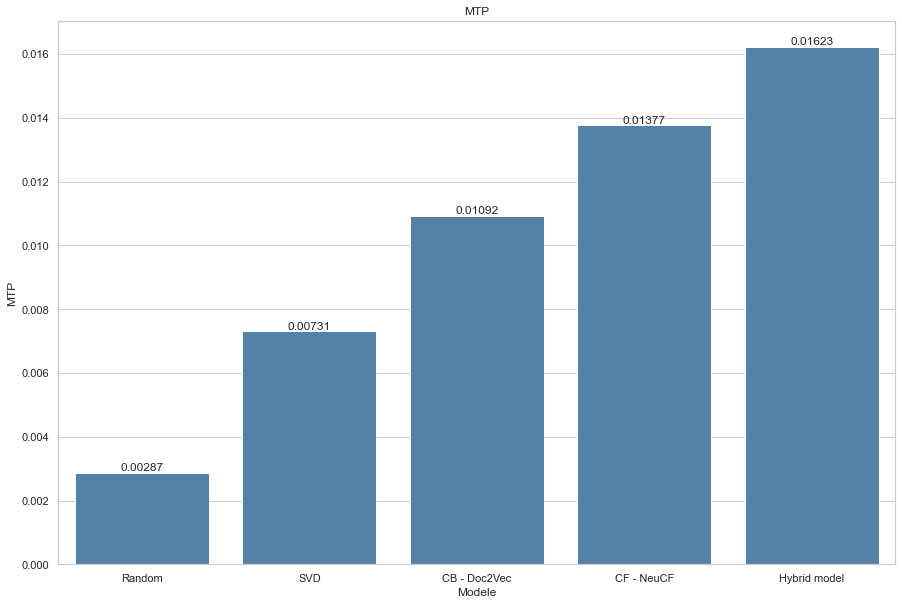
Rekomendacje modelu hybrydowego dla użytkownika user\_id = 1:



Rys. Rekomendacje modelu hybrydowego dla użytkownika user\_id = 1.

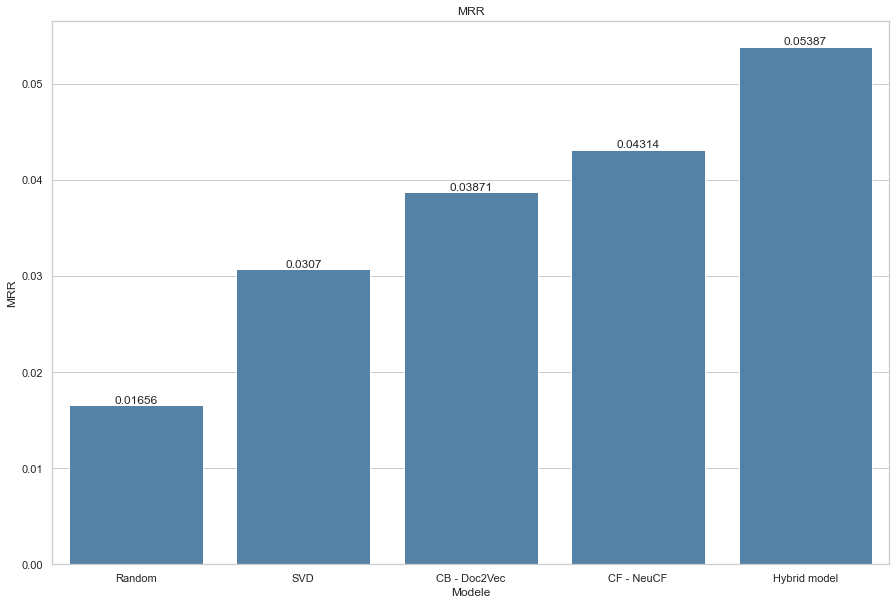
Spośród wymienionych wyżej list z rekomendacjami najbardziej adekwatnymi wydają się być te, zarekomendowane przez modele Neural Collaborative Filtering oraz przez model Hybrydowy.

Po obliczeniu metryk dla wszystkich modeli wyniki prezentują się następująco:



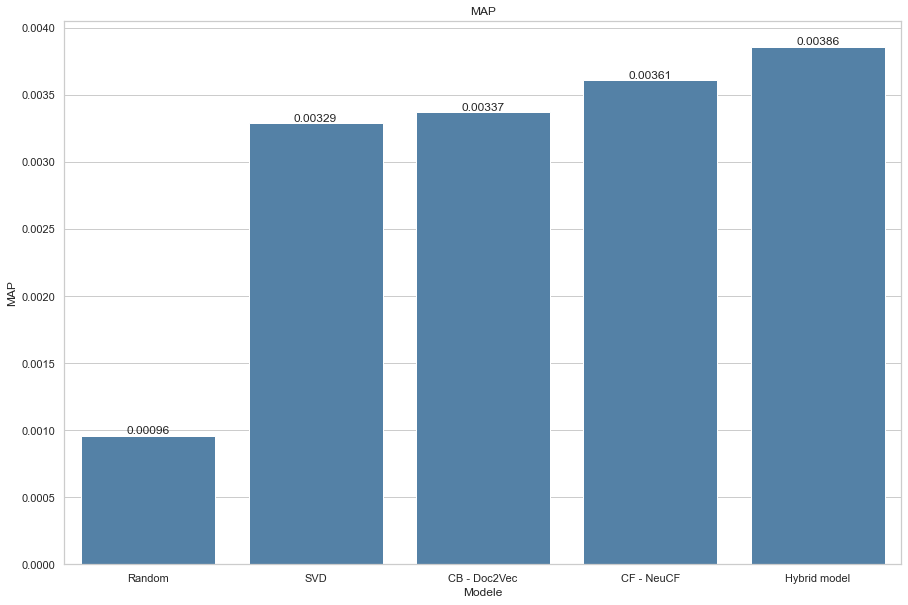
Wykres Wyniki metryki średniej precyzji.

W przypadku wartości średniej precyzji (ilość prawdziwych pozytów/ ilość pozytywów uśrednione przez listy wszystkich użytkowników), wyraźnie widać, że model hybrydowy osiągnął najlepsze rezultaty. Model NCF również uplasował się dość wysoko.



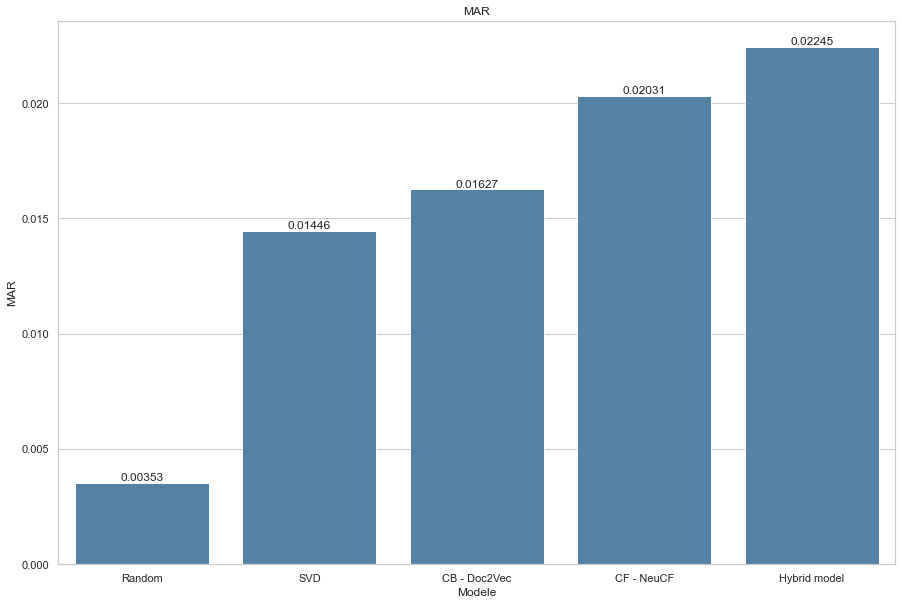
Wykres Wyniki metryki MRR.

W przypadku metryki MRR (uśrednione pierwsze wystąpienia prawdziwych pozytywów w listach rekomendacji), również zaobserwować można, że model hybrydowy osiągnął najlepsze rezultaty.



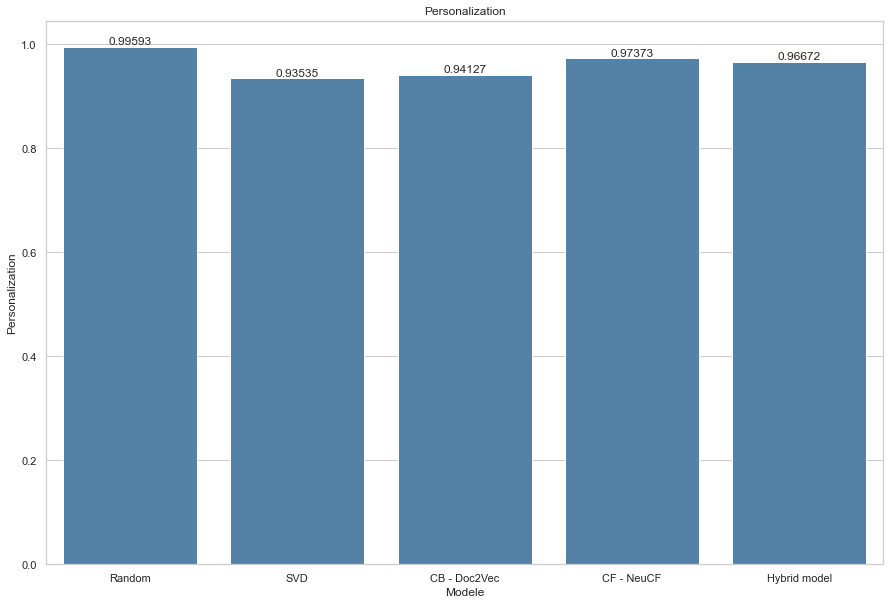
Wykres Wyniki metryki MAP.

Metryka MAP pozwala określić jak trafne są listy rekomendacji. Jak widać, najwyższą wartość tej metryki osiągnął model hybrydowy, pomimo, że różnice pomiędzy modelami zmniejszyły się w przypadku tej metryki, nadal występują one w tej samej kolejności pod względem oceny działania.



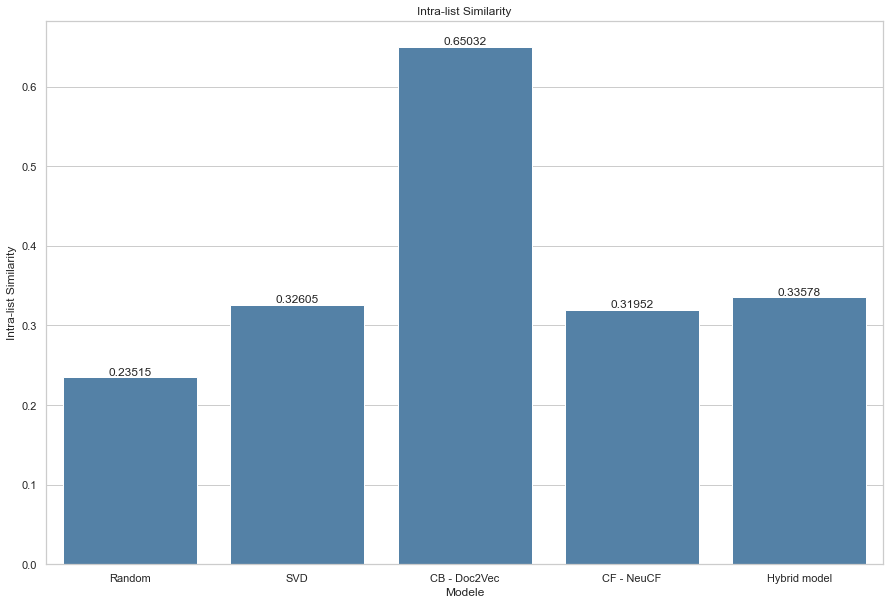
Wykres Wyniki metryki MAR.

Metryka MAR określa w jakim stopniu model jest w stanie zarekomendować przedmioty, które użytkownik ocenił w zbiorze testowym. Jak widać, najwyższą wartość tej metryki ponownie osiągnął model hybrydowy.



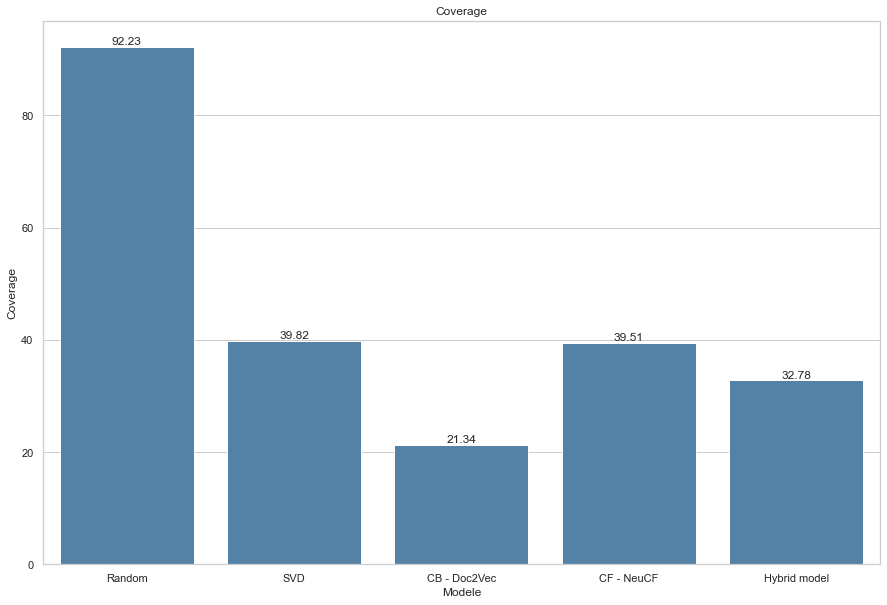
Wykres Wyniki metryki Personalization.

Miara personalizacji określa niepodobieństwo (1 – podobieństwo cosinusowe) pomiędzy listami rekomendacji użytkowników. Wysoka wartość tej metryki oznacza, że model rekomenduje zróżnicowane produkty dla poszczególnych użytkowników. W przypadku modelu losowego, wartość personalizacji bliska 1 jest spodziewana, natomiast w przypadku pozostałych modeli wysokie wartości tej metryki są zastanawiające.



Wykres Wyniki metryki Intra-list Similarity.

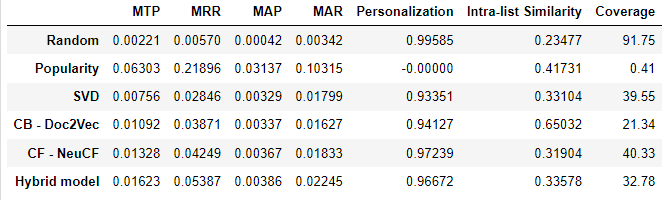
Metryka Intra-list Similarity czyli średnie podobieństwo cosinusowe dla wszystkich przedmiotów we wszystkich listach. Wysoka wartość tej metryki oznacza, że model rekomenduje dla użytkowników listy przedmiotów bardzo do siebie podobnych (np. użytkownik otrzymuje listę z samymi komediami). Na uwagę zasługuje tutaj model Doc2Vec, osiągający prawie 2 razy wyższą wartość tej metryki w porównaniu do innych modeli.



Wykres Wyniki metryki Coverage.

Metryka ta pokazuje jaki procent przedmiotów ze zbioru uczącego model jest wstanie zarekomendować w zbiorze testowym. Zgodnie z przewidywaniami model losowy osiągnął wynik niemal 100%. Ciekawym wydaje się fakt, że modele SVF oraz NCF osiągnęły podobny, wysoki wynik, natomiast model Doc2Vec osiągnął wynik najgorszy.

Poniżej przedstawione zostały omówione wyniki w formie zbiorczej:



Rys. Wyniki wszystkich metryk dla wszystkich modeli.

Model popularnościowy osiągnął znacznie wyższe wyniki niż pozostałe modele. Wytłumaczyć można to tym, że jak wspomniano w podrozdziale 3.1, wykorzystany zbiór MovieLens charakteryzuje się rozkładem typu „długi ogon” i filmy najpopularniejsze wywierają silny wpływ na rekomendacje.

Nie uwzględniając modelu popularnościowego, w większości metryk model hybrydowy osiągnął najwyższe lub jedne z najwyższych wartości wśród zaimplementowanych modeli. Widać również wyraźnie, że po złączeniu składowych modeli (model oparty na treści Doc2Vec oraz model filtracji kolaboracyjnej NCF) w model hybrydowy wyniki metryk dla hybrydowego modelu były zawsze wyższe niż wyniki metryk jego składowych modeli.

1. **Podsumowanie**

W ramach pracy omówione zostały pojęcia sieci neuronowych, uczenia głębokiego oraz systemów rekomendacyjnych. Przedstawione zostały rodzaje SR, ich historia, wymagania stawiane przed systemami, a także zastosowanie w nich uczenia głębokiego.

W części praktycznej pracy dokonano badań eksploracyjnych zbioru danych filmowych MovieLens. Zaimplementowanych zostało 6 systemów rekomendacyjnych: system losowy, system popularnościowy, system filtracji kolaboracyjnej SVD, system filtracji opartej na treści wykorzystujący algorytm Doc2Vec, system Neural Collaborative Filtering wykorzystujący uczenie głębokie oraz system hybrydowy łączący modele Doc2Vec i NCF.

Przeprowadzono analizę wspomnianych systemów, poprzez porównanie szeregu metryk wykorzystywanych do ewaluacji SR. Celem owej analizy było zidentyfikowanie, który z systemów rekomendacyjnych daje najlepsze wyniki i jest najbardziej skuteczny w generowaniu trafnych rekomendacji dla użytkowników.

Wyniki eksperymentów wskazały, że podejście hybrydowe, osiągnęło najlepsze rezultaty.

BIBLIOGRARFIA

1. How Many Videos are Uploaded to YouTube a Day in 2023? Dostęp 2023 https://thesmallbusinessblog.net/how-many-videos-are-uploaded-to-youtube-a-day/
2. The number of tweets per Day. Dostęp 2023   
   https://www.dsayce.com/social-media/tweets-day/
3. Przeciążenie informacyjne wyzwaniem dla edukacji doby cyfrowej Wojciech Marcin Czerski. 2020
4. Media jako instrument wpływu informacyjnego i manipulacji społeczeństwem [15-68] Hanna Batorowska, Rafał Klepka, Olga Wasiuta. 2019
5. Recommender Systems Handbook Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor. 2021
6. Information Overload and Usage of Recommendations Muhammad Aljukhadar, Sylvain Senecal. 2010
7. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions Gediminas Adomavicius, Alexander Tuzhilin. 2005
8. MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS, Yehuda Koren, Robert Bell, Chris Volinsky. 2009
9. Deep learning Yann LeCun, Yoshua Bengio & Geoffrey Hinton. 2015
10. The Hundred Page Machine Learning Book, Andriy Burkov, 2019
11. <https://www.analyticsvidhya.com/blog/2016/08/evolution-core-concepts-deep-learning-neural-networks/> . Dostęp 2023
12. Activation Functions: Comparison of trends in Practice and Research for Deep Learning

Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, Stephen Marshall. 2018

1. Deep Learning Recommendation Model for Personalization and Recommendation Systems Maxim Naumov et. Al. 2019
2. A Brief History of Recommender Systems Zhenhua Dong, Zhe Wang, Jun Xu, Ruiming Tang, Jirong Wen. 2017
3. Recommender system history. Dostęp 2023 <https://en.wikipedia.org/wiki/Recommender_system#History>
4. Evaluating Collaborative Filtering Recommender Systems JONATHAN L. HERLOCKER, JOSEPH A. KONSTAN, LOREN G. TERVEEN, and JOHN T. RIEDL . 2004
5. Hybrid Web Recommender Systems Robin Burke. 2007
6. Recommender Systems An Introduction DIETMAR JANNACH MARKUS ZANKER ALEXANDER GERHARD FRIEDRICH. 2011
7. A survey and critique of deep learning on recommender systems Lei Zheng. 2016
8. Netflix Update: Try This at Home, Simon Funk. Dostęp 2023. <https://sifter.org/~simon/journal/20061211.html>
9. Comparative study of recommender system approaches and movie recommendation using collaborative filtering, Taushif Anwar, V. Uma. 2021
10. Deep Learning based Recommender System: A Survey and New Perspectives, Shuai Zhang, Lina Yao, Aixin Sun, Yi Tay. 2019
11. Setting Goals and Choosing Metrics for Recommender System Evaluations, Gunnar Schröder, Maik Thiele, Wolfgang Lehner. 2011
12. Are we really making much progress? A worrying analysis of recent neural recommendation approaches, Maurizio Ferrari Dacrema, Paolo Cremonesi, Dietmar Jannach. 2019
13. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison, Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, Cong Geng. 2020
14. Neural Collaborative Filtering, Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, Tat-Seng Chua. 2017
15. Neural Collaborative Filtering vs. Matrix Factorization Revisited, Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020
16. Rethinking Neural vs. Matrix-Factorization Collaborative Filtering: the Theoretical Perspectives, Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, Kannan Achan. 2021
17. Distributed Representations of Sentences and Documents, Quoc Le, Tomas Mikolov, 2014