DOI: 10.15199/59.2023.2.3

Systemy rekomendacji
Remedium w cyfrowej klęsce urodzaju

XXXXX

mgr inż. KAROL CHĘCIŃSKI,
inż. RADOSŁAW WAWRZUSIAK,
Redge Technologies Sp. z o.o.

W dzisiejszych czasach ogromnego rozwoju technologicznego dostępność filmów i innych treści audio-wizualnych dla użytkowników stała się olbrzymia. Serwisy streamingowe takie jak Netflix, HBO Max, Amazon Prime Video, Disney+, Player.pl,

STRESZCZENIE: W dobie dynamicznego rozwoju branży OTT, konsument ma dostęp do setek tysięcy atrakcyjnych treści wideo oferowanych przez właścicieli serwisów streamingowych oraz dystrybutorów treści. Remedium na tytułową klęskę urodzaju stanowią systemy rekomendacyjne, które stają się powoli niezbędne dla rozwoju serwisów internetowych oferujących produkty lub treści. Funkcjonalność systemów rekomendacyjnych nie polega jednak tylko na przewidywaniu ocen użytkowników, ale wymaga wieloaspektowego podejścia. Istotne jest, aby systemy były elastyczne w kontekście obsługi danych i algorytmów oraz były zasilane danymi w czasie rzeczywistym. Autorzy we wprowadzaniu opisują genezę powstania systemów rekomendacyjnych dla serwisów streamingowych, wykorzystując perspektywę zarówno użytkownika jak i właściciela platformy dostarczającej treści. Artykuł omawia cechy dobrych rekomendacji oraz potrzebne dane do ich generowania. W kolejnych sekcjach artykułu przedstawione zostaną podstawy tworzenia systemów rekomendacji, na przykładzie serwisu VOD. Omówione zostaną kluczowe czynniki wpływające na jakość rekomendacji oraz dane potrzebne do ich generowania. Ponadto, poruszone zostały istniejące problemy związane z tworzeniem skutecznych systemów rekomendacyjnych, zarówno teoretyczne jak i praktyczne – takie jak implementacja algorytmów rekomendacyjnych w rzeczywistych systemach. W dalszej części artykułu, przedstawione zostaną różne techniki i podejścia, które mogą być wykorzystane do rozwiązania tych trudności w tworzeniu systemów rekomendacyjnych. Opisane techniki i podejścia zastosowane zostały w systemie rekomendacyjnym – *Redge Media Recommender* – stworzonym przez Redge Technologies. Projekt powstał we współpracy z ActumLab w ramach programu RPO WM 2014–2020.
SŁOWA KLUCZOWE: telewizja internetowa, systemy rekomendacji, streaming, uczenie maszynowe, sztuczna inteligencja

ABSTRACT: In the age of the dynamic development of the OTT industry, consumers have access to hundreds of thousands of attractive video content offered by streaming service providers and content distributors. The remedy for this abundance of choice is recommendation systems, which are becoming essential for the development of Internet services that offer products or content. However, the functionality of recommendation systems does not rely solely on predicting user ratings but requires a multi-faceted approach. It is crucial that the systems are flexible in terms of data and algorithm handling, and that they are powered by real-time data. The authors describe the evolution of recommendation systems for streaming services from the perspective of both the user and the content platform owner. The article discusses the characteristics of good recommendations and the data needed to generate them. The basics of building recommendation systems are presented in the following sections of the article, using a VOD service as an example. Key factors influencing the quality of recommendations and the data needed to generate them are discussed. In addition, existing problems related to the creation of effective recommendation systems are addressed, both theoretical and practical, such as the implementation of recommendation algorithms in real systems. The rest of the paper presents several techniques and approaches that can be used to overcome these difficulties in building recommendation systems. The described techniques and approaches have been applied in the recommendation system – Redge Media Recommender – developed by Redge Technologies. The project was developed in cooperation with ActumLab as part of the RPO WM 2014–2020 programme.

KEYWORDS: Internet TV, recommendation systems, streaming, machine learning, artificial intelligence

TVP VOD, Play NOW, CANAL+ Online, Go3 i wiele innych oferują dostęp do tysięcy filmów, seriali i programów telewizyjnych na życzenie. Serwis IMDB będący bazą filmową i telewizyjną online, zawiera ponad 6,3 miliona tytułów [1]. Spotify, strumieniowa platforma oferująca dostęp do treści audio, szczyci się bazą 100 milionów utworów oraz 5 milionów podcastów [2]. Serwis BookBeat, oferujący dostęp do audiobooków oraz e-booków ma w swojej ofercie 800 tysięcy pozycji [3].

Współcześnie każdy dostawca treści cyfrowych może zaoferować tysiące, jeśli nie miliony produktów. W czasach tak ogromnej różnorodności dostępnych multimediów mamy do czynienia z klęską urodzaju. Obszerna baza treści dla użytkowników może prowadzić do problemów z wyborem odpowiedniego filmu lub programu, co może wpłynąć na jakość doświadczenia użytkownika. Z jednej strony, użytkownicy mają teraz dużo większy wybór i mogą łatwiej znaleźć coś dla siebie, ale z drugiej strony, to ogromna liczba dostępnych tytułów może być przytłaczająca i sprawić, że wybór staje się trudny. Przy tej skali dotarcie przez użytkownika do treści, które mogą go zainteresować, jest mocno utrudnione, a zapoznanie się z całą ofertą fizycznie niemożliwe. Naturalnie przekłada się to na zysk, jaki serwis może czerpać z oferowania ich.

Powszechnym rozwiązaniem tego problemu jest wykorzystanie wyszukiwarek. Użytkownik podając wybraną przez siebie frazę może dzięki nim sprawdzić, czy interesujący go produkt jest dostępny w serwisie. By taka ścieżka przyniosła efekt, użytkownik sam musi mieć pomysł czego konkretnie szuka i wyjść z własną inicjatywą. Pula treści, do których w ten sposób dotrze będzie jednak nadal ograniczona jedynie do tych, których istnienia jest świadomy, więc potencjał bogatej oferty w dalszym ciągu nie ma szans na bycie wykorzystanym. Innym sposobem, w którym inicjatywa leży już po stronie serwisu, są wszelkiego rodzaju zestawienia, podsumowania, czy rankingi, przygotowywane przez zatrudnionych przez daną platformę redaktorów, w których będą oni proponować użytkownikom wartościowe i sprawdzone treści dostępne w serwisie. Rozwiązanie te jednak nie rozwiązuje problemu, ponieważ nie zawsze subiektywny ranking materiałów trafi w gust każdego użytkownika.

Narzędziem, które pozwala zniuansować przekaz i dostarczyć każdemu z użytkowników spersonalizowane propozycje w sposób automatyczny, jest system rekomendacji. Pojęcie systemu rekomendacji odnosi się do oprogramowania oraz technik, które dostarczają sugestie produktów, które z największym prawdopodobieństwem będą odpowiadać preferencjom poszczególnych użytkowników [4]. Za tą prostą definicją kryje się jednak wiele interesujących wyzwań i niuansów, którym firma Redge Technologies we współpracy z ActumLab w ramach programu RPO WM 2014-2020 miała okazję sprostać.

Mimo, że to ostatnie lata przynoszą lawinowy wzrost liczby dostępnych treści cyfrowych, potrzeba istnienia rekomenderów została dostrzeżona o wiele wcześniej. W 1992 firma Xerox Palo Alto Research stworzyła Tapestry – system mailingowy wyposażony w pierwszy system rekomendacyjny oparty o model typu *collaborative filtering*, w którym do filtrowania maili mogących interesować poszczególnych użytkowników wykorzystywane były informacje o reakcjach innych użytkowników na te maile [4]. Potencjał tego typu modeli wkrótce został połączony z modelami typu *content-based* przez twórców serwisu Fab – adaptacyjnego systemu rekomendacji dla stron internetowych. [5]. System rekomendacji oparty o dwuwarstwową architekturę, pozwalającą na uwzględnianie danych archiwalnych i danych napływających do systemu na bieżąco, został zaproponowany i wdrożony dla systemu telewizji internetowej należącej do Fastweb [6]. Podział systemu na kilka warstw, pozwalających na użycie złożonych modeli uczenia maszynowego oraz szybkiego uwzględnienia najnowszych danych, to popularne podejście obecne w systemach takich jak Netflix [7], czy Alibaba [8].

W ramach tego artykułu, przez rekomendację rozumie się posortowaną według trafności listę produktów wygenerowaną przez system w odpowiedzi na zapytanie.

Niniejszy artykuł w kolejnych sekcjach wprowadza do tematyki systemów rekomendacji na przykładzie systemu dla serwisu VOD. Przedstawia, czym charakteryzują się dobre rekomendacje i jakie dane są potrzebne do ich generowania. Opisuje związane z tym zagadnieniem problemy, zarówno w sferze teoretycznej, jak i w praktycznej, tj. implementacji algorytmów rekomendacyjnych w rzeczywistym systemie. Dalsza część opisuje, jakie techniki i podejścia zostały zastosowane do rozwiązania uprzednio scharakteryzowanych trudności w systemie rekomendacji stworzonym przez Redge Technologies.

OPIS PROBLEMU

W ramach tego artykułu, przez rekomendację rozumie się posortowaną według trafności listę produktów wygenerowaną przez system w odpowiedzi na zapytanie. Zapytanie obejmuje przede wszystkim identyfikator użytkownika, dla którego rekomendacje mają zostać wygenerowane. Może zawierać także informacje o kontekście, takim jak pora dnia, czy typ urządzenia, z którego użytkownik aktualnie korzysta.

Rekomendacje mogą być generowane w ramach zróżnicowanych scenariuszy. Przykładowo, na stronie głównej serwisu mogą znajdować się ogólne sekcje, gdzie prezentowane będą dopasowane do użytkownika materiały pochodzące z dostępnej biblioteki. Na stronie konkretnego produktu mogą być natomiast wyświetlane rekomendacje produktów podobnych do danego. Również wyszukiwarka może być wsparta poprzez rekomendacje. Produkty pasujące do wprowadzonej przez użytkownika frazy mogą być uszeregowane według przewidywanej preferencji. W takim przypadku fraza wyszukiwania może być jednym z elementów zapytania do rekomendera.

Generowanie spersonalizowanych rekomendacji wymaga dostępu do danych. Przede wszystkim o produktach, użytkownikach oraz interakcjach między nimi. To, co dokładnie będzie znajdowało się w tych danych, będzie zależało od konkretnego serwisu i dziedziny, dla której rekomendacje mają być dostarczane.

Najistotniejszym zasobem są **interakcje**. Można je podzielić na jawne i niejawne. Do jawnych zaliczyć można przykładowo oceny w skali od 1 do 5, czy też znane z serwisów społecznościowych "lajki". Istotne jest, że interakcje tego typu wynikają z bezpośredniego wyrażenia przez użytkownika preferencji na temat produktu. Ten typ interakcji jest jednak problematyczny z tego względu, że użytkownicy często niechętnie wyrażają takie opinie, więc gromadzenie tego typu danych bywa problematyczne. W kontraście stoją interakcje niejawne, które wynikają z zachowań i akcji wykonywanych przez użytkownika. Do tej grupy można zaliczyć np. kliknięcie w produkt, czas spędzony na stronie produktu, czy dokonanie zakupu. Takie dane cechują się dużym zaszumieniem. Fakt, że użytkownik obejrzał film w całości, nie daje gwarancji, że mu się podobał. Pozwala o tym wnioskować jedynie z pewnym prawdopodobieństwem. Gdy mierzony jest czas spędzony przez użytkownika na czytaniu artykułu, należy mieć na uwadze sytuacje, w których użytkownik odszedł od komputera pozostawiając otwartą stronę itd. Zaletą interakcji niejawnych jest łatwość ich pozyskiwania. Gromadzone są automatycznie, gdy użytkownicy korzystają z serwisu. Dzięki temu możliwe jest dostarczanie spersonalizowanych treści także dla użytkowników, którzy niechętnie dzielą się swoją opinią.

Kolejną grupę danych stanowią **atrybuty produktów**. W przypadku filmów mogą to być przykładowo informacje o gatunku, reżyserze, aktorach, czy kraju produkcji. Wartościowy może być także opis filmu, z którego można wydobyć słowa kluczowe, przydatne w charakteryzowaniu filmu [11]. Takie informacje pozwalają na porównywanie produktów między sobą. Istotne są również bardziej techniczne atrybuty, takie jak okres dostępności danego materiału, czy ograniczenia wiekowe. Ich uwzględnienie leży poza, tematem algorytmów rekomendacyjnych, jednak pozwala chociażby na wstępne odfiltrowanie produktów, które nie powinny być rekomendowane.

Analogicznie do produktów, także użytkownicy mogą być scharakteryzowani przez zestaw atrybutów. Podobnie jak wyżej, można użyć ich do wyszukiwania podobieństw oraz do realizacji logiki biznesowej po stronie systemu.

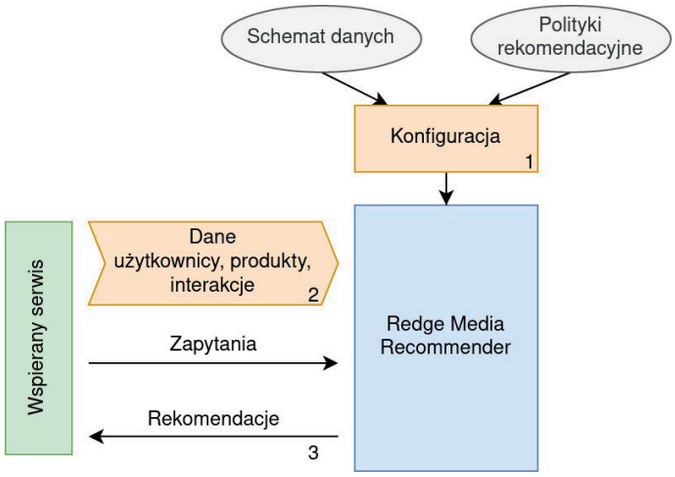
PERSONALIZACJA TO NIE WSZYSTKO

Przed dokonaniem oceny jakości systemu generującego rekomendacje, należy ustalić, jakie cechy powinny być spełnione przez generowane rekomendacje. Ostatecznym celem wdrożenia systemu rekomendacji nie jest personalizacja sama w sobie, a długofalowa poprawa satysfakcji użytkownika, która przekładać się będzie w wymierny sposób na jego utrzymanie oraz płynące z tego faktu zyski. Wobec tego, wyróżnić można 4 podstawowe własności, które powinny spełniać dostarczane rekomendacje: **trafność** (ang. *relevance*), **różnorodność** (ang. *diversity*), **nowość** (ang. *novelty*) i **zdolność do przypadkowych odkryć** (ang. serendipity) [10].

Trafność jest bezpośrednio związana ze spersonalizowaniem rekomendowanych treści. Rekomendacje są trafne, jeśli są zgodne z preferencjami użytkownika. Oparcie systemu rekomendacji wyłącznie na trafności jest jednak niewystarczające. Przykładowo, jeśli system określi, że ulubionym typem filmów danego użytkownika są horrory z lat osiemdziesiątych, to za najbardziej pasujące do jego profilu zostaną uznane właśnie filmy tego typu i przy zastosowaniu trafności jako jedynego kryterium, wyłącznie takie filmy trafią do rekomendacji. W ten sposób zmniejsza się prawdopodobieństwo, że wśród zarekomendowanych produktów znajdzie się taki, który zainteresuje w danym momencie użytkownika. Chcąc rozwiązać ten problem, należy zadbać również o to, by rekomendacje były odpowiednio różnorodne. Przedstawiały produkty różnych typów, nawet kosztem trafności.

Dobra perspektywa użytkowania systemu wymaga, aby system polecał użytkownikowi również nowe materiały. Sugestie powtarzające się przez dłuższy czas nie będą dla użytkownika pomocne. Istotne jest również, by w rekomendacjach uwzględniać treści niespodziewane. Warto pamiętać, że algorytmy opierają się m.in. na dotychczasowej aktywności użytkownika, która z kolei wynika z treści proponowanych przez algorytmy.. W ten sposób powstaje pętla sprzężenia zwrotnego, która może doprowadzić do szybkiego "zaszufladkowania" użytkownika. Uwzględnienie w wynikach elementów niespodziewanych, które nie pasują do stworzonego profilu, umożliwia poznanie preferencji w szerszym zakresie.

Działanie systemu rekomendacyjnego nie ogranicza się wyłącznie do przewidywania preferencji użytkownika. Ostatecznie działa on jako wsparcie dla serwisu, który może kierować się dodatkowymi regułami biznesowymi. Przykładowo, serwis może nie chcieć wyświetlać w rekomendacjach produktów, które nie są obecnie dostępne, mimo, że są zdefiniowane. W przypadku serwisu VOD użytkownicy mogą mieć wykupiony dostęp jedynie do części dostępnej biblioteki. Od polityki serwisu będzie zależało, czy takie filmy mają być pomijane w rekomendacjach, czy wręcz przeciwnie, mają być dodatkowo promowane, by skłonić klienta do zakupu. Mogą pojawić się również kwestie ograniczeń wiekowych, czy ustawień kontroli rodzicielskiej. Są to zagadnienia, które nie są brane pod uwagę w prostych algorytmach rekomendacyjnych. Mogą być realizowane po stronie serwisu, jednak zdecydowanie bardziej praktyczne jest uwzględnienie ich już na etapie generowania rekomendacji. W ten sposób system nie będzie tracił zasobów na wykonywanie predykcji dla pro-



» *Rys. 1. Schemat działania Redge Media Recommender*

duktów, które i tak zostaną odrzucone. Nie będzie także potrzeby wysyłania dodatkowych zapytań, jeśli na etapie filtrowania po stronie serwisu zostanie odrzuconych zbyt wiele produktów.

OPIS ROZWIĄZANIA

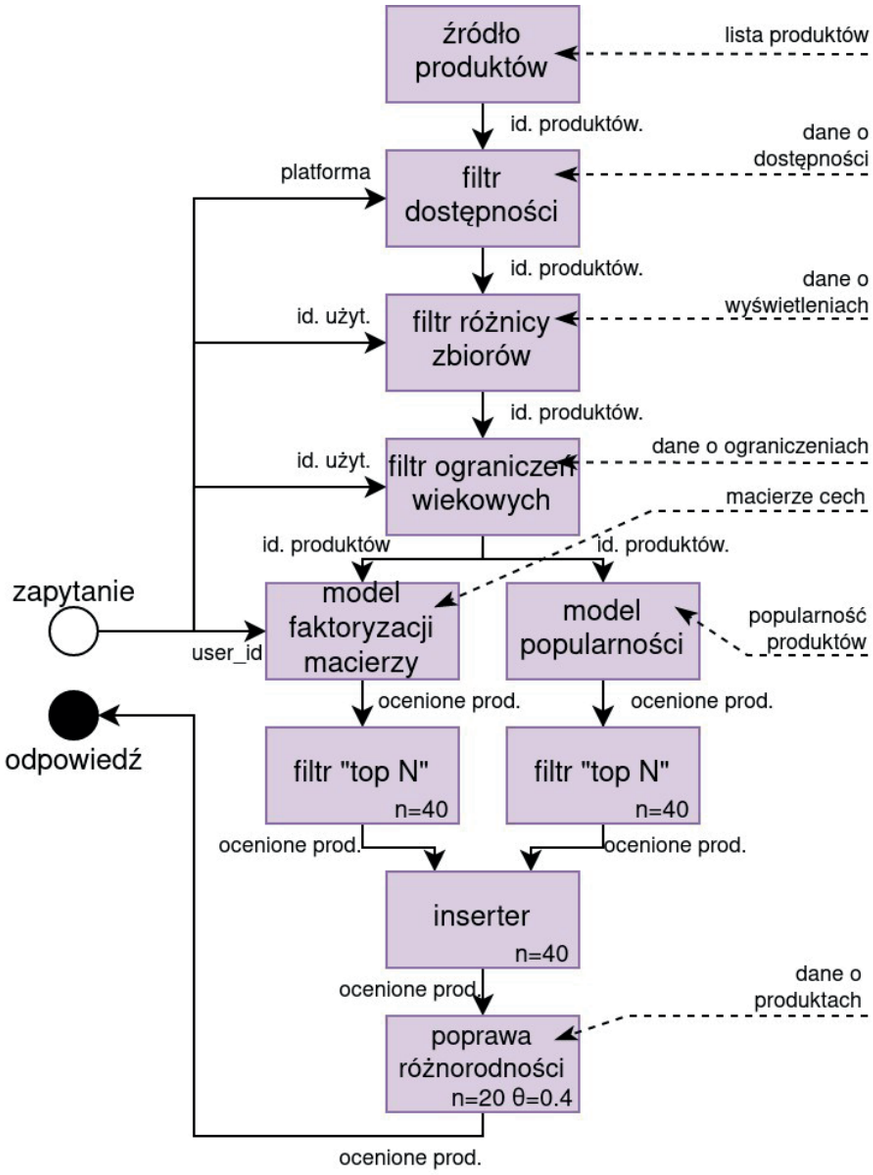
Firma Redge Technologies jako naturalne uzupełnienie swojego ekosystemu zorientowanego wokół przetwarzania i dostarczania multimediów, w odpowiedzi na scharakteryzowany powyżej problem, stworzyła Redge Media Recommender.

Jego działanie obrazuje rys. 1. RMR stanowi samodzielny system, z którym komunikacja odbywa się poprzez API (ang. *Application Programming Interface*). Jego pierwsza część służy do zarządzania systemem i konfigurowania go. Druga służy do zasilania systemu danymi. Trzecia natomiast zwraca rekomendacje w odpowiedzi na wysłane zapytania.

Konfiguracja systemu polega na dostosowaniu go do charakterystyki wspieranego rekomendacjami serwisu. Składają się na nią dwie części. Pierwszą stanowi schemat danych, czyli to, jakimi atrybutami, jakich typów opisywani są użytkownicy i produkty, a także jakie typy interakcji będą przesyłane do systemu. RMR obsługuje zarówno proste (liczby, tekst, wartości logiczne), jak i złożone (listy, słowniki, struktury) typy wartości atrybutów.

Do schematu danych zalicza się także sposób przetwarzania interakcji, którymi zasilany jest system. W niektórych przypadkach, do wyznaczania właściwych wartości interakcji przydatne są inne dane, już zgromadzone w systemie. Przykładowo, serwis może generować interakcje w postaci czasu oglądania danego materiału, który na potrzeby modeli rekomendacyjnych należy znormalizować całkowitym czasem jego trwania. Ze względów praktycznych operacja taka może być problematyczna po stronie serwisu (za generowanie takich interakcji może odpowiadać moduł, który nie ma połączenia z informacjami o filmach). Dlatego RMR oferuje możliwość wykonania takiego przetwarzania wstępnego po swojej stronie.

Drugim aspektem, który może być konfigurowany, są polityki rekomendacyjne. Określają one sposób, w jaki rekomendacje mają być generowane, to znaczy jakie algorytmy i jakie dane mają być użyte. Różne sekcje, czy moduły po stronie serwisu mogą być obsługiwane odrębnymi politykami. Inna polityka będzie wykorzystana do zrealizowania sekcji „podobne filmy", a inna do wyszukiwarki. Przykładowy schemat polityki przedstawia rys. 2. Na jej definicję składa się lista elementów oraz lista połączeń między nimi, która przekłada się na widoczną na schemacie strukturę grafową. Operację wykonywaną przez element w grafie definiuje transformer. Każdy transformer reprezentuje pojedynczy krok w procesie generowania rekomendacji. Generowanie rozpoczyna się od danych przychodzących w zapytaniu. W opisywanym przykładzie jest to identyfikator użytkownika oraz informacja o platformie, z której pochodzi zapytanie. Pierwszym elementem potoku przetwarzania jest źródło produktów, którego celem jest zwrócenie listy identyfikatorów wszystkich produktów. Kolejne 3 elementy mają za zadanie przefiltrować tę listę. Pierwszy filtr – filtr dostępności, odsiewa produkty, które nie są obecnie dostępne na danej platformie. Celem drugiego jest odrzucenie produktów, które dany użytkownik już widział. Trzeci eliminuje z dalszego prze-



» *Rys. 2. Przykładowy schemat polityki*

twarzania elementy niezgodne z ograniczeniami wiekowymi danego użytkownika. Pozostałe produkty trafiają do dwóch modeli rekomendacyjnych, których celem jest przypisanie ocen poszczególnym produktom. Model faktoryzacji macierzy jest przykładem modelu rekomendacyjnego należącego do grupy modeli wspólnego filtrowania (ang. *collaborative filtering*). Pozwala on na przewidywanie ocen na podstawie podobieństwa w aktywności różnych użytkowników. Szerzej będzie opisany w dalszej części tekstu. Drugi model ocenia produkty na podstawie ich popularności, więc poza kontekstem konkretnego użytkownika. Następnie, w obu gałęziach filtry "top N" wybierają z otrzymanych wyników N najwyżej ocenionych produktów. Kolejnym krokiem jest połączenie obu list. Lista produktów oparta o popularność może stanowić uzupełnienie w sytuacjach, gdy model faktoryzacji macierzy nie będzie w stanie wygenerować wyników, np. gdy użytkownik nie ma jeszcze swojej historii aktywności. Można też połączyć listy odpowiednio ważąc punktację z obu modeli. Kolejnym elementem jest transformer odpowiedzialny za zwiększenie różnorodności w wynikowej liście. Spośród produktów wchodzących wybiera ostateczną listę w taki sposób, by składała się ze zróżnicowanych produktów. Stojący za nim algorytm także zostanie opisany w dalszej części tekstu.

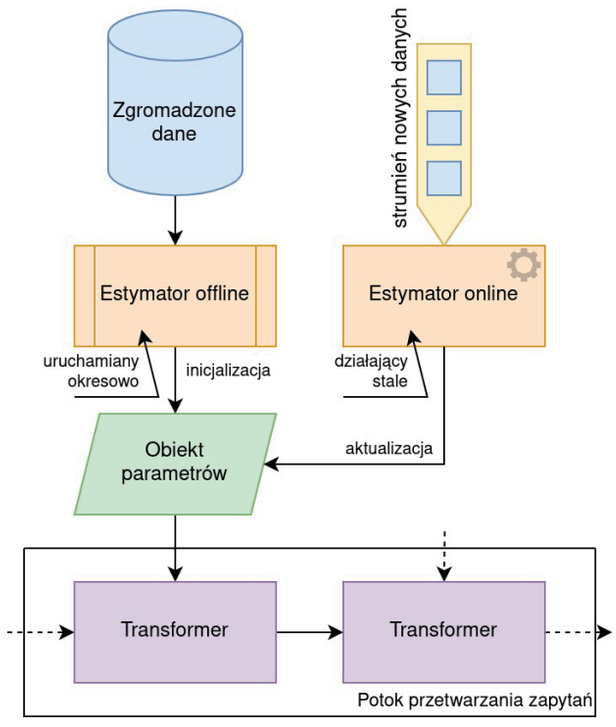
```
{
    "estimator": {
        "params_type": "item_features",
        "estimator_type": "item_features",
        "hyperparams_values": {},
        "train_inputs": {
            "items_attributes": {
                "category": "entities",
                "type_name": "item",
                "attributes_names": ["genres"]
            }
        }
    },
    "transformer": {
        "params_type": "item_features",
        "transformer_type": "item_features_topic_diversification",
        "hyperparams_values": {"n": 20, "theta": 0.4}
    }
}
```

» *Rys. 3. Definicja przykładowego elementu polityki*

Tak wygenerowana lista jest zwracana jako odpowiedź. Znaczna część transformerów, poza danymi wejściowymi, pochodzącymi z zapytania lub innych transformerów, potrzebuje także informacji bazujących na zmagazynowanych w systemie danych. W opisywanym systemie informacje te nazywane są parametrami. Na rys. 2 zaznaczone są przerywanymi strzałkami. Parametry mogą zawierać surowe dane o użytkownikach, produktach lub interakcjach. Tak jest m.in. w przypadku źródła produktów, które jako parametry przyjmuje listę wszystkich identyfikatorów lub filtru dostępności, który korzysta z wartości atrybutu opisującego dostępność produktów. Parametry mogą być również wynikiem prostych przekształceń surowych danych. Ma to miejsce np. dla filtra różnicy zbiorów. W tym przypadku parametry stanowią słownik, który identyfikatory użytkowników mapuje na zbiory identyfikatorów obejrzanych przez nich produktów. Jest on budowany na podstawie zgromadzonych interakcji. Inne parametry mogą być generowane przez algorytmy uczenia maszynowego. Dzieje się tak np. w przypadku modelu faktoryzacji macierzy, który przewiduje oceny produktów na podstawie pochodzących z parametrów macierzy cech.

Za przygotowanie parametrów odpowiedzialne są estymatory. Każdy element polityki, który korzysta z parametrów, poza definicją transformera, posiada także definicje estymatora offline oraz, opcjonalnie, estymatora online. Definicja obejmuje typ estymatora, jego hiperparametry, oraz wskazanie, na podstawie jakich danych estymator ma wygenerować parametry. Działanie estymatora offline sprowadza się do wykonania procesu wsadowego. Wejście do takiego procesu stanowią dane zgromadzone dotychczas w systemie. Wynikiem działania jest obiekt parametrów. Obiekt taki może zostać zapisany do statycznego pliku lub – w postaci modyfikowalnej - do bazy danych typu klucz–wartość, takiej jak Redis [12]. Ze względu na wolumen danych koniecznych do przetworzenia i potencjalną złożoność obliczeniową niezbędnych do osiągnięcia zamierzonego rezultatu algorytmów, wykonanie takiego procesu może być czasochłonne. Z tego powodu problemem staje się utrzymanie aktualności parametrów, tak, by na bieżąco napływające do systemu informacje znajdowały w nich swoje odzwierciedlenie. By rozwiązać ten problem, obok wsadowych estymatorów offline, stworzone zostały estymatory online. Ich celem nie jest tworzenie obiektu parametru od zera, ale modyfikowanie istniejących, na podstawie paczek nowych danych.

Opisana powyżej architektura związana z tworzeniem potoków rekomendacyjnych oraz przetwarzaniem zapytań pozwala na realizację zróżnicowanych zachowań i zadań w obrębie systemu rekomendacji. Poza elastyczną archi-



» *Rys. 4. Schemat procesu generowania parametrów*

tekturą potrzebne są jednak także odpowiednie elementy, które można w ramach niej umieścić, a które dobrze będą spełniać swoją rolę. Kluczowe są modele rekomendacyjne, takie jak wspomniany wyżej **model wspólnego filtrowania,** czy **model bazujący na treści** (ang. *content based*). Modele z tych dwóch grup różnią się wykorzystywanymi danymi, co przekłada się na ich odmienną charakterystykę. Modele wspólnego filtrowania bazują na macierzy ocen, gdzie w kolumnach występują użytkownicy, a w rzędach produkty. W przecięciu danej kolumny i wiersza znajduje się ocena, którą dany użytkownik wystawił produktowi lub wartość interakcji, która miała między nimi miejsce. Macierz ocen jest macierzą rzadką. Znane są wartości jedynie dla niewielkiego ułamka par użytkownik–produkt. Zadaniem algorytmów wspólnego filtrowania jest odgadnięcie brakujących wartości na podstawie tych znanych. Nie potrzebują one do tego żadnej wiedzy na temat poszczególnych produktów. Używanym w RMR algorytmem z tej grupy jest algorytm faktoryzacji macierzy [13], który polega na rozłożeniu macierzy ocen na iloczyn dwóch mniejszych macierzy cech, w taki sposób, by po ich przemnożeniu powstawała macierz jak najbardziej zbliżona do oryginalnej w obrębie znanych wartości. W wyniku mnożenia otrzymywane są również wartości dla par, dla których oceny nie były znane. Wartości te stanowią predykcję modelu. Warto wspomnieć, że ta metoda sprawdza się dobrze dla ocen jawnych, które wprost świadczą o preferencji. W przypadku ocen niejawnych można zastosować modyfikację polegającą na traktowaniu wartości interakcji jako wag w funkcji błędu wykorzystywanej przy optymalizacji macierzy cech [14]. Oba warianty są wspierane przez opisywany system. W obu przypadkach sam proces optymalizowania macierzy cech jest przeprowadzany w obrębie estymatora. Następnie są one zapisywane w postaci parametrów. Mogą być one modyfikowane na podstawie napływających danych przy użyciu odpowiednich metod [15]. Z parametrów korzysta transformer, który na ich podstawie wylicza predykcje dla poszczególnych par.

Algorytmy wspólnego filtrowania będą działać tym lepiej, im większa będzie dostępna baza interakcji. Nie sprawdzą się jednak w przypadku nowych lub nieznanych produktów. W bazie nie pojawią się ich oceny, które mogłyby zostać uwzględnione przez faktoryzację macierzy. Są również podatne na obciążenie ze względu na popularność (ang. *popularity bias*). Produkty popularne, występujące w historii aktywności wielu użytkowników, będą naturalnie rekomendowane częściej. Jest to luka, którą mogą zapełnić algorytmy bazujące na treści. W ich działaniu kluczowe są informacje o produktach. Metody z tej rodziny obejmują przeważnie następujące **etapy** [16]:

1. Przetwarzanie wstępne i ekstrakcja cech.
2. Tworzenie profili użytkowników.
3. Generowanie rekomendacji.

Pierwszy etap ma na celu doprowadzenie dostępnych danych na temat produktów do postaci wektorowej, opartej najczęściej o słowa kluczowe. Osadzenie produktów we wspólnej przestrzeni cech pozwala m.in. na zastosowanie ich w metodach uczenia maszynowego. W drugim etapie, w oparciu o wyznaczone uprzednio cechy oraz interakcje użytkowników tworzone są profile. Profil można rozumieć jako konkretny model służący do predykcji ocen dla danego użytkownika. Jego reprezentacja będzie w znacznej mierze zależeć od używanej metody i sformułowania problemu. Przykładowo, problem rekomendacji można sprowadzić do problemu klasyfikacji binarnej, gdzie produkty są dzielone na te z pozytywną i negatywną preferencją użytkownika. Jako model można wykorzystać przykładowo algorytm *k* najbliższych sąsiadów. Te dwa etapy odbywają się w systemie w obrębie estymatora. Jako parametry generowane są wektory cech produktów oraz profile użytkowników. Etap generowania rekomendacji odbywa się już w transformerze, gdzie znajdujące się w parametrach struktury są używane do wykonania predykcji dla wejściowej listy identyfikatorów produktów.

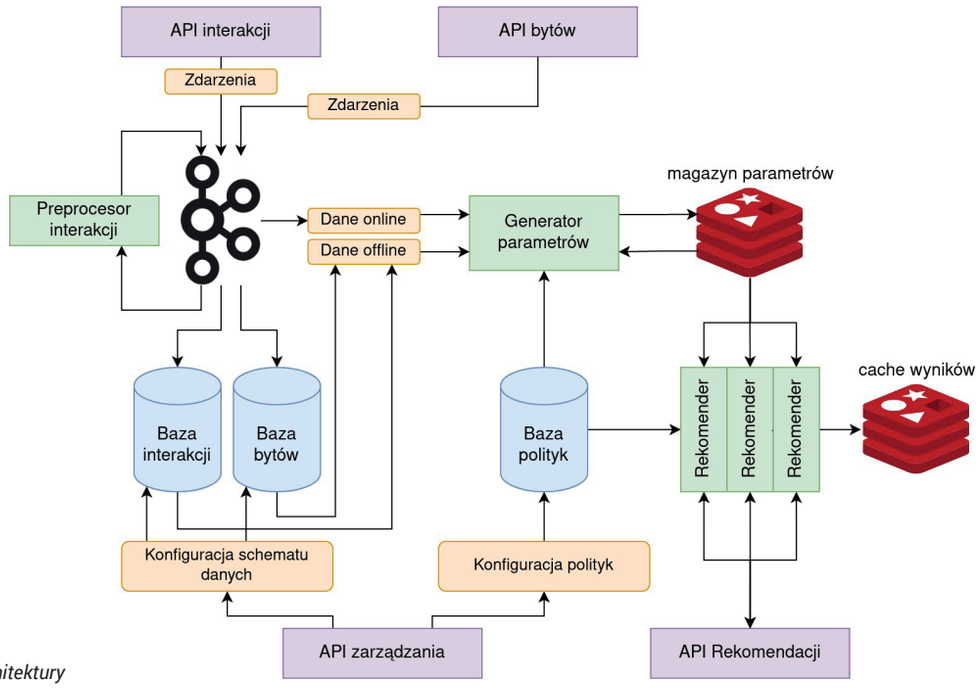
Oba wspomniane typy modeli rekomendacyjnych uzupełniają się, dlatego warto stosować je w politykach rekomendacyjnych równolegle. Ich połączenie pozwala na dostarczanie trafnych treści. Za dobór wynikowej listy rekomendacji tak, by zapewnić zróżnicowane propozycje odpowiada osobny algorytm [17]. Wybiera on z listy produktów wygenerowanej przez model rekomendacyjny kolejne produkty w następujący sposób. Jako pierwszy brany jest najlepszy produkt spośród produktów wejściowych. Następnie dobierane są kolejne produkty maksymalizujące wartość wyrażenia , gdzie jest hiperparametrem określającym jak duży wpływ ma mieć różnorodność na wynik (kosztem trafności). jest średnią odległością rozpatrywanego produktu do produktów z dotychczas zbudowanej listy, oceną produktu dostarczoną przez model, a funkcją, która zwraca numer produktu w malejąco posortowanej liście według danej wielkości, spośród produktów, które pozostały do wyboru. Algorytm jest realizowany w transformerze, a bazuje na wektorach cech produktów dostarczonych przez odpowiedni estymator, generowane w sposób analogiczny do cech generowanych na potrzeby algorytmu opartego o treść.

Innym istotnym w działaniu Redge Media Recommender mechanizmem jest połączenie zwrotne rekomendacji do interakcji. Produkty rekomendowane przez system poszczególnym użytkownikom mogą być rejestrowane jako osobny typ interakcji. Takie interakcje mogą posłużyć do realizacji elementu zapewniającego nowość rekomendacji. Wielokrotnie rekomendowane danemu użytkownikowi produkty mogą być dodatkowo "karane", co będzie zmniejszało ich szansę na pojawienie się w rekomendacjach następnym razem, tworząc przestrzeń dla nowych produktów, których użytkownik jeszcze nie widział.

ARCHITEKTURA

Całość architektury opisywanego systemu przedstawia w uproszczony rys. 5. Oprócz elementów opisanych wyżej, tj. API, generowania parametrów i polityk rekomendacyjnych, na uwagę zasługują również takie elementy jak system obsługi zdarzeń, użycie wielu rekomenderów, czy pamięć podręczna wyników.

System rekomendacji musi być gotowy na przyjmowanie znacznych ilości nowych danych, w szczególności interakcji, ponieważ są one na bieżąco generowane, gdy użytkownicy korzystają ze wspieranej usługi. By zapewnić skalowalność



» *Rys. 5. Schemat architektury systemu Recommender*

tego aspektu, a jednocześnie umożliwić realizację związanych z tym tematem zagadnień, tj. przetwarzania wstępnego interakcji oraz trenowania estymatorów online, wprowadzono system zdarzeń oparty o Kafkę – otwarto-źródłowy rozproszony system przesyłania wiadomości [18]. Każda informacja o dodaniu lub zmianie wartości atrybutu użytkownika bądź produktu, a także każda nowa interakcja, jest dodawana do Kafki jako zdarzenie. Stamtąd zdarzenia mogą być równolegle odczytywane zarówno przez estymatory online, zapisywane do bazy danych, a także użyte i przetworzone przez preprocesor interakcji.

Podobnie do dodawania nowych danych, skalowalności wymaga również generowanie rekomendacji. Zapewniają to dwa mechanizmy. Pierwszym z nich jest równoległe działanie wielu rekomenderów. Tu jako rekomender należy rozumieć moduł odpowiedzialny stricte za generowanie rekomendacji. To w nim załadowane są potoki przetwarzania zapytań złożone z transformerów. Przychodzące do systemu zapytania są rozdzielane pomiędzy działające rekomendery obsługujące poszczególne polityki. Działanie wielu rekomenderów kontroluje proces zarządcy, który może w razie potrzeby tworzyć kolejne instancje, a także stanowi źródło konfiguracji dla poszczególnych instancji w zakresie tego, które polityki mają zostać załadowane. Drugim mechanizmem jest buforowanie odpowiedzi. Wygenerowane odpowiedzi są zapisywane w pamięci podręcznej, realizowanej przy użyciu Redisa, dostępnej dla wszystkich instancji. W ten sposób, jeśli do systemu ponownie przyjdzie zapytanie o wygenerowanie rekomendacji przy użyciu tej samej polityki, dla tego samego użytkownika, możliwe będzie wykorzystanie już gotowego wyniku. Z racji na modyfikowane przez estymatory online parametry, których wartości wpływają na generowane rekomendacje, należy mieć na uwadze ograniczony czas życia wpisu w pamięci podręcznej, by rekomendacje również mogły być aktualizowane.

EKOSYSTEM REDGE

Firma Redge Technologies posiada w swojej ofercie **Redge Media** – kompletną platformę technologiczną do budowy telewizji internetowej. W skład oferty wchodzi Redge Media Service Delivery Platform, która zawiera narzędzia typu DAM i CMS, aplikacje dla różnych platform końcowych oraz posiada gotowość do wdrożeń w środowisku chmury publicznej. Redge Media Delivery Platform to natomiast publiczna chmura dla dostawców treści wideo, oferująca przechowywanie, transkodowanie, zabezpieczanie i dystrybucję multimedialnych treści, w tym system klasy CDN oferujący wieloterabitową pojemność i dostęp do sieci najważniejszych operatorów telekomunikacyjnych w regionie europejskim [19]. System rekomendacji nie tylko stanowi naturalne uzupełnienie tego ekosystemu, ale także sam może korzystać na dostępie do niego. W szczególności VCR – system **Video Content Recognition**, pozwalający na zastosowanie metod uczenia maszynowego i rozpoznawania obrazów na poziomie przetwarzania strumieni wideo, może stanowić źródło dodatkowych metadanych dla filmów, które mogą być uzupełniane automatycznie, bez udziału redaktorów. Dane mogą obejmować przykładowo nastrój, typ scenerii, tempo filmu, czy charakterystyczne obiekty. Są to dane przeważnie niedostępne w standardowych bazach opisujących filmy, co może dać przewagę nad innymi rozwiązaniami, pozwalając bardziej precyzyjne określać preferencje użytkownika.

PODSUMOWANIE

Współcześnie bez systemów rekomendacyjnych trudno wyobrazić sobie działanie serwisów internetowych oferujących dostęp do szerokiej bazy produktów czy treści. Działanie tych systemów nie ogranicza się jedynie do algorytmów predykujących oceny użytkowników, a wymagają wieloaspektowego podejścia do dostarczania rekomendacji. Znaczącym atutem jest zaprojektowanie systemu w sposób, który pozwala na zapewnienie elastyczności zarówno w kontekście obsługiwanych danych, jak i wykorzystywanych algorytmów. Należy także pamiętać, że rekomender nie jest jedynie abstrakcyjnym modelem, a systemem działającym, uczącym się i zasilanym danymi w czasie rzeczywistym, co wymaga zastosowania odpowiedniej architektury. Wszystkie te aspekty udało się zrealizować w Redge Media Recommender, co czyni go atrakcyjnym narzędziem mogącym stanowić wartościowe wsparcie dla dostawców treści.

Zaprezentowane efekty stanowią część prac badawczo-rozwojowych projektu pn. „Prace badawczo-rozwojowe nad stworzeniem platformy do klasyfikacji obiektów opartej o technologię Sztucznej Inteligencji oraz technologię przetwarzania obrazu", nr RPMA.01.02.00-14-b512/18-00 współfinansowanego w ramach Osi Priorytetowej I „Wykorzystanie działalności badawczo-rozwojowej w gospodarce" Działania 1.2 „Działalność badawczo-rozwojowa przedsiębiorstw" Regionalnego Programu Operacyjnego Województwa Mazowieckiego. ●

LITERATURA

[1] Press Room - IMDb: https://www.imdb.com/pressroom/stats/ (dostęp 24.02.2023)

[2] Spotify - About Spotify: https://newsroom.spotify.com/company-info/ (dostęp 24.02.2023)

[3] O BookBeat: https://www.bookbeat.pl/o-nas (dostęp 24.02.2023)

[4] D. Goldberg, D. Nichols, B. M. Oki, D. Terry: "Using collaborative filtering to weave an information tapestry" w *Communications of the AMC*, t. 32, nr 12. 1992, s. 61-70. (https://dl.acm.org/doi/10.1145/138859.138867)

[5] M. Balabanović: "An adaptive Web page recommendation service" w *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, USA, Luty 1997. Nowy Jork, NY, USA: Association for Computing Machinery, Luty 1997, s. 378–385. (https://dl.acm.org/doi/10.1145/267658.267744)

[6] R. Bambini, P. Cremonesi, R. Turrin: "A Recommender System for an IPTV Service Provider: a Real Large-Scale Production Environment" w B. Shapira, L. Rokach, R. Francesco (red.), *Recommender Systems Handbook*. New York: Springer, 2021, s. 299-331. (https://link.springer.com/chapter/10.1007/978-0-387-85820-3_9)

[7] X. Amatriain: "Big & personal: data and models behind netflix recommendations" w *BigMine '13: Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, Chicago, IL, USA, 11.08.2013. Nowy Jork, NY, USA: Association for Computing Machinery, 11.08.2013, s. 1-6. (https://dl.acm.org/doi/10.1145/2501221.2501222)

[8] Basic Concepts and Architecture of a Recommender System - Alibaba Cloud Community: https://www.alibabacloud.com/blog/basic-concepts-and-architecture-of-a-recommender-system_596642 (dostęp 15.03.2023)

[9] F. Ricci, L. Rokach, B. Shapira: "Recommender Systems: Techniques, Applications, and Challenges" w B. Shapira, L. Rokach, R. Francesco (red.), *Recommender Systems Handbook*. New York: Springer, 2021, s. 1-35.(https://doi.org/10.1007/978-1-0716-2197-4_1)

[10] C. C. Aggarwal, "An Introduction to Recommender Systems" w Recommender Systems. Cham: Springer, 2016, s. 1-28. (https://doi.org/10.1007/978-3-319-29659-3_1)

[11] S. Rose, D. Engel, N. Cramer, W. Cowley, "Automatic Keyword Extraction from Individual Documents" w M. W. Berry, J. Kogan (red.), *Text Mining: Applications and Theory.* Wiley, 2010, s. 1-20. (https://doi.org/10.1002/9780470689646.ch1)

[12] Introduction to Redis | Redis: https://redis.io/docs/about/ (dostęp 6.03.2023)

[13] Y. Koren, R. Bell, C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," w *Computer*, tom 42, nr 8, 2009, s. 30-37 (https://doi.org/10.1109/MC.2009.263)

[14] Y. Hu, Y. Koren, C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets" w *2008 Eighth IEEE International Conference on Data Mining,* Pisa, Włochy, 2008, s. 263-272 (https://doi.org/10.1109/ICDM.2008.22).

[15] X. He, H. Zhang, M. Kan, and T. Chua. "Fast Matrix Factorization for Online Recommendation with Implicit Feedback" w *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (*SIGIR '16*). Nowy Jork, USA, 2016, wyd. Association for Computing Machinery, 2016, s. 549–558. (https://doi.org/10.1145/2911451.2911489)

[16] C. C. Aggarwal, "Content-Based Recommender Systems" w *Recommender Systems.* Cham: Springer, 2016, s. 139-166. (https://doi.org/10.1007/978-3-319-29659-3_4)

[17] C. Ziegler, S. M. McNee, J. A. Konstan, G. Lausen, "Improving recommendation lists through topic diversification" w *Proceedings of the 14th international conference on World Wide Web* (WWW '05). Nowy Jork, USA, 2005, wyd. Association for Computing Machinery, 2016, s. 22–32. (https://doi.org/10.1145/1060745.1060754)

[18] Apache Kafka: https://kafka.apache.org/ (dostęp 10.03.2023)

[19] Poznaj nasze flagowe produkty | Redge Technologies: https://redge.com/pl/nasze-rozwiazania/ (dostęp 9.03.2023)

Projekt realizowany wspólnie z Actum Lab Sp. z o.o.