WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

Rozprawa doktorska

Algorytm rekomendacji bazujący na sesjach rekomendacji działający na podstawie zachowań użytkowników oraz atrybutów obiektów w systemie e-Commerce

mgr inż. Michał Malinowski

Promotor

dr hab. inż. Ryszard Antkiewicz

Warszawa 2022

Pragnę w szczególności podziękować Promotorowi dr hab. inż. Ryszardowi Antkiewiczowi za poświęcony czas oraz wiele rzeczowych i metodycznych wskazówek, dzięki którym powstała ta praca.

Chciałbym również podziękować rodzinie – w szczególności mojej żonie Angelice, synom Hubertowi i Szymonowi oraz rodzicom, teściom, wujkowi i bliskim znajomym. Bez ich wsparcia i wyrozumiałości praca nad tą rozprawą byłaby niemożliwa.

Spis treści

Spis skrótów i oznaczeń 4
1. Wstęp 5
2. Obszar e-Commerce 9
2.1. Zachowania użytkowników 10
2.2. Pierwotne pojęcie sesji 12
2.3. Atrybuty obiektów w systemie e-Commerce 14
2.4. Baza danych transakcji (koszyk zakupów) 16
2.5. Pojęcia związane ze strumieniem kliknięć 20
2.6. Czas odpowiedzi systemu e-Commerce 20
3. Obszar rekomendacji 23
3.1. Definicje i formy rekomendacji 23
3.2. Techniki rekomendacji 25
3.2.1. Content Based Filtering 25
3.2.2. Collaborative Filtering 26
3.2.3. Oparte o wiedzę 28
3.2.4. Bazujące na demografii 28
3.2.5. Bazujące na analizie asocjacji 28
3.2.6. Bazujące na sesjach 29
3.2.7. Bazujące na grafach heterogenicznych 30
3.2.8. Hybrydowe 33
3.3. Systemy rekomendacji 33
3.3.1. Fazy procesu rekomendacji 34
3.3.2. Główne problemy przy stosowaniu rekomendacji 34
3.4. Sformułowanie problemu badawczego rekomendacji 36
3.4.1. Obiekt 38
3.4.2. Użytkownik 38
3.5. Miary oceny systemów rekomendacji 39
4. Algorytmy rekomendacji oparte na wykorzystaniu analizy asocjacji 43
4.1. Analiza asocjacji 43
4.2. Algorytmy odkrywania reguł asocjacji 45
4.3. Algorytm rekomendacji bazujący na regułach asocjacji 50
5. Algorytm Rekomendacji Sesji 53

5.1. Algorytm rekomendacji bazujący na sesjach rekomendacji 53
5.1.1. Podstawowe pojęcia grafów 53
5.1.2. Model matematyczny danych 55
5.1.3. Kroki algorytmu 59
5.1.4. Oszacowanie złożoności obliczeniowej 63
6. Implementacja algorytmów 66
6.1. Badanie kroków algorytmu rekomendacji bazującego na sesjach rekomendacji 66
6.2. Implementacja algorytmu rekomendacji bazującego na sesjach rekomendacji 72
6.2.1. Implementacja struktur grafowych 73
6.2.2. Budowa grafu G 75
6.2.3. Implementacja kroków algorytmu 77
6.2.4. Oszacowanie złożoności obliczeniowej implementacji 79
6.3. Implementacja algorytmu bazującego na regułach asocjacji 81
6.3.1. Ekstrakcja danych transakcyjnych 81
6.3.2. Implementacja kroków algorytmu odkrywania reguł asocjacji 83
6.3.3. Implementacja kroków algorytmu rekomendacji 84
7. Badanie algorytmu 86
7.1. Wprowadzenie do badań 86
7.1.1. Kryteria oceny algorytmów 87
7.1.2. Wartości bazowe charakterystyk 88
7.1.3. Środowisko badawcze 89
7.1.4. Plan badań 90
7.1.5. Związek badań z problemem badawczym 91
7.1.6. Forma prezentacji wyników 91
7.1.7. Czas realizacji badań 92
7.2. Wyniki badań 93
7.2.1. System rekomendacji zbudowany na bazie algorytmu losowych rekomendacji 93
7.2.2. System rekomendacji zbudowany na bazie algorytmu ARS w oparciu o zachowania użytkowników 96

7.2.3. System rekomendacji zbudowany na bazie algorytmu ARS w oparciu o atrybuty obiektów 102
7.2.4. System rekomendacji zbudowany na bazie algorytmu ARS w oparciu łącznie o zachowania użytkowników i atrybuty obiektów 107
7.2.5. System rekomendacji zbudowany na bazie algorytmu rekomendacji bazującego na regułach asocjacji 112
7.3. Podsumowanie wyników badań 119
7.3.1. Porównanie algorytmów 119
7.3.2. Statystyczna weryfikację hipotezy dla CTR 124
8. Kierunki rozwoju algorytmu 126
8.1. Modyfikacja 1 - dodanie wag do łuków 126
8.2. Modyfikacja 2 - konstruowanie ścieżek użytkownika 128
9. Zakończenie 130
10. Streszczenie 133
11. Abstract 134
12. Bibliografia 135

Spis skrótów i oznaczeń

AR - ang. Association Rules – reguły asocjacji
ARS - ang. Algorithm of the Recommendations Session - algorytm rekomendacji bazujący na sesjach rekomendacji
Bot - oznacza program symulujący zachowanie żywego użytkownika
CB - ang. Content Based Filtering – filtrowanie oparte o treści
CF - ang. Collaborative Filtering - filtrowanie kolaboratywne (wspólne)
CTR - ang. Click-Through Rate - współczynnik kliknięć
DBMS - ang. Database Management System - system zarządzania bazą danych
ERD - ang. Entity-Relationship Diagram - diagram związków encji
GIODO - Generalny Inspektor Ochrony Danych Osobowych
GRS - ang. Graph-based Recommender System - graf wykorzystywany w systemie rekomendacji
HIN - ang. Heterogeneous Information Network - heterogeniczna sieć informacyjna
HTTP - ang. Hypertext Transfer Protocol – protokół przesyłania dokumentów hipertekstowych
IP - ang. IP address – liczbowy identyfikator elementu sieci komputerowej bazującej na protokole TCP/IP
LHS - ang. „left-hand-side”, antecedent - określenie poprzednikiem w ramach reguły asocjacyjnej
MC - ang. Markov Chains - łańcuchy Markowa
MAE - ang. Mean Absolute Error - średni błąd bezwzględny
MBA - ang. Market Basket Analysis - analiza koszyka zakupów
PGRec - ang. Preference Graph based Recommendation - graf preferencji wykorzystywany w systemie rekomendacji
RHS - ang. „right-side-side”, consequent – określenie następnikiem w ramach reguły asocjacyjnej
RM - ang. Rating Matrix - macierz ocen
RMSE - ang. Root Mean Squared Error - średni błąd kwadratowy
RODO - ogólne rozporządzenie o ochronie danych osobowych
RS - ang. Recommendation System - system rekomendacji
SR - ang. Sequential Rules - reguły sekwencyjne
SBR - ang. Session-based Recommendation - rekomendacja bazująca na sesji
SZBD - system zarządzania bazą danych
TPG - ang. Tripartite Preference Graph - graf preferencji wykorzystywany w systemie rekomendacji
URL - ang. Uniform Resource Locator – ujednolicony format adresowania zasobów, stosowany w Internecie i w sieciach lokalnych
VOD - ang. Video on Demand – wideo na życzenie

1. Wstęp

Zagadnieniem badanym w rozprawie jest autorski algorytm rekomendacji bazujący na sesjach rekomendacji tworzonych na podstawie zachowań użytkowników oraz statycznych parametrów (atrybutów) obiektów w systemie e-Commerce.

Ogromna wielkość danych w sieci Internet stanowi utrudnienie związane z wyszukiwaniem informacji, zarówno dla użytkowników, jak i organizacji. W celu rozwiania tego problemu, badacze zaproponowali między innymi systemy rekomendacji (RS, ang. Recommendation System) (Singh i in. 2022). Wyniki ich działań są tak powszechne, że często nie zauważa się rozwiązań teleinformatycznych, które je generują. Systemy rekomendacji stanowią część praktycznie każdego nowoczesnego urządzenia i aplikacji. Potrzeba, aby systemy rekomendacji pomagały użytkownikom w kierowaniu ich do pożądanych informacji, rośnie wraz z rozwojem treści internetowych, a w zakresie aspektów technicznych i algorytmów rekomendacji dokonuje się ciągły postęp i rozwój (Neal 2011).

Zadaniem systemu rekomendacji jest często pomaganie konsumentom w poznawaniu nowych i atrakcyjnych towarów spośród wielu dostępnych opcji. Systemy tego typu dążą do wyeliminowania zbędnych lub niepożądanych informacji ze zbiorów danych. Ich funkcją jest dostarczanie istotnych treści i zmniejszenie przeciążenia informacyjnego (Nandini i in. 2018).

Współczesne systemy rekomendacji bazują na algorytmach, które w swej idei badają podobieństwo pomiędzy użytkownikami (klientami) lub obiektami (przedmiotami). Drugą wykorzystywaną ideą jest bazowanie na ocenach nadawanych obiektom przez użytkowników, a następnie badanie ich podobieństw (Bobadilla i in. 2013). Podstawami skuteczności funkcjonowania tych systemów są:

- mechanizmy gromadzenia informacji o użytkownikach takich jak wiek, płeć, lokalizacja, zainteresowania;
- mechanizmy oceniania obiektów i zbiory informacji o nadanych ocenach.

Aby te postulaty były spełnione, po pierwsze system wykorzystujący algorytm rekomendacji musi dawać możliwość gromadzenia informacji o użytkownikach, jak również użytkownicy muszą być skłonni, aby je udostępniać (Wójcik 2018). Aktualne zagrożenia cyberprzestrzeni i świadomość użytkowników powodują, że bardzo

niechętnie udostępniają oni informacje na swój temat (Interactive Advertising Bureau 2017). Po drugie, użytkownicy muszą chcieć oceniać obiekty. Aktualnie w systemach e-Commerce jest trend jednorazowych interakcji zakupowych połączony z minimalizacją działań wymaganych od użytkownika (Stolecka-Makowska 2016). Nie przywiązują się oni do systemów e-Commerce i bardzo niechętnie dzielą się swoimi opiniami (ocenami) na temat interesujących ich produktów (Gemius i Izba Gospodarki Elektronicznej 2020). Po trzecie, systemy rekomendacji bazujące na ww. założeniach są bardzo podatne na problem zwany „zimnym startem", czyli pracę w początkowym okresie bez wystarczających danych (Burke 2007) i „rzadkość danych" dla nowych obiektów (Sarwar i in. 2000).

Ponadto, aby algorytmy spełniały swoje funkcje rekomendacji w systemach e- Commerce, muszą zwracać wyniki w bardzo krótkim czasie, liczonym w ułamkach sekund. Jeśli bazują one na porównywaniu dużych zbiorów danych o użytkownikach lub produktach determinuje to konieczność wykorzystania środowisk serwerowych o dużych mocach obliczeniowych. Takie środowiska są drogie i dostępne głównie dla dużych organizacji (korporacji).

Liczną grupę algorytmów rekomendacji stanowią rozwiązania bazujące na teorii grafów i sieci. Ze swej natury grafy i sieci oddają bardzo dobrze model powiązań pomiędzy obiektami i użytkownikami (Jang i in. 2006). Zagadnienia związane z tym obszarem matematyki są wykorzystywane współcześnie przy budowie systemów rekomendacji w wielu nowych rozwiązaniach e-Commerce (Ben Fraj 2018).

Alternatywą do systemów opartych na podobieństwie obiektów lub podobieństwie użytkowników są systemy wykorzystujące reguły asocjacji powstałe w wyniku metod analizy asocjacji zwanej również analizą koszykową (MBA, ang. market basket analysis). Nie wymagają gromadzenia informacji o atrybutach użytkowników oraz wykonywania operacji badania podobieństwa obiektów w momencie dokonywania rekomendacji. Analiza asocjacji bazuje na informacjach dotyczących zachowań użytkowników (Raeder i Chawla 2011).

Zbliżony w swojej idei do tej klasy metod jest proponowany w pracy autorski algorytm rekomendacji bazujący na sesjach rekomendacji zwany algorytmem ARS (ang. Algorithm of the Recommendations Session). Algorytm ten opiera się na teorii grafów i sieci. W zasadniczej części pracy są opisane założenia funkcjonowania

i implementacji przedmiotowego algorytmu. Ponadto, przedstawiony jest proces wyznaczania wartości charakterystyk systemu rekomendacji bazującego na zaimplementowanym algorytmie w funkcjonującym systemie e-Commerce oraz wyniki porównania jego parametrów względem konkurencyjnego rozwiązania rekomendacyjnego. W ramach pracy dokonano również oszacowania parametrów wydajnościowych systemów w długim okresie jego funkcjonowania na bazie wyznaczonych trendów oraz zaproponowano kierunki rozwoju i modyfikacji algorytmu, które mogą stanowić cel przyszłych badań.

Aktualnie dla małych i średnich rozwiązań e-Commerce występuje potrzeba wykorzystania systemów rekomendacji prostych w implementacji, skalowalnych, efektywnych i obsługujących duże wielkości danych bez znacznego obciążania środowiska serwerowego. Odbiorcami takich algorytmów są głównie sklepy internetowe, ale również serwisy informacyjne (rekomendowanie artykułów prasowych), wyszukiwarki internetowe (rekomendowanie słów kluczowych), portale korporacyjne (rekomendowanie dokumentów) lub serwisy związane z pracą (rekomendowanie ofert pracy). We wszystkich tych miejscach potencjalnie algorytm ARS może znaleźć swoje zastosowanie (Malinowski i Krysiński 2020).

W rozdziale 2 przedstawiono opis i definicję obszaru e-Commerce oraz związane z nim pojęcia istotne dla dalszych rozdziałów pracy takie jak zachowania użytkowników, sesje i atrybuty obiektów.

Rozdział 3 zawiera definicję, w oparciu o literaturę z obszaru rekomendacji ze szczególnym uwzględnieniem przeglądu znanych technik i problemów rekomendacji. Ponadto został zdefiniowany problem badawczy. Szczególny wysiłek został położony na przedstawienie, w oparciu o istniejące opracowania, algorytmów rekomendacji bazujących na grafach i sieciach oraz przegląd istotnych miar oceny systemów rekomendacji.

W rozdziale 4 została przedstawiona idea analizy asocjacji oraz bazującego na niej algorytmu odkrywania silnych reguł asocjacji, a następnie opierającego się na nich algorytmu rekomendacji. Rozwiązanie to zostało wybrane jako konkurencyjne do algorytmu ARS. Wybór ten wynikał z faktu, że oba algorytmy bazują w swej idei na zachowaniach użytkowników.

Szczegółowo przedmiotowy algorytm ARS został przedstawiony w rozdziale 5. Reprezentacja ta opiera się na pojęciu grafów i sieci, na bazie których został skonstruowany model matematyczny stanowiący podstawę do opisu działania autorskiego algorytmu rekomendacji.

Praktyczne zastosowanie algorytmu ARS zostało zawarte w rozdziale 6, w którym to przedstawiono implementację rozwiązania w funkcjonującym systemie e- Commerce w oparciu o popularne rozwiązania bazodanowe i programistyczne.

W rozdziale 7 zostały zawarte informacje dotyczące badań algorytmu rekomendacji ARS i konkurencyjnego rozwiązania opartego o silne reguły asocjacji. Badania te bazowały na eksperymentach prowadzonych w funkcjonującym w czasie rzeczywistym systemie e-Commerce opisanym w rozdziale 6.

Na zakończenie pracy w rozdziale 8 zostały przedstawione możliwe drogi rozwoju algorytmu ARS i kierunki badań nad nim.

2. Obszar e-Commerce

Idea e-Commerce zwanego również handlem elektronicznym, odnosi się do szerokiego zakresu działalności gospodarczej online w zakresie produktów i usług. Jej główną cechą jest to, że uczestnicy transakcji biznesowych wchodzą w interakcje drogą elektroniczną. Obszar e-Commerce jest zwykle kojarzony z kupowaniem i sprzedawaniem przez Internet lub realizacją transakcji obejmujących przeniesienie własności, lub praw do korzystania z towarów, jak również usług za pośrednictwem sieci komputerowej (Bartczak 2016).

Chociaż ta definicja jest popularna, nie jest wystarczająco wyczerpująca, aby uchwycić aktualnie pojawiające się działalności biznesowe bazujące na sieciach komputerowych, zarówno tych globalnych jak Internet, ale również lokalnych jak intranet. Bardziej kompletna definicja brzmi: e-Commerce to wykorzystanie komunikacji elektronicznej i technologii cyfrowego przetwarzania informacji w transakcjach biznesowych w celu tworzenia, przekształcania i redefiniowania relacji w zakresie generowania wartości między organizacjami oraz między organizacjami a osobami fizycznymi (Gupta 2014).

Transakcje biznesowe mogą dotyczyć zarówno dóbr fizycznych (takich jak książki, ubrania, sprzęt AGD, produkty spożywcze), dóbr informacyjnych (artykuły prasowe, licencje, e-Book'i, kody dostępu, oferty pracy) oraz usług (wynajęcie samochodu, remont mieszkania, badanie lekarskie) (Malinowski i Sokólski 2001). W ramach wyżej zdefiniowanego obszaru e-Commerce funkcjonuje wiele aplikacji (systemów informatycznych), których celem jest realizacja i usprawnienie ww. transakcji. Między innymi występują wśród nich systemy rekomendacji.

Idea e-Commerce wiąże się ze społeczeństwem informacyjnym, które to zostało ukonstytuowane przez powszechny dostęp do narzędzi teleinformatycznych (komputerów, laptopów, smartfonów, sieci komputerowych), jak również umiejętność ich wykorzystania przez członków społeczeństwa determinując ich pozycję społeczną (Malinowski i Krysiński 2021).

2.1. Zachowania użytkowników

Z punktu widzenia systemów rekomendacji w rozwiązaniach e-Commerce wyróżnia się cztery główne zachowania użytkowników zwane krokami zakupowymi (Lee i in. 2000):

- wyświetlenie hiperłącza - wyświetlenie linku do strony WWW z opisem produktu;
- kliknięcie (ang. click-through) - kliknięcie w hiperłącze i wyświetlenie strony WWW z opisem produktu;
- dodanie do koszyka (ang. basket insertion) - umieszczenie produktu w koszyku elektronicznym;
- zakup (ang. purchase) - zakup produktu (zakończenie transakcji zakupowej).

Zachowania użytkowników identyfikowane są poprzez ewidencję wystąpień adresów URL w strumieniach kliknięć. Identyfikowane są te adresy URL, które związane są z ww. zachowaniami oraz są one mapowane na obiekty (produkty, usługi) z bazy danych systemu e-Commerce. Strumienie kliknięć to ścieżki, którymi użytkownicy przechodzą przez strony WWW.

Analiza strumieni kliknięć pokazuje, w jaki sposób użytkownicy poruszają się po stronach i jak z nich korzystają. Jej wyniki zawierają informacje przydatne do zrozumienia skuteczności działań marketingowych i merchandisingowych, czyli odpowiedzi między innymi na pytania: jak klienci trafiają do sklepu, jakie produkty oglądają i jakie kupują (Lee i in. 2001).

Opierając się na założeniu, że wszyscy użytkownicy systemu e-Commerce kupują produkty jedynie zgodnie z ww. krokami zakupowymi (zachowaniami), można sklasyfikować wszystkie produkty do grup, takich jak produkty zakupione, produkty dodane do koszyka, produkty, na które kliknięto, oraz pozostałe produkty. Na potrzeby opracowania nie identyfikuje się produktów związanych z krokiem „wrażenie produktu” oraz dodaje się nowe zachowanie: „wyróżnienie”, które polega na dodaniu produktu do listy życzeń użytkownika. Klasyfikacja ta zapewnia relację pomiędzy różnymi grupami, np. produkty zakupione to produkty umieszczone w koszyku, a produkty umieszczone w koszyku to produkty, na które kliknięto (nie można umieścić w koszyku i następnie kupić produktu bez uprzedniego kliknięcia w niego). Na podstawie tej relacji

można uzyskać kolejność preferencji pomiędzy produktami w następujący sposób: {produkty nigdy nieklikane} a {produkty tylko klikane} a {produkty tylko umieszczane w koszyku} a {zakupione produkty} (Cho i in. 2002). W związku z tym może być zasadne przypisanie wyższej wagi wystąpieniom produktów zakupionych niż produktom tylko umieszczonym w koszyku. Analogicznie, wyższą wagę można przypisać produktom tylko dodanym do koszyka niż produktom tylko kliknięty, i tak dalej. Może mieć to znaczenie dla rozwoju algorytmu będącego przedmiotem pracy co zostało przedstawione w rozdziale 8.

W związku z powyższym można zapisać zbiór typów zachowań użytkowników oznaczony przez B w następujący sposób:

$$B = \{b_1, b_2 , b_3, b_4\} \qquad (2.1)$$

gdzie:

b_1 - 'kliknięcie'

b_2 - 'dodanie'

b_3 - 'zakup'

b_4 - 'wyróżnienie'

Zakładając, że O to zbiór wszystkich produktów (obiektów, usług) w systemie e- Commerce można wyodrębnić takie podzbiory produktów wynikające z zachowań użytkowników:

O_c - produkty „kliknięte”

O_d - produkty dodane do koszyka

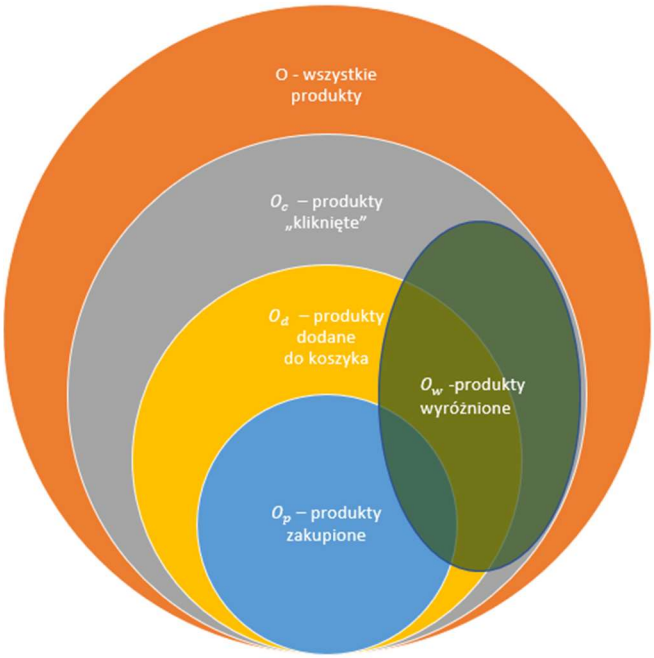
O_p - produkty zakupione

O_w - produkty wyróżnienie

gdzie:

$O_p \subseteq O_d \subseteq O_c \subseteq O$

$O_w \subseteq O_c$



Rysunek 2.1. Podzbiory produktów wynikające z zachowań użytkowników. Źródło: opracowanie własne.

2.2. Pierwotne pojęcie sesji

Strumień kliknięć (żądań, zapytań, trafień, odpowiedzi) serwisu WWW to ogólny termin opisujący ścieżki odwiedzających (użytkowników) w jednej lub kilku witrynach internetowych. Serię żądań o strony WWW odwiedzającego użytkownika podczas jednej wizyty w serwisie WWW określa się mianem sesji. Strumień kliknięć to między innymi zbiór sesji. Dane o strumieniu kliknięć, a w konsekwencji o sesjach, można uzyskać z nieprzetworzonych żądań o stronę i związanych z tym informacji takich jak: znacznik czasu, adres IP (ang. IP address), adres URL (ang. Uniform Resource Locator), status, liczba przesłanych bajtów, odsyłacz, agent użytkownika, a czasami dane cookie. Dane te są rejestrowane w plikach dziennika (logach) serwera WWW.

Analiza strumieni kliknięć (sesji) pokazuje, w jaki sposób odwiedzający poruszają się po witrynie WWW systemu e-Commerce i jak z niej korzystają (Lee i in. 2001). Pojęcie żądania jest technicznie związane z protokołem przesyłania dokumentów hipertekstowych HTTP (ang. Hypertext Transfer Protocol) należącym do rodziny protokołów sieciowych TCP/IP.

Sesje ze swej natury, charakteryzują krótkie odcinki czasu, w których użytkownik wchodzi w interakcję z witryną WWW (Han i in. 2012), gdzie jako witryna WWW jest

rozumiany zbiór stron WWW dostępnych pod jednym adresem elektronicznym (URL). W wielu systemach internetowych w przypadku braku interakcji użytkownika przez czas rzędu 4-20 minut sesja jest automatycznie przerywana. Po przerwaniu może być rozpoczęta nowa sesja, która może wymagać ponownego logowania użytkownika do systemu w celu jego identyfikacji (np.: systemy bankowe) lub być anonimowa. Użytkownik nie jest wówczas jawnie personalizowany. Z sesją przeważnie jest związany jej identyfikator, przy pomocy którego można odróżnić dwie sesje. Ponadto z pojedynczym żądaniem w ramach sesji może być związane jedno z zachowań użytkownika ze zbioru B zdefiniowanym w (2.1) oraz obiekt o ze zbioru obiektów O którego dotyczy żądanie.

W związku z powyższym można formalnie zapisać, że sesja Q jest uporządkowanym chronologicznie ciągiem żądań stron q (trafień, kliknięć), gdzie porządek determinuje czas pojawienia się żądań.

$$Q = [q_1, q_2, \dots, q_n] \quad (2.2)$$

gdzie:

q_i – pojedyncze żądanie strony

n – numer ostatniego żądania strony

Ponadto:

$$S = [Q_1, Q_2, \dots, Q_m] \quad (2.3)$$

gdzie:

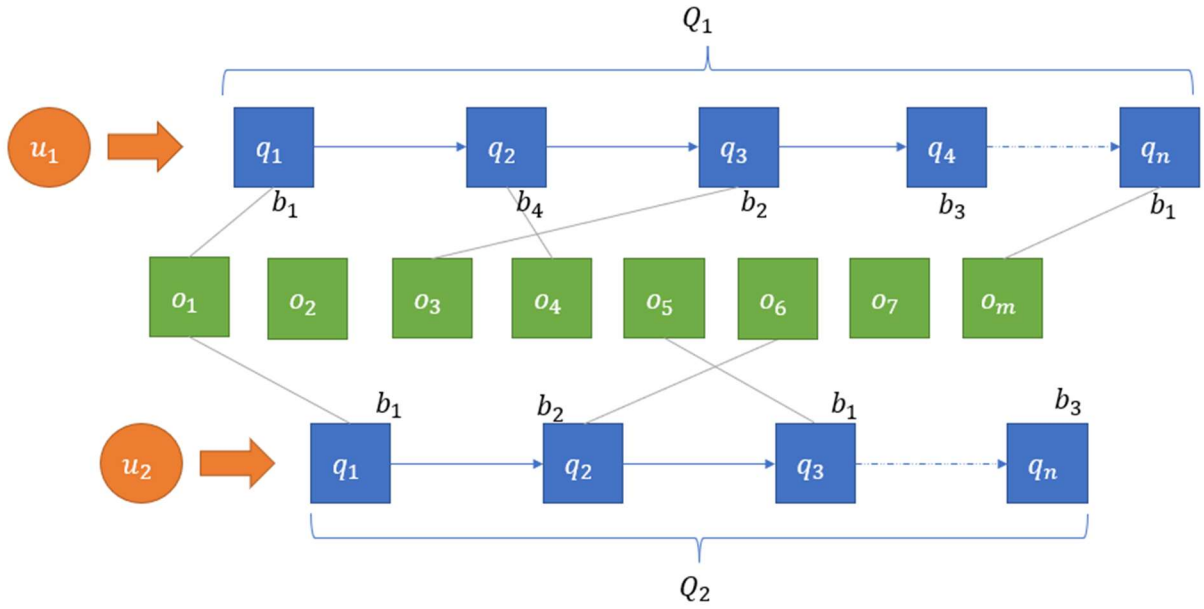
S – uporządkowany chronologicznie względem czasu rozpoczęcia ciąg sesji

m – numer ostatniej rozpoczętej sesji

Na ogół problem rekomendacji na podstawie sesji zdefiniowanych w postaci (2.2) sprowadza się do przewidzenia następnego kroku użytkownika, czyli predykcji żądania kolejnej strony q_{n+1} (He i in. 2009). W przypadku rozwiązań e-Commerce strony przede wszystkim opisują produkty co można sprowadzić do stwierdzenia, że systemy bazujące sesjach są wykorzystywane do rekomendacji produktów do potencjalnych zakupów.

Jak wyżej wspomniano, żądania te mają często związek z zachowaniami użytkowników ze zbioru B zdefiniowanymi w (2.1), czyli 'kliknięcie', 'dodanie', ' zakup'

lub 'wyróżnienie' i związane są z obiektami ze zbioru wszystkich produktów (obiektów, usług) w systemie e-Commerce. Bazowanie rekomendacji na sesjach jest utożsamiane z takimi pojęciami jak „Sequence-Aware Recommendation" (Quadrana i in. 2018), „Session-Based Recommendation" lub „Sequence Learning" (Ludewig i Jannach 2018) .



Rysunek 2.2. Model sesji Q_1 i Q_2 gdzie: $u \in U$ (zbiór użytkowników), $o \in O$ (zbiór obiektów) oraz $b \in B$ (zbiór zachowań taki, że: $b_1 =' kliknięcie'$ $b_2 =' dodanie'$ $b_3 =' zakup'$ $b_4 =' wyróżneinie'$).
Źródło: opracowanie własne.

2.3. Atrybuty obiektów w systemie e-Commerce

Aktualnie w dobie zakupów realizowanych z wykorzystaniem sieci komputerowej Internet, klienci podejmują głównie decyzje o wyborze obiektu do zakupu na podstawie atrybutów prezentowanych w systemach e-Commerce (Huang i in. 2020). Dlatego też sprzedawcy są zainteresowani dobrą identyfikacją istotnych dla potencjalnych klientów atrybutów (Fuchs i in. 2010). Zidentyfikowane istotne atrybuty produktu mogą być następnie wykorzystane do promocji produktu lub przez system rekomendacji.

Atrybuty obiektu (produktu, usługi) to zestaw cech, które określają konkretny obiekt. Obejmują one rozmiar, kolor, typ produktu, cenę i inne cechy (Sriram 2018). Z punktu widzenia e-Commerce rozważane są te informacje, które wpływają na decyzje zakupowe klientów lub są związane z działaniami użytkowników w systemie.

Atrybuty definiuje się jako materialne, niematerialne, dynamiczne oraz złożone (Bartram 2020):

- materialne to takie cechy jak rozmiar, kolor, zapach, wygląd lub waga;
- niematerialne odnoszą się do takich rzeczy jak cena, jakość i estetyka;
- dynamiczne to między innymi częstość zakupu, częstość oglądania, polecane lub aktualna promocja;
- złożone to struktury wielowartościowe na przykład przypisanie do kilku kategorii.

Można opisać zbiór atrybutów i na jego podstawie przedstawić każdy obiekt należący do zbioru obiektów $o \in O$ wektorem wartości atrybutów i zbiorami wartości atrybutów (w przypadku atrybutów złożonych) (Yu i in. 2014):

$$A = \{A_1, A_2, \dots, A_n\} \qquad (2.4)$$

$$o =< a_1, a_2, a_3 \dots, a_n : a_i \in A_i > \qquad (2.5)$$

gdzie:

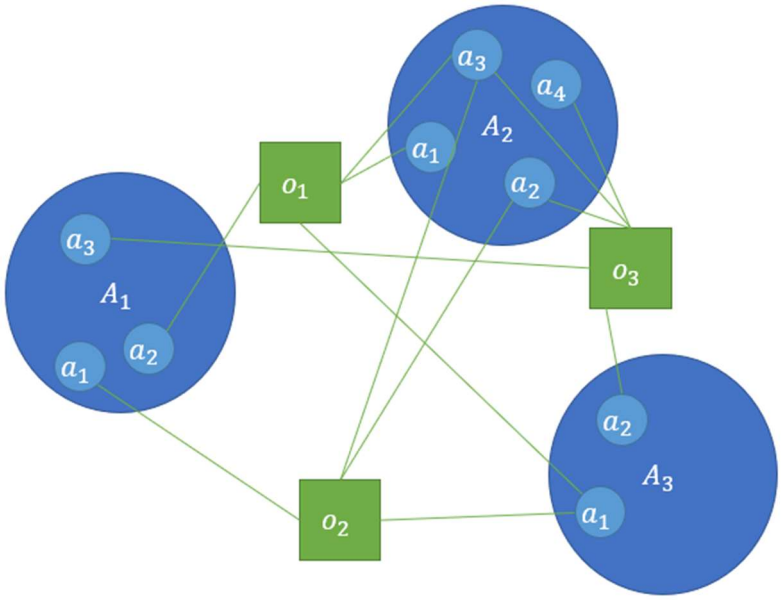
A – zbiór atrybutów

A_i – zbiór wartości atrybutu i

a_i – wartość atrybutu i

n – liczba atrybutów

oraz niektóre atrybuty mogą być złożone, np. $a_2 = \{a_{2,}^1 a_2^2\}$.



Rysunek 2.3. Model obiektów o_1, o_1 i o_3 przedstawiony w postaci przypisanych wartości atrybutów a_x ze zbiorów wartości atrybutów A_1, A_1 i A_3.
Źródło: opracowanie własne.

Model obiektów w systemach e-Commerce bazujący na wektorze atrybutów znajduje szerokie zastosowanie w obszarze rekomendacji. Również i implementacja algorytmu ARS w systemie e-Commerce wykorzystuje elementy wyżej opisanego modelu związanego z wybranymi zbiorami atrybutów.

2.4. Baza danych transakcji (koszyk zakupów)

Baza danych transakcji (koszyk zakupów) stanowi podstawę do analizy asocjacji zwanej również analizą koszykową. Analiza asocjacji jest techniką eksploracji danych, która koncentruje się na "wzorcach zakupowych" klientów (Larose 2006). Celem tej analizy jest identyfikacja interesujących wzorców poprzez uwzględnienie produktów kupowanych przez klientów na podstawie transakcji zakupowych. Aby zidentyfikować wzorce, analizuje się produkty kupowane przez klientów podczas pojedynczych akcji zakupów zwanych transakcjami. W klasycznych marketach odpowiada to akcji płatności w kasie za produkty zgromadzone w koszyku (Rao i in. 2021).

Analiza asocjacji jest procesem, który poszukuje związków pomiędzy obiektami, które "pasują do siebie" w kontekście biznesowym. W rzeczywistości analiza ta wykracza poza scenariusz supermarketu, od którego pochodzi jej nazwa. Jest to

analiza dowolnego zbioru obiektów (przedmiotów, usług) w celu zidentyfikowania podobieństw, które można z punktu widzenia biznesu wykorzystać. Niektóre przykłady zastosowania analizy asocjacji obejmują (Loshin 2013):

- Lokowanie produktu. Zidentyfikowanie produktów, które mogą być często kupowane razem i umieszczenie ich w pobliżu (np. na półkach sklepu, w papierowym katalogu lub na stronie WWW), aby zachęcić nabywcę do zakupu obu produktów.
- Fizyczne rozmieszczenie produktów na półkach. Alternatywnym zastosowaniem fizycznego rozmieszczenia produktów w sklepie jest oddzielenie produktów, które są często kupowane w tym samym czasie, aby zachęcić klientów do wędrowania po sklepie w celu znalezienia tego, czego szukają i potencjalnie zwiększyć prawdopodobieństwo dodatkowych zakupów pod wpływem impulsu.
- Sprzedaż dodatkowych produktów (ang. up-selling) i sprzedaż krzyżowa (ang. cross-selling) oraz możliwość łączenia produktów w pakiety. Firmy mogą wykorzystać fakt grupowania wielu produktów jako wskazówkę, że klienci mogą być predysponowani do kupowania tych produktów w tym samym czasie. Umożliwia to prezentację produktów w celu sprzedaży krzyżowej lub może sugerować, że klienci będą skłonni kupić więcej produktów, jeśli pewne produkty będą w pakiecie.
- Zatrzymanie klienta. Kiedy klienci kontaktują się z firmą w celu zerwania współpracy, przedstawiciel firmy może wykorzystać wyniki analizy asocjacji, aby użyć odpowiedniej zachęty, którą warto zaoferować w celu utrzymania klienta.

Aktualnie do budowy baz danych transakcji wykorzystuje się dwie główne metody eksploracji danych. Jedna z nich opiera się na ekstrakcji danych o zakupach z opisanych wcześniej sesji bazujących na strumieniu kliknięć (Ludewig i Jannach 2018). W tym przypadku ze zbioru sesji wyodrębnia się, te które mają żądania q powiązane z zachowaniem użytkownika typu $'zakup' \in B$ zdefiniowanym w (2.1). Następnie na bazie tych sesji buduje się podzbiory z żądaniami, które związane są z zachowaniem użytkownika $'dodanie' \in B$. Dzięki właściwości sesji, że każda dotyczy tylko jednego użytkownika podczas pojedynczych odwiedzin systemu e-Commerce, tak otrzymane podzbiory można traktować jako transakcje zakupowe.

Jeśli przyjmiemy, że pojedyncze żądanie w sesji ma postać:

$$q =< b, o > \qquad (2.6)$$

gdzie:

$b \in B$ - zachowanie użytkownika w systemie e-Commerce

$o \in O$ – obiekt (produkt, usługa) z bazy danych systemu e-Commerce związany z zachowaniem b

Wówczas na podstawie ciągu sesji S zdefiniowanego w (2.3) można zbudować podzbiór S', taki, że:

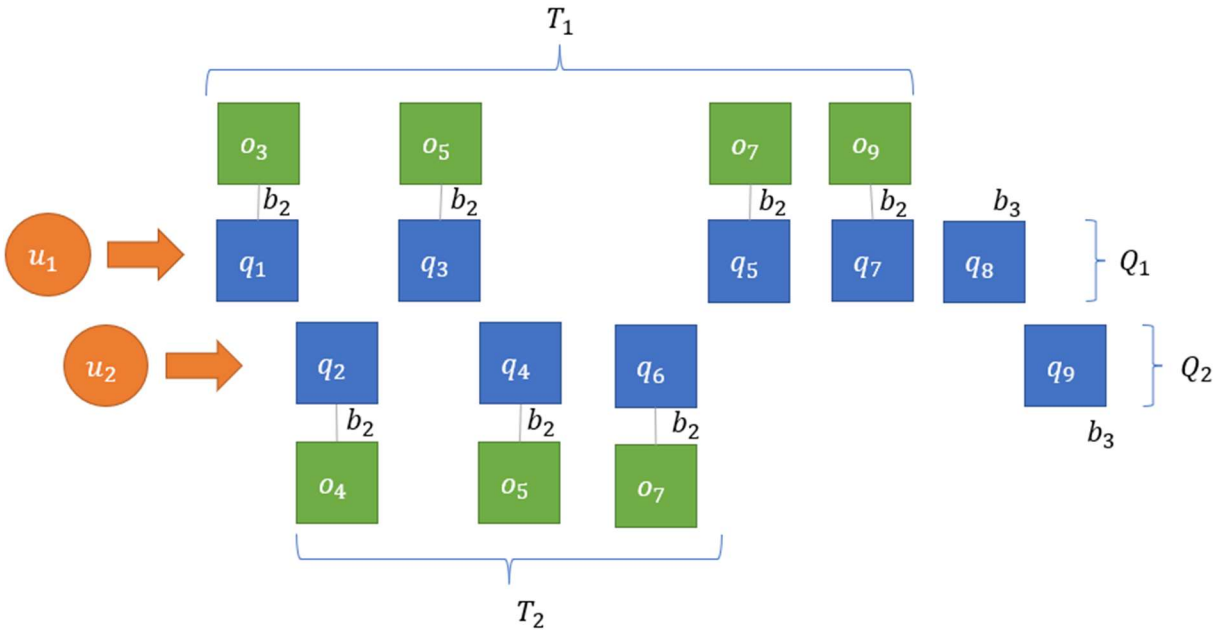
$$S' = \{Q = (q_1, .., q_{|Q|}): \exists i \epsilon \{1, .., |Q|: q_i = (b_i, o_i) \wedge b_i =' zakup'\} \qquad (2.7)$$

Następnie dla każdej sesji $Q \in S'$ zdefiniowanej w (2.2) można zbudować podzbiór T złożony z obiektów takich, że:

$$T(Q) = \{o \epsilon O: \exists i \epsilon \{1, .., |Q|: q_i = (b_i, o_i) \wedge b_i =' dodanie' \wedge o_i = o\} \qquad (2.8)$$

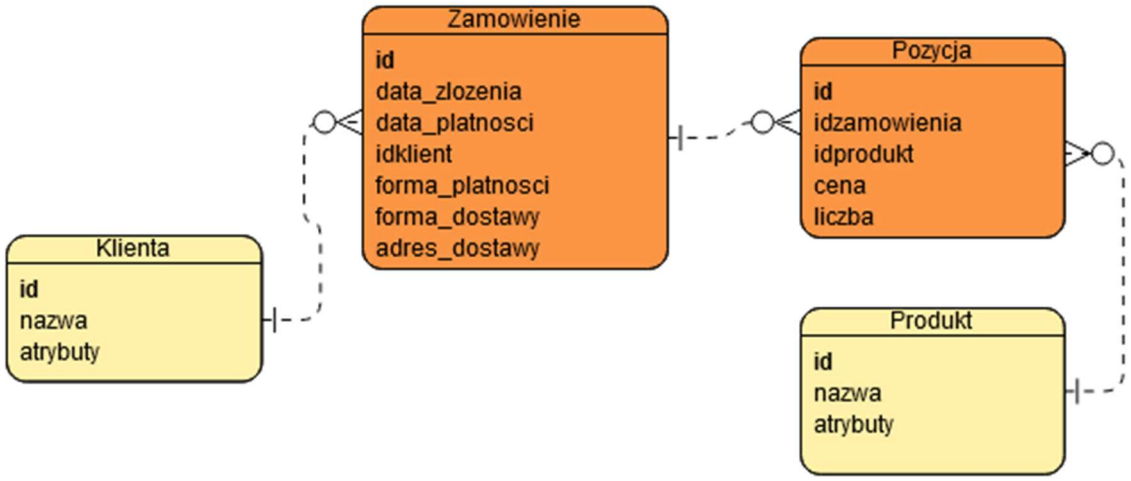
$T(Q) \subseteq O$ zawiera obiekty zakupione podczas jednej sesji (transakcji). Na tej podstawie zbiór transakcji zwany bazą danych transakcji lub koszykiem zakupów przyjmuje postać:

$$D = \{T_1, T_2, \dots, T_n\} \qquad (2.9)$$



Rysunek 2.4. Model transakcji $T_1 \subseteq O$ i $T_2 \subseteq O$ zbudowany na bazie sesji Q_1 użytkownika u_1 i Q_2 użytkownika u_1 zawierających żądania powiązane z zachowaniami $b =' dodanie'$ obiektu $o \in O$ oraz $b_3 =' zakup'$.
Źródło: opracowanie własne.

Drugą metodą jest bezpośrednie wykorzystanie jako bazy danych transakcji danych zawartych w systemie magazynowo-księgowym, w którym zapisywane są bieżące transakcje zakupowe klientów danego systemu e-Commerce. Z reguły w systemie takiego typu każda transakcja zapisywana jest jako pojedyncze zamówienie i pozycje tego zamówienia, gdzie pozycje przechowują informacje o zakupionych produktach (usługach), ich liczbie oraz cenie. Natomiast zamówienie przechowuje informacje o dacie złożenia, dacie płatności, miejscu i formie dostawy oraz kliencie (Pondel i Korczak 2017). Schemat ten przedstawia poniższy diagram związków encji (ERD, ang. Entity-Relationship Diagram).



Rysunek 2.5. Diagram ERD tabel „Zamówienie" i „Pozycja" w przykładowej bazie danych systemu e-Commerce.
Źródło: opracowanie własne.

Szczegółowo wykorzystanie idei analizy asocjacji w kontekście systemów rekomendacji zostanie opisane w rozdziale 4 pracy.

2.5. Pojęcia związane ze strumieniem kliknięć

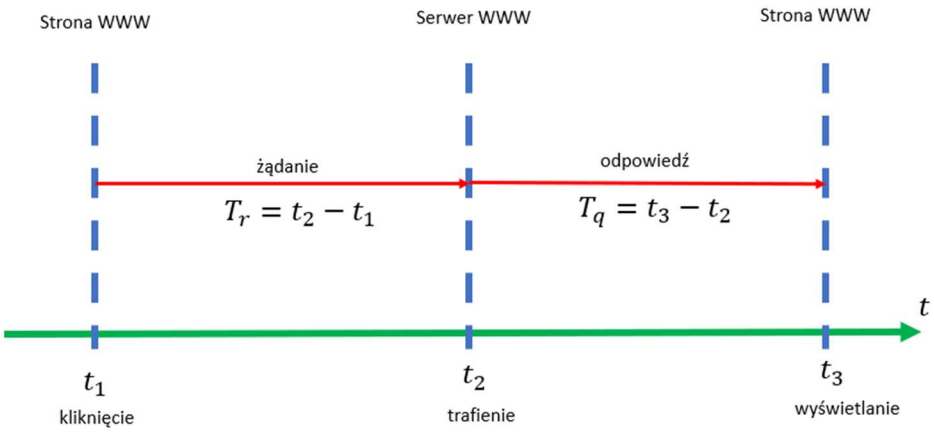
W tym momencie należy wskazać, na techniczne różnice pomiędzy pojęciami kliknięcie, żądanie, zapytanie, trafienie i odpowiedź serwera (serwisu, strony) WWW. Pojęcia te wiążą się z wymienionym wcześniej protokołem HTTP i rozważanym strumieniem kliknięć (Siyan i Parker 2002):

- kliknięcie – świadome i celowe zainicjowanie przez użytkownika przesłania żądania (zapytania) o zasób będący na serwerze WWW a związany (powiązany linkiem w postaci adresu elektronicznego URL) ze stroną WWW, na której przebywa użytkownik. Kliknięcie fizycznie realizowane jest poprzez naciśnięcie w przycisk (ang. button), link tekstowy lub plik graficzny umieszczony na stronie WWW;
- żądanie (zapytanie) (ang. HTTP request) – jest to przesłanie do serwera WWW, na którym jest umieszczona strona (serwis) WWW, żądania (zapytania) o zasób. Zasobem może być plik HTML, plik graficzny lub dokument Word;
- trafienie – jest to interpretacja informacji przekazanej przez serwer WWW o istnieniu zasobu, o który było żądanie. W przypadku nieskutecznego żądania wynikającego z niedostępności serwera WWW lub błędnego wskazania na zasób (błąd w linku) może nie dojść do trafienia;
- odpowiedź (ang. HTTP response) - po otrzymaniu żądania serwer WWW odpowiada (tj. wysyła dane zasobu), a przeglądarka wyświetla go w odpowiedniej formie w postaci strony WWW.

Co jest istotne z punktu widzenia rozważanego pojęcia strumienia kliknięć i sesji, to jest to, że wszystkie ww. pojęcia wiążą się z tym samym procesem, który można nazwać „pobraniem z serwera klikniętej strony WWW” i są jego elementami zachodzącymi sekwencyjnie. Dlatego też z punktu widzenia niniejszego opracowania można je używać zamiennie.

2.6. Czas odpowiedzi systemu e-Commerce

Na poniższym wykresie na osi czasu są zaznaczone poszczególne elementy występujące w procesie oraz momenty ich powstawania i czasy trwania. Bardzo ważnym czasem z punktu widzenia rozwiązań e-Commerce jest T_q zwany czasem odpowiedzi (ang. Response Time).



Rysunek 2.6. Przebieg procesu „pobrania z serwera klikniętej strony WWW”
Źródło: opracowanie własne.

Czas ten odgrywa w rozwiązaniach e-Commerce kluczową rolę. Jeśli jest zbyt długi, wówczas użytkownicy rezygnują z korzystania z danego rozwiązania i nie ma znaczenia cena i jakość oferowanych produktów lub usług (Jaiswal i Singh 2020). Oznacza to, że nie ma tak dużego wpływu na decyzję klienta efektywność oraz funkcjonalność wykorzystywanych w systemie e-Commerce algorytmów. Paradoksalnie lepsze dla użytkowników są rozwiązania oferujące gorsze wyniki, ale szybsze. Mówi się, że milisekundy warte są miliony (Glynn i in. 2020).

Szacuje się, że 43% użytkowników komputerów stacjonarnych rezygnuje, gdy ładowanie strony trwa dłużej niż 3 sekundy. Natomiast połowa użytkowników mobilnych rezygnuje, jeśli czas odpowiedzi wynosi ponad 5 sekund. Ponadto prawie 9 na 10 osób nigdy nie wraca na stronę WWW, jeśli jej działanie jest powolne (Loisel 2001). Podobne wartości ładowania strony podaje Google wskazując następujące czasy ładowanie strony i dzieli na trzy grupy (Google 2022):

- ≤2,5s – dobra wartość
- <2,5s i ≤4s – wymaga poprawy
- <4s – słaba wartość

Czas odpowiedzi można zapisać jako:

$$T_q = T_s + T_t + T_b \qquad (2.10)$$

gdzie:

T_s - czas odpowiedzi serwera, którego główną zmienną jest czas dostępu do zasobu

T_t - czas przesłania danych poprzez sieć od serwera WWW do przeglądarki WWW

T_b - czas przetworzenia danych przez przeglądarkę WWW celem wyświetlenia w postaci strony WWW

O ile wartości T_t i T_b w dobie ogólnie dostępnego szybkiego Internetu oraz bardzo wydajnych urządzeń końcowych użytkowników są względnie małe i co ważne stałe, o tyle kluczowym składnikiem jest T_s. Na tę składową wpływa głównie czas generowania danych w systemie e-Commerce, w ramach którego powstaje treść strony WWW przesyłanej w odpowiedzi serwera (Bogárdi-Mészöly i in. 2005). W dalszej części pracy będą prowadzone badania nad czasem jaki wiąże się z czasem generowania rekomendacji, który to bezpośrednio wpływa na generowanie danych w systemie e-Commerce, a w konsekwencji na czas odpowiedzi serwera WWW.

3. Obszar rekomendacji

Użytkownik, korzystając z jednego z narzędzi e-Commerce, takiego jak na przykład sklep internetowy lub tablica ogłoszeń, staje przed ogromnym zbiorem jednorodnych, często nie w pełni zrozumiałych dla niego obiektów. Dla jego świadomości mają one zbliżony wygląd oraz charakterystykę. Jak zatem rozwiązać problem wyboru z tego zbioru jednego poszukiwanego obiektu? W przypadku sklepów internetowych nie ma możliwości bezpośredniego poproszenia sprzedawcy o poradę. Można ewentualnie wysłać wiadomość elektroniczną z prośbą o pomoc i następnie oczekiwać na odpowiedź. Metoda ta, we współczesnym mobilnym świecie, nie jest zadowalająca dla wielu użytkowników ze względu na opóźnienie odpowiedzi (Malinowski i Krysiński 2020).

Na dużą liczbę danych gromadzonych w systemach informatycznych, w tym i systemach e-Commerce, wskazuje następujący przykład: w 2020r. każdego dnia tworzonych było ponad 2,5 tryliona bajtów danych, co daje 1,7 MB danych co sekundę na każdą osobę na Ziemi (Ahmad 2021). W związku z tym nieodzowna stała się pomoc narzędzi teleinformatycznych wspierających użytkowników w procesie poszukiwania i wyboru.

Do rozwiązywania problemu wyboru obiektów w systemach e-Commerce wykorzystywane są między innymi systemy rekomendacji bazujące na technikach rekomendacji.

3.1. Definicje i formy rekomendacji

Definicja systemu rekomendacji ewoluowała w ciągu ostatnich 14 lat. Pierwsza próba opisu systemu rekomendacji miała miejsce w 1997r. i została przestawiona w następujący sposób (Park 2010):

"W typowym systemie rekomendacji ludzie dostarczają rekomendacji jako danych wejściowych, które system następnie agreguje i kieruje do odpowiednich odbiorców. W niektórych przypadkach podstawowa transformacja polega na agregacji, w innych wartość systemu polega na jego zdolności do dobrego dopasowania osób rekomendujących do osób poszukujących rekomendacji." (Resnick i Varian 1997)

Należy zauważyć, że definicja ta kładzie nacisk na systemy rekomendacji jako wspierające współpracę między użytkownikami. Później badacze rozszerzyli tę definicję na systemy, które sugerują interesujące obiekty, niezależnie od tego, w jaki sposób te rekomendacje są tworzone. Na tej bazie powstała definicja systemu rekomendacji, która mówi, że jest to:

"Każdy system, który generuje zindywidualizowane rekomendacje jako wynik lub ma efekt kierowania użytkownika w spersonalizowany sposób do interesujących, lub użytecznych obiektów w dużej przestrzeni możliwych opcji." (Burke 2002)

Z czasem ta ogólna definicja została, na podstawie teorii mnogości i funkcji użyteczności, sformalizowana (Adomavicius i Tuzhilin 2005). Szczegółowo formuła ta zostanie przedstawiona w rozdziale 3.4.

Zasadniczo przyjmuje się, że systemy rekomendacji, służą do oszacowania preferencji użytkowników (klientów) dotyczących przedmiotów, usług lub innych obiektów, których jeszcze nie widzieli lub nie znają. Systemy rekomendacji często używają danych wejściowych, takich jak preferencje użytkownika, cechy (atrybuty) obiektu, historia przeszłych interakcji użytkowników z obiektami, dane czasowe i dane przestrzenne (Bhaskar i in. 2020).

Istotną cechą systemu rekomendacji jest jego zdolność do przewidywania preferencji i zainteresowań użytkownika poprzez analizę jego zachowania lub zachowania innych użytkowników w celu wygenerowania spersonalizowanych wyników.

Rekomendacje dostarczane przez systemy rekomendacji mają dwie różne formy:

- przewidywania ocen (ang. prediction) - w tym przypadku szacowana jest ocena użytkownika dla nowego przedmiotu. Przewidywanie bazuje na podstawie ocen rozpatrywanego obiektu dokonanych przez innych użytkowników. Dla przykładu przewidywaniem oceny jest prognoza odpowiadająca na pytanie „czy konkretnemu użytkownikowi spodoba się określony film?”. W serwisie Netflix prognoza tego typu bazuje na podstawie historycznego zachowania innych użytkowników tego serwisu;
- rankingu (ang. ranking) - szacowanie wyniku polega na utworzeniu listy rankingowej „Top-N” z N pozycjami dla konkretnego użytkownika. System bazujący

na tym schemacie może polecać „10 najlepszych książek do przeczytania, jeśli lubisz Harry'ego Pottera” (Cremonesi i in. 2010).

W rozwiązaniach e-Commerce preferowane są systemy rekomendacji dające wyniki w postaci „rankingu” (Top-N), a nie „przewidywanych ocen”, ponieważ firmy preferują wiedzę na temat oczekiwanych obiektów zamiast ich ocen (Steck 2013).

Obszar rekomendacji rozpatrywany jest w trzech głównych płaszczyznach (Liling 2019):

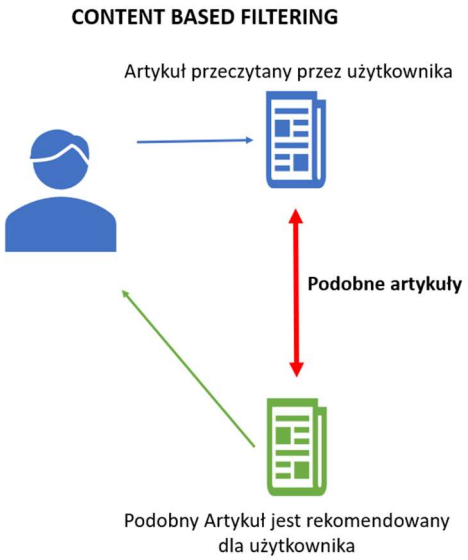
- technik rekomendacji – w kontekście rozwoju metod analitycznych problematyki rekomendacji;
- algorytmów rekomendacji – w kontekście opracowywania algorytmów opartych o techniki rekomendacji;
- systemów rekomendacji – w kontekście budowy rozwiązań informatycznych wspierających procesy rekomendacji wykorzystujących algorytmy rekomendacji.

3.2. Techniki rekomendacji

Aktualnie, w opracowaniach dotyczących obszaru rekomendacji wyróżnia się następujące główne techniki rekomendacji (Ricci i in. 2015):

3.2.1. Content Based Filtering

Algorytmy bazujące na tej technice są konstruowane na podstawie założenia, że obiekty o podobnych funkcjach otrzymują podobne oceny od tego samego użytkownika. W konsekwencji systemy tego typu polecają obiekty, które są podobne do przedmiotów, w stosunku do których użytkownik wykazywał zainteresowanie w przeszłości. Podobieństwo obiektów jest obliczane na podstawie określonych atrybutów i z wykorzystaniem miar podobieństw (Li i in. 2012).



Rysunek 3.1. Idea techniki Content Based Filtering
Źródło: opracowanie własne.

3.2.2. Collaborative Filtering

Idea tej techniki opiera się na założeniu, że jeśli dwóch użytkowników podejmowało w przeszłości podobne decyzje, to w przyszłości ich preferencje też będą zbieżne. Przykładem może być sklep z grami planszowymi, w którym dwaj użytkownicy wykazali zainteresowanie podobnymi grami, ale jeden z nich zainteresował się jeszcze inną grą. W takiej sytuacji mechanizm Collaborative Filtering zarekomenduje tę grę również drugiemu użytkownikowi, zakładając, że i jemu też przypadnie ona do gustu. W powyższej sytuacji system nie opiera swojego działania na atrybutach obiektów, a jedynie na zbieżnych cechach osób, dla których zainteresowania i preferencje zostały uznane za podobne (Su i Khoshgoftaar 2009).

Rysunek 3.2. Idea techniki Collaborative Filtering
Źródło: opracowanie własne.

W ramach tej techniki wyróżnia się podtypy:

Memory Based

Charakteryzuje się on tym, że do przeprowadzenia rekomendacji używana jest cała baza danych systemu. Zakłada się, że każdy użytkownik jest częścią pewnej grupy, dlatego używa się metod statystycznych do znalezienia tzw. „najbliższych sąsiadów", czyli grupy użytkowników o podobnych zainteresowaniach. W pierwszym kroku mechanizm rekomendacji tego typu wyszukuje grupę najbliższych sąsiadów poprzez obliczenie wagi, która określa podobieństwo lub korelację pomiędzy dwoma użytkownikami. Następnie obliczana jest prognoza określająca czy dany produkt może być polecony użytkownikowi na podstawie jego najbliższych sąsiadów. Ostatnim krokiem jest wybór N obiektów najbardziej pasujących do użytkownika i przedstawienie ich w postaci rekomendacji (Su i Khoshgoftaar 2009).

Model Based

Podstawą działania jest utworzenie modelu ocen użytkownika, który będzie w stanie przewidzieć jego ocenę dotyczącą obiektów. Podczas tworzenia modelu wykorzystywane są techniki uczenia maszynowego na podstawie danych treningowych (Sarwar i in. 2001). Jednym ze sposobów oszacowania rekomendacji obiektu dla danego użytkownika jest użycie klasyfikatora bayesowskiego, który jest

często stosowany w systemach rekomendacji. Stanowi on jedną z metod uczenia maszynowego określającą, do której z klas decyzyjnych należy przypisać nowy przypadek. W kontekście mechanizmów rekomendacji określane jest, jak bardzo wybrany obiekt jest odpowiedni dla danego użytkownika (Miyahara i Pazzani 2000).

3.2.3. Oparte o wiedzę

Systemy oparte na wiedzy rekomendują elementy na podstawie specyficznej wiedzy dziedzinowej o tym, jak pewne cechy obiektu spełniają potrzeby i preferencje użytkowników, a w efekcie jak dany obiekt jest użyteczny dla użytkownika. Systemy oparte na wiedzy zwykle działają lepiej niż inne na początku ich funkcjonowania, ale jeśli nie są wyposażone w komponenty uczące się, okazują się gorsze od konkurencyjnych metod, które wykorzystują wiedzę na temat zachowań użytkowników (Bouraga i in. 2014).

3.2.4. Bazujące na demografii

Ta technika rekomendacji bazuje na podstawie profilu demograficznego użytkownika. Zakłada się, że dla różnych grup demograficznych powinny być generowane różne rekomendacje, a natomiast dla członków tej samej grupy powinny występować te same rekomendacje. Na podstawie tego założenia w wielu serwisach internetowych stosuje się proste i skuteczne rozwiązania personalizacji oparte na danych demograficznych. Na przykład, użytkownicy są kierowani do określonych witryn na podstawie ich języka lub kraju. Sugestie mogą być też dostosowywane do wieku użytkownika (Bobadilla i in. 2013).

3.2.5. Bazujące na analizie asocjacji

Techniki eksploracji danych, takie jak wyszukiwanie reguł asocjacji odkrywanych w wyniku metod analizy asocjacji są stosowane do analizy zachowań zakupowych klientów w celu poznania ich zwyczajów zakupowych i opracowania strategii marketingowych (Han i in. 2012). Dane te stanowią również podstawę do budowy systemów rekomendacji. Ich bardzo dużą zaletą jest to, że informacje nie są pozyskiwane ze statycznych charakterystyk obiektów (atrybutów) oraz ocen

użytkowników, lecz bazują na ich rzeczywistych zachowaniach takich jak na przykład zakupy lub odwiedziny podstron produktów (Zhang 2007).

3.2.6. Bazujące na sesjach

Kolejna technika to technika bazująca na sesjach (SBR, ang. Session-based Recommendation). Systemy budowane na jej podstawie przewidują następne kliknięcia użytkowników na podstawie ich wcześniejszych zachowań podczas bieżącej sesji (Rong i in. 2022). Rozwiązanie problemu predykcji na podstawie tej techniki bazuje między innymi na:

Łańcuchach Markowa (MC, ang. Markov Chains)

Podejście to opiera się na oszacowaniu prawdopodobieństw przejść między stanami, które są reprezentowane poprzez żądania z sesji Q zdefiniowanej w (2.2). W podstawowym podejściu liczone jest jak często żądanie q_n następuje bezpośrednio po żądaniu q_m i na tej podstawie budowana jest macierz przejść pomiędzy stanami, która stanowi podstawę do dalszych oszacowań (Norris 1997).

Reguły sekwencyjne (SR, ang. Sequential Rules)

Podejście to podobnie jak wcześniejsze uwzględnia kolejność żądań, ale w mniej restrykcyjny sposób. W przeciwieństwie do metody MC tworzone są reguły, gdy żądanie q_n pojawiło się po żądaniu q_m w sesji, nawet jeśli między q_n a q_m wystąpiły inne żądania. Dodatkowo następuje przypisanie wag żądaniom. Waga ta zależy od liczby elementów pojawiających się między q_n a q_m. Konkretnie, funkcja wagi ma postać $wsr(x) = 1/x$, gdzie x odpowiada liczbie żądań między dwoma elementami, co stanowi bazę do oszacowania predykcji (Kamehkhosh i in. 2017).

Item-based kNN (IKNN)

Metoda ta uwzględnia jedynie ostatni element w danej sesji, a następnie zwraca jako rekomendacje te obiekty, które są do niego najbardziej podobne pod względem ich współwystępowania w innych sesjach. Technicznie rzecz biorąc, każdy obiekt jest kodowany jako wektor binarny, gdzie każdy element odpowiada sesji i jest ustawiany na „1” w przypadku, gdy obiekt pojawił się w sesji lub „0” w przeciwnym przypadku. Podobieństwo dwóch pozycji można następnie określić, np. za pomocą miary podobieństwa cosinusowego (Hidasi i in. 2016).

Session-based kNN (SKNN)

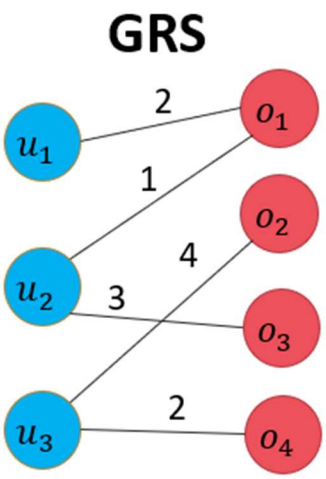
W podejściu tym, zamiast brać pod uwagę tylko ostatnie zdarzenie w bieżącej sesji, porównuje się całą bieżącą sesję $Q_n \epsilon S$, gdzie zostało zdefiniowane w (2.3), z sesjami przeszłymi $[Q_1, Q_2, \dots, Q_{n-1}]$ w celu określenia elementów, które mają być rekomendowane. Technicznie najpierw określa się k najbardziej podobnych sesji, stosując odpowiednią miarę podobieństwa sesji, np. indeks Jaccarda lub podobieństwo cosinusowe (Lerche i in. 2016).

3.2.7. Bazujące na grafach heterogenicznych

Grafy wykorzystywane w obszarze rekomendacji nazywane są „graph-based recommender system" i określane skrótem GRS (Hekmatfar i in. 2021). W szczególności w wielu algorytmach i systemach rekomendacji istotną rolę stanowią grafy heterogeniczne. Zasadniczo pojęcie heterogeniczności jest związane z niejednorodnością (zróżnicowaniem) (PWN 2022). Graf jest heterogeniczny, jeśli zawiera różne typy węzłów i krawędzi (łuków) (X. Wang i in. 2019). Tego typu modele nazywa się również heterogenicznymi sieciami informacyjnymi (ang. Heterogeneous Information Network) I są określane skrótem HIN (Shi i in. 2015).

W GRS podstawowymi typami węzłów N są obiekty O i użytkownicy U. Natomiast typem krawędzi E są to krawędzie łączące obiekty z użytkownikami $(o, u) \in E$, gdzie $o \in O$ i $u \in U$ (Hekmatfar i in. 2021). Jeśli do krawędzi zostanie dodana funkcja wag $w(e)$, na przykład reprezentująca oceny użytkowników nadawane obiektom, wówczas mamy do czynienia z grafem ważonym, który stanowi podstawę do konstruowania macierzy ocen (RM, ang. rating matrix) (Ramlatchan i in. 2018). Macierz ocen ma wymiar $|U|x|O|$, gdzie $|U|$ i $|O|$ oznaczają moce (liczność) zbiorów U i O. Ponadto element rm_{uo} macierzy RM przyjmuje wartość funkcji wagi $w(e)$ dla $e \in E$ i 0 dla $e \notin E$, gdzie $e = (u, o)$.

Macierz tego typu stanowi podstawę dla całej grupy algorytmów rekomendacji bazujących na technice Collaborative Filtering.



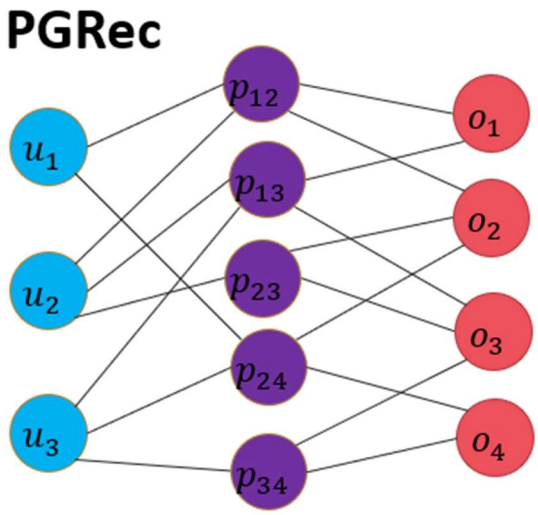
RM

	o_1	o_2	o_3	o_4
u_1	2	0	0	0
u_2	1	0	3	0
u_3	0	4	0	2

Rysunek 3.3. Przykład podstawowego modelu heterogenicznego GRS i jego odpowiednik w postaci macierzy ocen RM. Graf zawiera dwa typy węzłów powiązanych ze sobą krawędziami z wagami.
Źródło: opracowanie własne.

Kolejną grupę GRS stanowią tak zwane grafy preferencji „Preference Graph based Recommendation" określane skrótem PGRec (Hekmatfar i in. 2021). Główną ideą PGRec jest modelowanie problemu systemu rekomendacji jako problemu przewidywania wagi w nowym typie grafu heterogenicznego. Graf tego typu posiada dodatkowy typ węzłów w postaci zbioru preferencji $P = \{p_1, p_2, \dots, p_n\}$. Inna nazwa spotykana w literaturze dla tego typu grafów jest „Tripartite Preference Graph" określana jako TPG (Shams i Haratizadeh 2017).

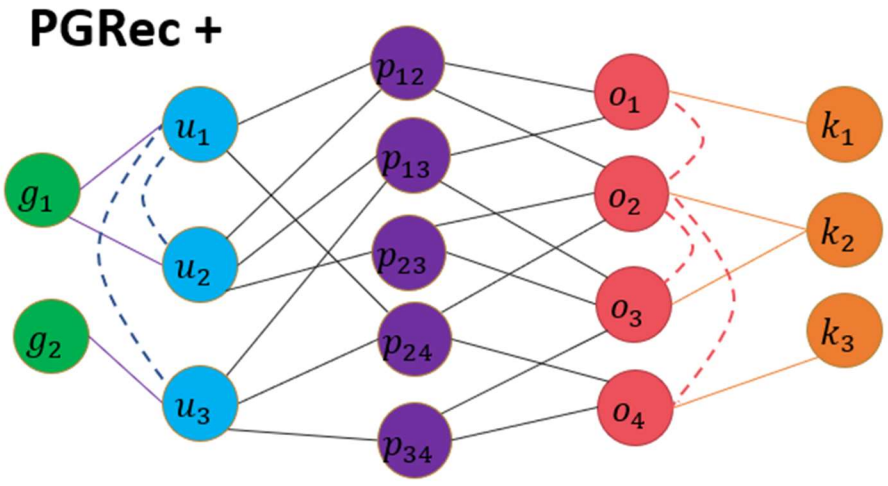
PGRec jest heterogenicznym grafem ważonym, który w swojej najprostszej postaci jest grafem trójdzielnym, w którym $N = U \cup P \cup O$ i $E = E_{po} \cup E_{up}$, gdzie $U \cap P \cap O = \emptyset$ i $E_{po} \cap E_{up} = \emptyset$. E_{po} reprezentuje ważone krawędzie łączące węzły preferencji z węzłami obiektów o wadze W_{PO}. E_{up} to zestaw ważonych krawędzi łączących węzły użytkowników z węzłami preferencji z wagą W_{UP}. Szczegółowa interpretacja wag W_{PO} i W_{UP} zależy od algorytmów wykorzystujących PGRec.



Rysunek 3.4. Przykład PGRec. Graf zawiera trzy typy węzłów: użytkownicy, preferencje i obiekty.
Źródło: opracowanie własne.

Bardziej rozbudowana forma PGRec+ zawiera dodatkowe informacje o użytkownikach w postaci zbioru grup $G = \{g_1, g_2, \dots, g_n\}$ i dodatkowe informacje o obiektach w postaci kategorii $K = \{k_1, k_2, \dots, k_n\}$. Jest to realizowane poprzez dodanie kolejnych typów węzłów do grafu PGRec. Nowe węzły należące do G i K są połączone krawędziami nieważonymi z odpowiednimi użytkownikami lub obiektami. Ponadto można dodać krawędzie w ramach poszczególnych warstw na przykład pomiędzy użytkownikami ($U - U$) lub obiektami ($O - O$).

Systemy rekomendacji wykorzystujące opisany powyżej graf preferencji to przede wszystkim bazujące na rodzinie algorytmów GRank (Shams i Haratizadeh 2017).



Rysunek 3.5. Przykład rozbudowanego PGRec+. Graf zawiera 7 typów węzłów. W stosunku do pierwotnego modelu PGRec zostały dodane typy węzłów zawierające dodatkowe informacje o użytkownikach (grupy) i obiektach (kategorie).
Źródło: opracowanie własne.

W ostatnich latach obserwuje się gwałtowny wzrost badań nad HIN w postaci PGRec+. Jedną z metod badania tych struktur jest analiza ścieżek i ich semantyczna interpretacja, która również znajduje zastosowanie w budowie algorytmów rekomendacji (Shi i in. 2015).

Wzór ścieżki	Semantyczna interpretacja	Technika rekomendacji
UU	znajomi	Social recommendation
UGU	użytkownicy w tej samej grupie (z tą samą cechą)	Member recommendation
UOU	użytkownicy, którzy interesują się tym samym produktem	Collaborative recommendation
UOKOU	użytkownicy, którzy interesują się produktami tej samej kategorii (z tą samą cechą)	Content recommendation

Tabela 3.1. Przykłady ścieżek i ich interpretacja w odniesieniu do technik rekomendacji. Do budowy wzorów ścieżek zostały wykorzystane typy (zbiory) węzłów występujące w grafie PGRec+ będącym również przykładem HIN.
Źródło: „Semantic Path based Personalized Recommendation on Weighted Heterogeneous Information Networks" (Shi i in. 2015).

3.2.8. Hybrydowe

Techniki tego typu wykorzystują właściwości wcześniej opisanych podejść i stanowią ich kompilację czemu umożliwiają eliminowanie wad poszczególnych podejść.

Między innymi techniki hybrydowe wykorzystują profile użytkowników i opisy obiektów, aby znaleźć użytkowników o podobnych zainteresowaniach, a następnie zbliżone obiekty w celu prognozowania (Cremonesi i in. 2010).

3.3. Systemy rekomendacji

Na bazie technik rekomendacji konstruowane są algorytmy rekomendacji, które są implementowane w systemach rekomendacji. Systemy te realizują proces rekomendacji.

3.3.1. Fazy procesu rekomendacji

W zaimplementowanych w systemach rekomendacji procesach można wyróżnić następujące trzy główne fazy:

Faza 1. Gromadzenie informacji

W trakcie trwania tej fazy gromadzone są dane o obiektach i użytkownikach. Ma to na celu wygenerowanie zbioru atrybutów i ich wartości. Docelowe rozwiązanie musi posiadać jak najwięcej informacji na temat tych elementów, aby od samego początku udzielać skutecznych rekomendacji (Isinkaye i in. 2015) (Kumar i Kumar 2019).

Faza 2. Uczenie się

Podczas tej fazy, na podstawie pozyskanych informacji, budowane są modele zależności pomiędzy obiektami względem siebie, użytkownikami względem siebie oraz między obiektami a użytkownikami. Modele te mogą mieć różne implementacje na przykład list, macierzy, implikacji lub grafów (Kumar i Kumar 2019).

Faza 3. Przewidywanie

Celem tej fazy jest analiza modeli uzyskanych w fazie uczenia się, a w konsekwencji pozwala dostarczyć rankingi lub prognozy dla zebranych danych. Opracowane modele w fazie uczenia się dostarczają wzorców, które następnie stanowią dane wejściowe dla algorytmów rekomendacji, które to jako wyniki swego działania zwracają określone rekomendacje (Kumar i Kumar 2019).

3.3.2. Główne problemy przy stosowaniu rekomendacji

Podczas implementacji i produkcyjnego funkcjonowania systemów rekomendacji w zastosowaniach e-Commerce mogą występować następujące problemy:

- zimny start (ang. cold start) – odnosi się do sytuacji, kiedy system nie posiada wystarczających informacji o użytkownikach lub obiektach w celu oszacowania rekomendacji. Taka sytuacja ma bardzo często miejsce podczas uruchomienia systemu (Burke 2007);
- rzadkość (mała liczba) danych (ang. sparsity) - jest to problem, który pojawia się w wyniku braku wystarczającej liczby informacji dotyczących ocenianego obiektu lub oceniającego użytkownika. Tego typu sytuacja pojawia się wówczas, gdy do

bazy danych zostaje dodany nowy obiekt lub z systemu korzysta nowy użytkownik (Sarwar i in. 2000);

- skalowalność (ang. scalability) - problem ten na ogół związany jest z algorytmami i wynika z faktu, że wraz ze wzrostem liczby użytkowników i obiektów system rekomendacji potrzebuje więcej zasobów (pamięci, mocy obliczeniowej) do przetwarzania informacji i szacowania rekomendacji. Jest to szczególnie istotny problem w przypadku rozwiązań e-Commerce, które w swej naturze dedykowane są dla dużej liczby użytkowników i wymagają szybkich oszacowań (Sarwar i in. 2000);
- długi ogon (ang. long-tail) – jest to zjawisko charakteryzujące się tym, że niewielka liczba obiektów jest bardzo popularna (często rekomendowana), a reszta, stanowiąca większość, występuje w wynikach rekomendacji znacznie rzadziej (Liu i Zheng 2020);
- synonimy (ang. synonyms) - to tendencja do posiadania w bazie różnych nazw lub wpisów dla bardzo podobnych obiektów. Większość systemów rekomendacji ma trudności z powiązaniem tego typu obiektów i traktowania ich jako swego rodzaju „zamienników" (Sarwar i in. 2000).

Oprócz wyżej wymienionych zagadnień jest również kilka ważnych czynników, które nie są bezpośrednio związane z aspektami informatyczno-technicznymi systemów rekomendacji, a mają wpływ na ich funkcjonowanie. Są to:

- różnorodność (ang. diversity) - użytkownicy są bardziej zadowoleni z rekomendacji, gdy występuje większe zróżnicowanie wewnątrz list rekomendowanych obiektów, np. przedmioty od różnych producentów (Ziegler i in. 2005);
- wytrwałość rekomendowania (ang. recommender persistence) - w niektórych sytuacjach skuteczniejsze jest ponowne pokazanie tej samej rekomendacji lub pozwolenie użytkownikom na ponowną ocenę obiektu niż pokazywanie nowych obiektów. Dzieje się tak na przykład dlatego, że użytkownicy mogą zignorować obiekt, gdy był wyświetlany po raz pierwszy lub nie było czasu na dokładne zapoznanie się z nim (Beel, Nürnberger, i in. 2013);
- prywatność (ang. privacy) – użytkownicy, podczas korzystania z systemów rekomendacji, mają obawy dotyczące prywatności, ponieważ muszą ujawniać informacje na swój temat (wiek, wykształcenie, płeć, zainteresowania, miejsce

zamieszkania, preferencje, itd.…). W wielu krajach europejskich panuje silna kultura prywatności danych, a każda próba wprowadzenia obszerniejszego poziomu profilowania użytkownika skutkuje negatywną reakcją (Pu i in. 2012);

- dane demograficzne użytkowników (ang. user demographics) – ustalono, że dane demograficzne użytkowników (w szczególności wiek) wpływa na to, jak są oni zadowoleni z rekomendacji (Joeran i in. 2013);
- zaufanie (ang. trust) - system rekomendacji ma niewielką wartość dla użytkownika, jeśli użytkownik mu nie ufa. Zaufanie buduje się, wyjaśniając w jaki sposób są generowane rekomendacje i dlaczego polecany jest dany obiekt (Montaner i in. 2002);
- oznakowanie (ang. labelling) - na reakcję użytkownika na rekomendację ma wpływ jej oznakowanie. Na przykład zbadano, że współczynnik kliknięć CTR rekomendacji oznaczonych jako „Sponsorowane" był niższy niż CTR dla identycznych rekomendacji oznaczonych jako „Bezpłatne". Ponadto rekomendacje „bez etykiety" wypadły najlepiej (Beel, Langer, i in. 2013);
- prezentacja (ang. presentation) – na skuteczność rekomendacji ma wpływ forma i miejsce prezentacji rekomendacji w systemie e-Commerce. Na przykład prezentacja rekomendowanych obiektów na górze strony WWW serwisu internetowego jest efektywniejsze niż umiejscowienie ich na dole (Nanou i in. 2010).

Każdy z ww. problemów ma znaczenie dla efektywności działania systemu rekomendacji. Część z nich zostanie przedstawiona szczegółowo w opracowaniu oraz zbadana w ramach eksperymentów przeprowadzonych na autorskim algorytmie rekomendacji bazującym na sesjach rekomendacji w rozdziale 7.

3.4. Sformułowanie problemu badawczego rekomendacji

Z obszarem rekomendacji związany jest problem badawczy opisany w następujący sposób. Niech $C = \{c_1, c_2, \dots, c_n\}$ będzie skończonym zbiorem użytkowników wykorzystujących system rekomendacji i $O = \{o_1, o_2, \dots, o_n\}$ będzie skończonym zbiorem wszystkich możliwych obiektów, które mogą być rekomendowane. Ponadto niech u będzie funkcją użyteczności mierzącą użyteczność obiektu $o \in O$ dla użytkownika $c \in C$. To jest $u: CxO \rightarrow W$, gdzie W jest uporządkowanym zbiorem (np.

nieujemnych liczb całkowitych). Następnie, dla każdego użytkownika $c \in C$ zostaje wybrany podzbiór $R_c^* \subset O$, który maksymalizuje użyteczność dla użytkownika. Oznacza to, że dla ustalonego $c \in C$ wyznaczany jest zbiór rekomendowanych obiektów następująco (Adomavicius i Tuzhilin 2005):

$$R_c^* = \{r \in O: u(c,r) = \max_{o \in O} u(c,o)\} \quad (3.1)$$

W przypadku ogólnym zadaniem algorytmów rekomendacji jest znalezienie podzbioru R_c^* zwanego zbiorem rekomendowanych obiektów dla użytkownika c.

Jednym z głównych problemów związanych z działaniem algorytmów rekomendacji jest rzadkość danych wynikająca z posiadania nielicznych lub braku informacji o użytkownikach (McAuley i Leskovec 2013). Może mieć to miejsce w następujących sytuacjach: użytkownik po raz pierwszy ma kontakt z systemem lub system nie gromadzi informacji o użytkownikach co może mieć związek z problemem prywatności. W związku z powyższym zadanie ulega następującej modyfikacji polegającej na tym, że rekomendacji dokonuje się względem wybranego obiektu m bez bezpośredniego uwzględnienia użytkownika c, dla którego rekomendacja następuje.

W związku z powyższym zadanie (3.1) uwzględniając zmianę w funkcji użyteczności taką, że $w: OxO \rightarrow W$ ulega zmianie tak, że zbiór rekomendacji dla użytkownika R_c^* zostaje zastąpiony przez R_m^*, czyli zbiór rekomendacji dla obiektu. W konsekwencji, dla ustalonego $m \in O$, zostaje wyznaczony zbiór rekomendowanych obiektów następująco (Malinowski 2020):

$$R_m^* = \{r \in O: w(m,r) = \max_{o \in O} w(m,o)\} \quad (3.2)$$

Uwzględnienie użytkownika c polega na tym, że to on świadomie kierując swoim zachowaniem (kliknięciem), dokonuje wyboru obiektu m ze zbioru O, względem którego będzie budowana rekomendacja R_m^*. Ponad to funkcja $w(m,o)$ określa popularność obiektu $o \in O$ o użyteczności podobnej do obiektu $m \in O$.

Problem badawczy, rozpatrywany w pracy, polega na skonstruowaniu skutecznego i efektywnego algorytmu rekomendacji rozwiązującego zadanie w postaci (3.2). Propozycja takiego algorytmu, bazującego na sesjach rekomendacji tworzonych na podstawie zachowań użytkowników oraz atrybutów obiektów w systemie e-Commerce, przedstawiona została w rozdziale 5.

3.4.1. Obiekt

Obiekt $o \in O$ fizycznie może reprezentować produkt w sklepie internetowym, film w wypożyczalni na życzenie (VOD, ang. Video on Demand), ofertę pracy w serwisie związanym z pracą lub ogłoszenie w serwisie ogłoszeniowym. Informacje (atrybuty) związane z obiektem dzieli się na: materialne, niematerialne, dynamiczne lub złożone. Należy zaznaczyć, że kiedy użytkownik nabywa jakiś obiekt, zawsze ponosi koszt, na który składa się koszt pieniężny oraz koszt poznawczy związany z poszukiwaniem odpowiedniego obiektu. Na przykład, nawet jeśli użytkownik nie płaci za czytanie wiadomości w serwisie informacyjnym, to zawsze istnieje koszt związany z wyszukaniem wiadomości. Jeśli wybrana wiadomość jest istotna dla użytkownika, wówczas koszt jest zdominowany przez korzyści wynikające ze zdobycia użytecznych informacji. Natomiast jeśli dana wiadomość nie jest istotna, wówczas dominuje poczucie tak zwanej „straty czasu" (Ricci i in. 2015).

Obiekty mogą być reprezentowane na przykład: w sposób minimalistyczny jako pojedynczy numer ID (identyfikator) lub w bogatszej formie jako wektor wartości atrybutów przedstawiony w rozdziale 2.3.

3.4.2. Użytkownik

Użytkownik $c \in C$ to przeważnie osoba korzystająca z systemu. Jednak nie zawsze, w szczególnych wypadkach użytkownika może reprezentować na przykład usługa informatyczna lub Bot. Użytkownika opisują parametry statyczne, na przykład płeć, wiek lub dane geograficzne, oraz dane dynamiczne wynikające z wcześniejszych zachowań podejmowanych w ramach systemu (np. kliknięcie w stronę WWW produktu, dodanie do koszyka lub zakup) i opisane w rozdziale 2.1. W celu personalizacji systemy rekomendacji wykorzystują szereg informacji o użytkownikach. Wybór informacji, które mają być modelowane, zależy od techniki rekomendacji. Na przykład w Collaborative Filtering użytkownicy są modelowani jako prosta lista zawierająca oceny wystawione przez użytkowników określonym obiektom (Berkovsky i in. 2009). W systemie rekomendacji bazującym na demografii wykorzystuje się atrybuty socjodemograficzne, takie jak wiek, płeć, zawód i wykształcenie. Dane o użytkownikach stanowią model użytkownika (Fischer 2001). Model profiluje użytkownika, tzn. odwzorowuje jego preferencje i potrzeby. Ponadto, dane

o użytkownikach często zawierają relacje pomiędzy nimi. Systemy rekomendacji wykorzystują te informacje do rekomendowania użytkownikom obiektów, które są preferowane przez na przykład znajomych (Berkovsky i in. 2008).

3.5. Miary oceny systemów rekomendacji

Ewaluacja systemów rekomendacji wymaga zidentyfikowania miar, które czynią je doskonalszymi. Definiuje się kilka grup miar, które są wykorzystywane jako charakterystyki systemów rekomendacji.

Typowymi miarami służącymi do oceny jakości rekomendacji są: czułość (ang. recall), precyzja (ang. precision), dokładność (ang. accuracy), F_β oraz dokładność (ang. accuracy) (Fayyaz i in. 2020). Rodzaj stosowanych miar zależy od rodzaju techniki rekomendacji (Isinkaye i in. 2015).

Zasadniczo miary oceny jakości przewidywania rekomendacji budowane są na podstawie macierzy pomyłek (ang. Confusion Matrix), która reprezentuje cztery możliwe wyniki rekomendacji przedstawione w tabeli poniżej.

		rzeczywistość	
Rekomendacje systemu		obiekt atrakcyjny dla klienta	obiekt nieatrakcyjny dla klienta
	rekomendowany	a	b
	nie rekomendowany	c	d

Tabela 3.2. Macierz pomyłek dla systemu rekomendacji.
Źródło: „Recommendation systems: Algorithms, challenges, metrics, and business opportunities" (Fayyaz i in. 2020)

Gdzie wartość "a" oznacza liczbę obiektów, które są atrakcyjne dla klienta i rekomendowane przez system rekomendacji. Wartość "b" oznacza liczbę obiektów, które były rekomendowane przez system, ale nie są w rzeczywistości atrakcyjne dla klienta. Wartość "c" reprezentuje liczbę pozycji, które są atrakcyjne dla klienta, ale nie zostały rekomendowane przez system. Wartość "d" to liczba obiektów, które są nie są atrakcyjne dla klienta i nie były rekomendowane przez system (Fayyaz i in. 2020).

Jeśli potraktować rekomendacje systemu jako klasyfikację obiektów do grupy atrakcyjnych i nieatrakcyjnych dla klienta, to:

- wartość „a” oznacza liczbę przypadków TP – TRUE POSITIVE;
- wartość „b” oznacza liczbę przypadków FP – FALSE POSITIVE;
- wartość „c” oznacza liczbę przypadków FN – FALSE NEGATIVE;
- wartość „d” oznacza liczbę przypadków TN – TRUE NEGATIVE.

Zatem, można ocenić jakość algorytmu rekomendacji za pomocą miar jakości stosowanych dla algorytmów klasyfikacji, definiowanych na podstawie macierzy pomyłek:

$$\text{precision} = \frac{a}{a+b} \qquad (3.3)$$

Precyzja (ang. precision) - mówi o tym, jaka część obiektów rekomendowanych przez system było atrakcyjnych dla klienta.

$$\text{recall} = \frac{a}{a+c} \qquad (3.4)$$

Czułość (ang. recall) - mówi o tym, jaka część obiektów atrakcyjnych dla klienta była rekomendowana przez system.

$$F_\beta = (1+\beta^2)\frac{\text{precision}*\text{recall}}{(\beta^2*\text{precision})+\text{reca}} \qquad (3.5)$$

F_β to średnia harmoniczna pomiędzy precyzją i czułością. Jeżeli obie miary są ważne to podstawiamy za $\beta = 1$, jeżeli bardziej zależy nam na precyzji to β powinna być z przedziału (0,1), w przeciwnym przypadku gdy bardziej zależy na czułości to $\beta > 1$.

$$\text{accuracy} = \frac{a+d}{a+b+c+d} \qquad (3.6)$$

Dokładność (ang. accuracy) - mówi, jaka część rekomendacji, ze wszystkich, została dokonana poprawnie. W tym przypadku oznacza to, że obiekt atrakcyjny dla klienta był rekomendowany przez system, a obiekt nietrakcyjny dla klienta nie był przez system rekomendowany.

Na bazie macierzy pomyłek jest budowanych bardzo dużo różnych miar związanych z oceną jakości przewidywania rekomendacji. Powyżej zostały przedstawione tylko najważniejsze z nich. Podstawowym warunkiem zastosowania tych miar jest posiadanie wiarygodnego wzorca decyzji a w przypadku systemów rekomendacji informacji o prawidłowych, przewidywanych rekomendacjach. Niestety

w wielu przypadkach posiadanie takiego wzorca nie jest możliwe ze względu na brak jednoznacznej metody określenia, czy rekomendacja jest przewidywana, czy też nie (Vaishnavi i in. 2013).

Popularnymi miarami dokładności dla systemów rekomendacji dostarczających wyniki w formie przewidywania ocen są średni błąd bezwzględny określany jako MAE (ang. Mean Absolute Error) oraz średni błąd kwadratowy określany przez RMSE (ang. Root Mean Squared Error). Aby było możliwe dokonanie oszacowania tych miar dla zadanego systemu rekomendacji konieczne jest posiadanie testowego zbioru ocen R_{test} , służącego do oceny dokładności (Ricci i in. 2015). Przedstawione miary szacuje się zgodnie ze wzorami:

$$\mathrm{MAE} = \sqrt{\frac{1}{R_{test}} \sum_{r_{ui} \in R_{test}} |\hat{r}_{ui} - r_{ui}|} \qquad (3.7)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{R_{test}} \sum_{r_{ui} \in R_{test}} (\hat{r}_{ui} - r_{ui})^2} \qquad (3.8)$$

gdzie:

$\hat{r}_{ui}$ – oszacowana ocena dla zadanego użytkownika u oraz obiektu i

Istotną grupę, z punktu widzenia rozwiązań e-Commerce stanowią metody oceny oparte na użyteczności stosowane do oceny efektywności systemów rekomendacji dla użytkowników. Wywodzą się z reklamy internetowej i handlu elektronicznego. W ich przypadku pomiary są realizowane online na bazie danych pozyskiwanych z działających w czasie rzeczywistym systemów (Beel i Langer 2015). Do tej grupy miar zalicza się współczynnik kliknięć określany jako CTR (ang. Click-Through Rate) (Kuanr i Mohapatra 2021). Jest on definiowany zgodnie ze wzorem:

$$\mathrm{CTR} = \frac{D_{click}}{D_{all}} * 100\% \qquad (3.9)$$

gdzie:

D_{all} – liczba wszystkich wyświetlonych rekomendacji

D_{click} – liczba rekomendacji klikniętych

Kolejną miarą systemu rekomendacji szczególnie istotną z punktu widzenia e-Commerce jest pokrycie (ang. coverage) i definiuje się jako iloraz wszystkich obiektów, które mogą być rekomendowane do wszystkich obiektów systemu e-Commerce.

W wielu przypadkach miarę tę można obliczyć bezpośrednio na podstawie algorytmu i zbioru danych wejściowych (Ricci i in. 2015).

$$\text{coverage} = \frac{|O_r|}{|O|} \qquad (3.10)$$

gdzie:

$|O_r|$ – liczba obiektów mogących wystąpić w rekomendacjach

$|O|$ – liczność zbioru wszystkich obiektów w systemie e-Commerce

W dalszej części pracy zostaną zaprezentowane wyniki przeprowadzonych eksperymentów online, w ramach których zostaną przedstawione charakterystyki opisanych powyżej miar dla badanych systemów rekomendacji. Z przedstawionych powyżej miar zostały wybrane, te które były możliwe do przedstawienia na bazie posiadanych danych.

4. Algorytmy rekomendacji oparte na wykorzystaniu analizy asocjacji

Zbliżoną w swej idei do autorskiego algorytmu rekomendacji ARS bazowania na zachowaniach użytkowników i wyciąganiu na tej podstawie wniosków użytecznych w obszarze e-Commerce jest analiza asocjacji (ang. association analysis). Podstawy jej funkcjonowania zostały opisane w rozdziale 2.4. Celem przypomnienia, metody analizy asocjacji należą do grupy technik eksploracji danych. Ich celem jest kojarzenie (asocjacja) wzorców zakupowych klientów na bazie danych transakcji zakupowych (koszyk zakupów) (Rao i in. 2021). Celem opisu tej metody jest jej porównanie ze skutecznością algorytmu ARS.

4.1. Analiza asocjacji

Analiza asocjacji służy do znajdowania w dużym zbiorze danych ukrytych zależności w postaci prostych reguł. Analiza ta polega na badaniu współwystępowania obiektów w podzbiorach. Wyniki analizy tego typu mają postać reguł asocjacji (AR, ang. association rules), czyli prostych zdań warunkowych (implikacji) postaci: jeżeli A, to B zapisywanych w postaci $A \rightarrow B$.

Każda reguła asocjacji $A \rightarrow B$ składa się z dwóch zbiorów (zestawów) obiektów (ang. itemsets) A i B, gdzie A jest nazywane „poprzednikiem” (LHS ,ang. „left-hand-side”, antecedent) a B jest nazywany „następnikiem” (RHS, ang. „right-side-side”, consequent). Zbiory A i B są różne, czyli $A \cap B = \emptyset$ (Larose 2006).

Celem stosowania w obszarze e-Commerce analizy asocjacji jest odkrywanie wzorców zachowań klientów, aby między innymi uzyskać odpowiedź na pytania biznesowe takie jak:

- które produkty kupowane są najczęściej razem?
- które produkty wykluczają się, a które wspierają nawzajem swoją sprzedaż?
- jakie jest prawdopodobieństwo, że klienci, którzy kupili produkt A, kupią również produkt B?

Zasadniczo celem analizy asocjacji jest pozyskanie informacji z bazy danych transakcji będącej strukturą informacyjną przechowującą dane o transakcjach zakupowych dokonanych przez klientów w sklepie (Agrawal i in. 1993).

W celu sprecyzowania pojęcia reguł asocjacji definiuje się następujące pojęcia (Pęczkowski M. i Lasek M. 2013):

$O = \{o_1, o_2, \dots, o_n\}$ - zbiór wszystkich dostępnych obiektów (produktów) w systemie (sklepie)

$T \subseteq O$ gdzie $T \neq \emptyset$ - transakcja (koszyk, zamówienie) będąca zbiorem obiektów (produktów), które pojedynczy klient kupił podczas jednych zakupów w systemie (sklepie)

$D = \{T_1, T_2, \dots, T_n\}$ - zbiór transakcji zwany bazą danych transakcji lub koszykiem zakupów

Ponadto mówi się, że:

- transakcja T wspiera element $x \in O$, jeżeli $x \in T$
- transakcja T wspiera zbiór $X \subseteq O$, jeżeli T wspiera każdy element ze zbioru X to znaczy $X \subseteq T$

oraz definiuje się:

- $A \rightarrow B$ – reguła asocjacji gdzie $A, B \subseteq O$ i $A \cap B = \emptyset$
- $cover(X)$ - pokrycie zbioru obiektów $X \subseteq O$ w bazie D, to zbiór transakcji zawierających obiekty ze zbioru X

$$cover(X) = \{T: X \subseteq T \ \wedge X \subseteq O \wedge \ T \in D\} \qquad (4.1)$$

- $supp(A \rightarrow B)$ - wsparcie (ang. support) to iloraz liczby transakcji zawierających obiekty ze zbioru A i B, gdzie $A, B \subseteq O$ i $A \cap B = \emptyset$, do liczby wszystkich transakcji ze zbioru D zgodnie ze wzorem:

$$supp(A \rightarrow B) = \frac{|cover(A \cup B)|}{|D|} \qquad (4.2)$$

- $conf(A \rightarrow B)$ - ufność, wiarygodność (ang. confidence) to iloraz liczby transakcji zawierających obiekty ze zbioru A i B, gdzie $A, B \subseteq O$ i $A \cap B = \emptyset$, do liczby transakcji zawierających obiekty ze zbioru A zgodnie ze wzorem:

$$conf(A \rightarrow B) = \frac{|cover(A \cup B)|}{|cover(A)|} \qquad (4.3)$$

Problem odkrywania reguł asocjacji w bazie D jest problemem wyszukiwania wszystkich silnych reguł asocjacji $A \rightarrow B$, czyli takich, które przy ustalonych wartościach minimalnego wsparcia s i minimalnej ufności c spełniają poniższe warunki łącznie (Agrawal i in. 1993):

- $supp(A \rightarrow B) \geq s$
- $conf(A \rightarrow B) \geq c$

Należy zaznaczyć, że wskazanie minimalnego wsparcia s i minimalnej ufności c stanowi bardzo duży problem w obszarze odkrywania reguł asocjacji. Zawyżone wartości mogą powodować, że liczba odnalezionych reguł będzie bardzo niska a zaniżone powodują natomiast znaczne obciążenie systemu informatycznego dokonującego oszacowania reguł. W praktyce wartości tych parametrów są często wskazywane przez ekspertów z danej dziedziny (Hikmawati i in. 2021).

4.2. Algorytmy odkrywania reguł asocjacji

Przedstawiony we wcześniejszym rozdziale problem odkrywania reguł asocjacji w ramach analizy asocjacji jest rozwiązywany przez dedykowane w tym celu algorytmy. Pierwszy algorytm tego typu został przedstawiony na początku lat 90' ubiegłego wieku w ramach publikacji wyników prac Agrawala i Srikanta (Agrawal i in. 1993). Opisali oni dwa algorytmy odkrywania silnych reguł asocjacji o nazwach: Apriori i AprioriTID. Zawarte w nich idee stanowią podstawę do budowy i rozwoju wielu nowych algorytmów tego typu. Idea ta sprowadza się do podejścia bazującego na zasadzie „generuj i testuj" (ang. genetate-and-test). Należy zaznaczyć, że istnieje alternatywna grupa algorytmów tego typu bazująca na idei „konstrukcji wzorca" (ang. patern growth). Pierwszy algorytm tego typu o nazwie FP-Growth został opisany w pracy Jiawei Han, J. Pei oraz Yiwen Yin (Han i in. 1999).

Prowadzone badania nad algorytmami tego typu w zakresie wydajności zarówno na podstawie eksperymentów szacowania czasu pracy, jak i rozważań teoretycznych. Mimo zasadniczych różnic dotyczących zastosowanych strategii ich funkcjonowania wykazują dość podobne cechy wydajnościowe. Badania te nie wykazują algorytmu, który w sposób zasadniczy pokonuje pozostałe. Można przyjąć, że w rzeczywistości

wady i zalety, tych algorytmów równoważą się, a konkretne zastosowanie danego algorytmu zależy od architektury bazy danych podlegającej analizie asocjacji (Nakhaeizadeh i in. 2000). Z tego też powodu do porównania funkcjonowania algorytmu ARS z konkurencyjnym rozwiązaniem podczas badań będzie rozważny algorytm Apriori jako podstawowy algorytm wyszukiwania reguł asocjacji i bazujący na nim algorytm rekomendacji (Osadchiy i in. 2018).

Z reguły algorytmy wyszukiwania reguł asocjacji składają się z dwóch etapów (Morzy 2022):

1) generowania zbiorów częstych na podstawie minimalnego wsparcia s,
2) na podstawie utworzonych zbiorów częstych odkrywania reguł o ufności większej niż minimalna ufność c.

Kluczową właściwością wykorzystywaną przy generowaniu zbiorów częstych jest tzw. własność Apriori zwana również własnością antymonotoniczną, z której wynika, że każdy podzbiór zbioru częstego jest zbiorem częstym, albo inaczej: jeżeli jakiś zbiór nie jest zbiorem częstym, to jego nadzbiór też nie jest zbiorem częstym. Formalnie właściwość antymonotoniczności (Apriori) zachodzi wówczas jeśli:

$$\forall_{X,Y \in Z}\, X \subseteq Y \Rightarrow f(X) \geq f(Y) \qquad (4.4)$$

czyli jeżeli X jest podzbiorem zbioru Y, to wartość $f(Y)$ nie może być większa od $f(X)$ (Andrzejewski 2014). Własność Apriori ułatwia przeszukiwanie zbiorów, ponieważ jeżeli jakiś zbiór A nie jest częsty, to można pominąć w rozważaniach wszystkie jego nadzbiory, tzn. zbiory X, takie, że $A \subset X$.

Podstawowy algorytm Apriori w części generowania zbiorów częstych bazuje na iteracyjnym zwiększaniu liczby elementów docelowego zbioru zbiorów częstych.

Oznaczmy:

C_k – zbiór transakcji k-elementowych

L_k – zbiór transakcji częstych k-elementowych

Algorytm zaczyna się od znalezienia wszystkich zbiorów częstych jednoelementowych L_1. Następnie L_1 jest wykorzystywany do tworzenia zbiorów częstych dwuelementowych L_2 i tak dalej, aż do momentu stwierdzenia, że dla

pewnego k nie ma już częstych zbiorów k-elementowych. W etapie szukania zbiorów częstych istotne są dwie główne operacje (Pęczkowski M. i Lasek M. 2013):

- łączenie (ang. join);
- przycinanie (ang. prune).

Operacja łączenia polega na realizacji czynności takiej, że mając ustalony L_{k-1} do zbioru C_k wstawiamy $A \cup B$ takich par $A, B \epsilon L_{k-1}$, które mają wspólne $k-2$ początkowych elementów.

Niech:

$A = \{i_{A_1}, \dots, i_{A_{k-2}}, i_{A_{k-1}}\}$,

$B = \{i_{B_1}, \dots, i_{B_{k-2}}, i_{B_{k-1}}\}$,

wówczas warunkiem dołączenia $A \cup B$ do zbioru C_k jest, aby:

$(i_{A_1} = i_{B_1}) \wedge \dots \wedge (i_{A_{k-2}} = i_{B_{k-2}}) \wedge (i_{A_{k-1}} < i_{B_{k-1}})$

Warunek $i_{A_{k-1}} < i_{B_{k-1}}$ został wprowadzony, aby zapobiec występowaniu powtarzających się elementów w zbiorze C_k.

Druga z wymienionych operacji etapu szukania zbiorów częstych to operacja przycinania. Powstający w wyniku łączenia transakcji zbiór C_k nie musi składać się z samych zbiorów częstych, ale wszystkie k-elementowe zbiory częste należą do C_k, czyli $L_k \subset C_k$. Celem operacji przycinania jest usunięcie ze zbioru C_k transakcji, które nie są częste. Wykorzystuje się tutaj własności Apriori. Usuwane z C_k są zbiory, których nie wszystkie podzbiory $(k-1)$ – elementowe należą do L_{k-1}. Z własności Apriori wynika, że jeżeli jakiś podzbiór takiego zbioru nie należy do L_{k-1}, to taki zbiór nie może należeć do L_k.

Algorytm generowania zbiorów częstych, w postaci pseudokodu przyjmuje poniższą postać (Pęczkowski M. i Lasek M. 2013):

Dane wejściowe:

T – zbiór transakcji

$minSupp$ – minimalne wsparcie

Dane wyjściowe:

L – rodzina zbiorów częstych

Niech:

C_k – rodzina kandydatów na zbiory częste k-elementowe

L_k – rodzina zbiorów częstych k-elementowych

Kroki algorytmu:

1. oblicz wsparcie dla wszystkich transakcji jednoelementowych tworzących zbiór $C_1 = \{\{i_1\}, \{i_2\}, \dots, \{i_m\}\}$
2. wybierz te transakcje jednoelementowe, które spełniają warunek minimalnego wsparcia i utwórz z nich zbiór L_1, taki, że $L_1 = \{x: supp(x) \geq minSupp\}$
3. $k = 1$
4. dopóki $|L_k| > k + 1$, gdzie $|L_k|$ jest mocą zbioru, czyli liczbą elementów w zbiorze powtarzaj:
   4.1. utwórz zbiór kandydatów $C_{k+1} = L_k \times L_k$ (łączenie)
   4.2. usuń z C_{k+1} zbiory, które zawierają nieczęsty podzbiór o rozmiarze k, tzn. zbiory $x \in L_k$
   4.3. oblicz wsparcie dla pozostałych zbiorów $x \in C_{k+1}$ (nieusuniętych w pkt. 5.2)
   4.4. usuń z C_{k+1} zbiory, które nie spełniają warunku minimalnego wsparcia
   4.5. z pozostałych elementów zbioru C_{k+1} utwórz zbiór L_{k+1}
   4.6. zwiększ k, tak że $k := k + 1$
5. jeżeli $L_k = \emptyset$, to $k := k - 1$
6. zachowaj wszystkie zbiory częste $L = \bigcup_{i=1}^{k} L_i$

Odkrywanie reguł na podstawie zbiorów częstych polega na zamianie zbioru częstego na regułę (wydzielenie poprzednika i następnika) i sprawdzeniu, czy tak określona reguła ma ufność nie mniejszą niż przyjęta minimalna wartość minConf. Definicja zbioru częstego określa, które elementy występują w regule, ale nie określa poprzednika i następnika. Budowanie reguł polega na przerzucaniu po kolei elementów z poprzednika do następnika i sprawdzaniu, czy w ten sposób utworzona reguła $A \rightarrow B$ spełnia warunek $conf(A \rightarrow B) \geq minConf$.

W tym celu najpierw tworzy się reguły zawierające jeden element w następniku. Następnie usuwa się reguły, które nie spełniają warunku minimalnej ufności.

W następnym kroku należy dla każdej z nieodrzuconych reguł przerzucić jeden element z poprzednika do następnika. Odrzuca się te reguły, które nie spełniają warunku minimalnej ufności. W algorytmie korzysta się z faktu, że jeżeli reguła $AB \rightarrow CD$ spełnia warunek minimalnej ufności, to $ABC \rightarrow D$ i $ABD \rightarrow C$ też spełniają Pseudokod ilustrujący algorytm odkrywania silnych reguł asocjacji ma postać (Pęczkowski M. i Lasek M. 2013):

Dane wejściowe:

L – rodzina zbiorów częstych

$minConf$ – minimalna ufność

Dane wyjściowe:

AM – zestaw silnych reguł asocjacji w postaci $y \rightarrow x$

Niech:

L_k – rodzina zbiorów częstych k-elementowych

Kroki algorytmu:

1. $k := 2$
2. dla każdego $x \in L_k$
   2.1. dla każdego $y \subset x$ gdzie $y \neq 0$, $y \neq x$
      2.1.1. zbuduj regułę $y \rightarrow x \backslash y$ (tzn. poprzednik y, a następnik zawiera te elementy z x, które nie należą do y)
      2.1.2. jeżeli $conf(y \rightarrow x \backslash y) \geq minConf$, to zapamiętaj tę regułę
3. zwiększ k, tak że $k := k + 1$
4. jeżeli $L_k = \emptyset$, to przejdź do punktu 2.
5. zwróć zapamiętane reguły AM

Teoretyczna złożoność czasowa i pamięciowa algorytmu Apriori wynosi $O(2^d)$, gdzie d jest liczbą unikalnych obiektów (Tahyudin i in. 2019). Zatem jest to algorytm wykładniczy, a funkcja złożoności jest klasy $O(2^n)$. Oznacza to, że algorytm ten jest efektywny dla małych danych natomiast dla dużej liczby danych będzie nieefektywny.

4.3. Algorytm rekomendacji bazujący na regułach asocjacji

Wyznaczone silne reguły asocjacji z wykorzystaniem algorytmu Apriori lub innego bazującego na nim algorytmu (na przykład AprioriTid, Apriori Hybryd (Agrawal i Srikant 1994)) stanowią podstawę do budowy grupy bazujących na nich systemów rekomendacji. Ich bardzo dużą zaletą jest to, że informacje nie są pozyskiwane ze statycznych charakterystyk obiektów (atrybutów) oraz ocen użytkowników, lecz bazują na ich zachowaniach takich jak na przykład zakupy lub odwiedziny podstron produktów (kliknięcia) (Zhang 2007).

Przykłady algorytmów wyznaczania rekomendacji na bazie reguł asocjacji zostały opisane w 2018r. przez T. Osadchiy, I. Poliakov, P.Olivier, M. Rowland oraz E.Foster w publikacji „Recommender system based on pairwise association rules" (Osadchiy i in. 2018). W przedmiotowej pracy algorytmy były badane pod kątem wykorzystania ich w obszarze zachowań żywieniowych respondentów.

Podejście przedstawione w pracy bazuje na tym, że algorytm rekomendacji nie posiada żadnej wcześniejszej wiedzy o danym użytkowniku osobie (użytkowniku z punktu widzenia e-Commerce), poza aktualnie wybieranymi przez nią produktami spożywczymi (elementy te odpowiadają zachowaniom użytkowników z obszaru e- Commerce). Ponadto algorytmy te bazują na zbiorach zaobserwowanych posiłków, gdzie posiłek to grupa unikalnie identyfikowalnych produktów (posiłek odpowiada transakcji z obszaru e-Commerce), które zostały zjedzone przy jednej okazji. Każdy z pokarmów (odpowiednik obiektu z obszaru e-Commerce) może być rejestrowany jako zjedzony tylko raz w trakcie posiłku. W kroku rekomendacji, dla produktów wprowadzonych (IF) zwracany jest zestaw produktów rekomendowanych (RF) (Osadchiy i in. 2018).

W przedmiotowej publikacji zostały opisane trzy algorytmy:

- bazujący na silnych regułach asocjacji odkrytych w zbiorze danych o posiłkach;
- bazujący na zaadaptowaniu metod ukrytego grafu społecznego związanego z preferencjami żywieniowymi, oparty na transakcyjnym zaufaniu do produktów spożywczych;
- oparty na kojarzeniu w par rekomendacji produktów spożywczych;

Bazę danych do badania stanowiło 4800 produktów spożywczych oraz zaewidencjonowane złożone z nich posiłki. Przedmiotowe produkty odpowiadają obiektom z baz danych systemów e-Commerce, badani respondenci odpowiadają użytkownikom a posiłki transakcjom. Stąd też opisane algorytmy mogą znaleźć szersze zastosowanie nie tylko w obszarze badania nawyków żywieniowych, ale również w innych rozwiązaniach e-Commerce.

Podczas badań algorytmu ARS w ramach porównania z innym konkurencyjnym algorytmem, został wykorzystany pierwszy z opisanych w pracy algorytmów, gdyż w najbardziej zbliżony sposób wykorzystuje on zachowania użytkowników w postaci reguł asocjacji wyznaczanych w wyniku działania algorytmu Apriori (Agrawal i in. 1993).

Algorytm ten opiera się na zasadzie, że każda reguła składa się z zestawu poprzednich produktów i pojedynczego następującego po nich produktu. Co odpowiada opisanemu wcześniej modelowi reguł asocjacji $A \rightarrow B$ gdzie $A, B \subseteq O$ (O w tym wypadku to zbiór produktów spożywczych) gdzie $A \cap B = \emptyset$ oraz dodatkowo $|B| = 1$ liczność zbioru B wynosi jeden co oznacza, że następnik jest jednoelementowy (jeden produkt). Algorytm dokonuje rekomendacji na podstawie poprzedników przechowywanych reguł asocjacji i tworzy rekomendacje na podstawie następników ww. reguł (Osadchiy i in. 2018).

Przedmiotowy algorytm w postaci pseudokodu przyjmuje postać:

Dane wejściowe:

AM – zestaw silnych reguł asocjacji

IF – produkty względem których następuje rekomendacja

Dane wyjściowe:

RF – tablica rekomendowanych produktów

Niech:

$rl \in AM$ – pojedyncza reguła asocjacji

$rl.consequent$ – następnik reguły asocjacji rl

$rl.antecedent$ – poprzednik reguły asocjacji rl

$rl.confidence$ – ufność reguły rl

$size(x)$ – liczba elementów zbioru x (wielkość x)

Kroki algorytmu:

1. $RF := \emptyset$
2. dla każdej $rl \in AM \ \& \ rl.consequent \notin IF$
   2.1. $f := rl.consequent$
   2.2. jeżeli istnieje af takie, że $af \in rl.antecedent \ \& \ af \in IF$, to
      2.2.1. jeżeli $f \notin RF$, to $RF[f] := 0$
   2.3. $antc := rl.antecedent$
   2.4. $c := rl.confidence$
   2.5. $intr := size\ (\{af : af \in antc \ \& \ af \in IF\})$
   2.6. $ms := intr^2/(size(antc) * size(IF))$
   2.7. $RF[f] := RF[f] + c * ms$
3. zwróć RF

Algorytm oblicza prawdopodobieństwo rekomendowanego produktu f jako współczynnik ufności reguły c pomnożonej przez podobieństwo między $antc$ i IF (tj. wynik dopasowania ms). Wynik dopasowania ms jest obliczany jako liczba produktów spożywczych, które występują zarówno w IF, jak i $antc$ (tzn. przecinają się) podniesiona do potęgi drugiej i podzielona przez iloczyn wielkości IF i wielkości $antc$. W algorytmie jest wprowadzony wynik dopasowania, aby rekomendacje, które są bardziej podobne do IF, były preferowane. Następnie sumowany jest wynik dopasowania dla każdego produktu f jako pojedynczy element tablicy wyników rekomendacji $RF[f]$.

Teoretyczna złożoność obliczeniowa i pamięciowa przedstawionego algorytmu są stosunkowo proste do oszacowania. Na złożoność pamięciową wpływa przede wszystkim liczność zestawu silnych reguł asocjacji. Czyli można przyjąć, że jest ona klasy O(n), gdzie n jest liczbą reguł asocjacji. Ponadto złożoność obliczeniowa zależy od jednej pętli dla której liczba przebiegów jest uzależniona również od liczności zestawu silnych reguł asocjacji, czyli również jest ona klasy O(n), gdzie n jest liczbą reguł asocjacji. Zatem jest to algorytm liniowy, a funkcja złożoności jest klasy O(n).

5. Algorytm Rekomendacji Sesji

W celu rozwiązania przedstawionego w rozdziale 3.4 problemu badawczego rekomendacji został opracowany autorski algorytm rekomendacji bazujący na sesjach rekomendacji (ARS, ang. Recommendation Algorithm Based On Recommendation Sessions). W celu przedstawienia idei algorytmu została wykorzystana teoria grafów i sieci. Z tego też powodu można zaliczyć go do grupy algorytmów rekomendacji bazujących na grafach (Malinowski 2021a). Sposób wykorzystania modelu, sesji powoduje, że jednocześnie metoda ARS może być potraktowana jako metoda należąca do grupy technik rekomendacji bazujących na sesjach (ang. Session-based recommendation).

5.1. Algorytm rekomendacji bazujący na sesjach rekomendacji

W swojej nazwie wykorzystuje on pojęcie sesji. Wynika to z faktu, że jako dane podczas funkcjonowania algorytmu w systemie e-Commerce wykorzystywane są sesje związane z zachowaniami użytkowników. Są one jednak ograniczone tylko do informacji związanej z pierwszym pojawieniem się zachowania typu „kliknięcie" w obiekt podczas sesji. Odzwierciedla to fakt samego zainteresowanie obiektem przez użytkownika, a nie intensywności tego zainteresowania na przykład poprzez powtórne „kliknięcia" w obiekt.

Ponadto algorytm został tak skonstruowany, że do jego funkcjonowania mogą być wykorzystane atrybuty obiektów. Przyjmują one wówczas postać modelu sesji rekomendacji zdefiniowanego w (5.2).

5.1.1. Podstawowe pojęcia grafów

Teoria grafów i sieci znajduje zastosowanie w obszarze rekomendacji, przede wszystkim na poziomie modeli danych niezbędnych do funkcjonowania algorytmów rekomendacji, na podstawie których budowane są systemy rekomendacji implementowane w rozwiązaniach informatycznych. W modelach tych wykorzystywane są głównie grafy heterogoniczne z dwoma podstawowymi typami węzłów w postaci obiektów i użytkowników, które mogą być rozbudowywane o inne

typy w zależności od przyjętego modelu. Szeroko to zagadnienie zostało opisane w rozdziale 3.2.4.

W obszarze aparatu matematycznego pojęcie grafu bazuje na następującej definicji (Wojciechowski i Pieńkosz 2013):

Grafem jest uporządkowana para:

$$G =< N, E > \qquad (5.1)$$

gdzie:

N – zbiór wierzchołków

E – rodzina krawędzi (rodzina w odróżnieniu od zbioru może zawierać elementy powtarzające się)

ponadto:

$N \neq \emptyset$ - graf ma co najmniej jeden wierzchołek

$N \cap E = \emptyset$ - zbiór wierzchołków i rodzina krawędzi nie mają części wspólnej.

Krawędzią grafu $e \in E$ jest uporządkowana para $e =< n_1, n_2 >$ gdzie $n_1, n_2 \in N$.

Jeżeli rodzina E zawiera powtarzające się elementy, to takie krawędzie nazywamy równoległymi lub wielokrotnymi.

Jeżeli $< n_1, n_2 > \in E$ i $< n_2, n_1 > \in E$, to taka para jest nazywana krawędzią niezorientowaną (nieskierowaną) lub wprost krawędzią.

Jeżeli $< n_1, n_2 > \in E$ i $< n_2, n_1 > \notin E$, to krawędź nazywamy krawędzią zorientowaną (skierowaną) lub łukiem.

Jeżeli łuk łączy wierzchołek n_1 z n_2, to n_1 jest poprzednikiem wierzchołka n_2, a n_2 jest następnikiem n_1.

Klasyfikacja grafów

Graf niezorientowany (nieskierowany) – graf niezawierający łuków.

Graf zorientowany (skierowany) – graf niezawierający krawędzi niezorientowanych.

Multigraf – graf z krawędziami równoległymi.

Unigraf – graf bez krawędzi równoległych.

Parametry

$d(n)$ – stopień wierzchołka n dla grafu niezorientowanego to liczba krawędzi incydentnych (sąsiadujących) z wierzchołkiem n.

$deg_{out}(n)$ – stopień wyjściowy wierzchołka n dla grafu skierowanego to liczba łuków incydentnych (sąsiadujących) z wierzchołkiem n i zorientowanych od tego wierzchołka.

$deg_{in}(n)$ – stopień wejściowy wierzchołka n dla grafu skierowanego to liczba łuków incydentnych (sąsiadujących) z wierzchołkiem n i zorientowanych do tego wierzchołka.

5.1.2. Model matematyczny danych

W rozwiązaniu proponuje się budowę modelu danych opartą na bazie grafu rekomendacji. W obszarze aparatu matematycznego pojęcie to jest zdefiniowane na podstawie dwudzielnego unigrafu skierowanego G zwanego grafem sesji rekomendacji takiego, że graf zdefiniowany w (5.1) w postaci: $G =< N, E >$ przyjmuje jako (Malinowski 2020):

$N = J \cup O$ – zbiór wierzchołków

$E \subset J \times O$ – zbiór łuków (krawędzi skierowanych)

gdzie:

O – zbiór obiektów, gdzie obiekt ($o \in O$) może stanowić:

- towar w sklepie internetowym;
- film w wypożyczalni na żądanie;
- pracownik w serwisie związanym z zatrudnieniem;
- artykuł prasowy w serwisie informacyjnym;
- osoba w serwisie społecznościowym.

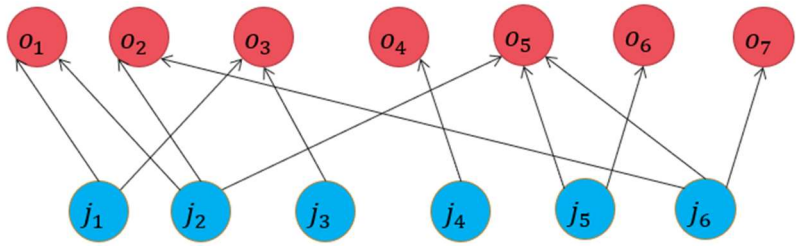
J – zbiór jąder, gdzie jądro ($j \in J$) może stanowić:

- kategoria produktów – jeden z podzbiorów produktów mających wspólne cechy;
- zamówienie – wynik działań klienta w sklepie zakończony zakupem;

- lista życzeń klienta – podzbiór produktów sklepu związanych z klientem, wynikający z jego preferencji;
- ekspert – podzbiór produktów wskazanych przez specjalistę dziedzinowego;
- identyfikator odwiedzin strony WWW – unikalny klucz nadawany odwiedzinom (sesji) użytkownika w serwisie WWW.
- osoba – struktura informatyczna identyfikująca i opisująca użytkownika w systemie.

Ze względu na przyjętą i omówioną w dalszej części interpretację modelu i jego składowych, zakłada się spełnienie następujących ograniczeń:

- $J \cap O = \emptyset$ - żadne jądro nie może być obiektem i żaden obiekt nie może być jądrem
- $\forall_{j \in J}\ \exists_{o \in O}\ \exists_{e \in E}\ e =< j, o >$ – każde jądro musi być związane z co najmniej jednym obiektem
- $\forall_{o \in O}\ \exists_{j \in J}\ \exists_{e \in E}\ e =< j, o >$ – każdy obiekt musi być związany z co najmniej jednym jądrem



Rysunek 5.1. Przykładowy graf sesji rekomendacji G zawierający zbiór obiektów $\{o_1, o_2, o_3 \ldots . o_7\}$ i zbiór jąder $\{j_1, j_2, j_3 \ldots . j_6\}$.
Źródło: opracowanie własne.

W oparciu o funkcje pomocnicze takie jak:

- $\Gamma: J \to 2^O$ taką, że $\Gamma(j) = \{o \epsilon O: \langle j, o \rangle \epsilon E\ \}$
  gdzie wartość $\Gamma(j)$ to zbiór wszystkich następników jądra j w grafie G
- $\Gamma^{-1}: O \to 2^J$ taką, że $\Gamma^{-1}(o) = \{j \epsilon J: \langle j, o \rangle \epsilon E\ \}$
  gdzie wartość $\Gamma^{-1}(o)$ to zbiór wszystkich poprzedników obiektu o w grafie G

pojedynczą sesję rekomendacji sr można przedstawić jako podgraf grafu G taki, że:

$$sr =< N', E' > \qquad (5.2)$$

gdzie:

$N' = \{j\} \cup O'$ – zbiór wierzchołków sesji rekomendacji

$j \in J$ – jądro sesji rekomendacji

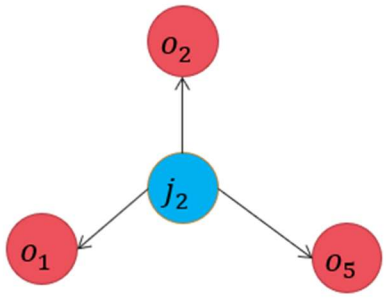
$O' = \Gamma(j)$ – zbiór obiektów sesji rekomendacji (następników jądra)

$E' = \{< j, o > \epsilon E : o \epsilon \Gamma(j)\}$ – zbiór łuków sesji rekomendacji

Zakłada się spełnienie następujących ograniczeń:

- $\forall_{o \in O'} \exists_{e \in E'} e =< j, o >$ – każdy obiekt sesji powiązany jest z jądrem
- $\neg \exists_{o \in O'} \exists_{e \in E'} e =< j, o >$ – nie istnieje obiekt powiązany z jądrem niebędący elementem sesji

Oznacza to, że sesja rekomendacji złożona jest tylko z jednego jądra i wszystkich łuków oraz obiektów z nim związanych.



Rysunek 5.2. Przykład pojedynczej sesji rekomendacji zawierającej zbiór obiektów $\{o_1, o_2, o_3\}$ *oraz jądro* $\{j_2\}$.
Źródło: opracowanie własne.

Jako SR określamy zbiór sesji rekomendacji, gdzie element tego zbioru ($sr \in SR$) może stanowić:

- kategoria produktów wraz z jej produktami;
- zamówienie wraz z pozycjami;
- lista życzeń klienta wraz z elementami;
- ekspert wraz ze swoimi ocenami;
- identyfikator odwiedzin strony WWW wraz z odwiedzonymi stronami;
- osoba wraz ze znajomymi.

Ponadto, ze względu na swoje właściwości fizyczne i podobieństwa, oznaczmy przez $\mathbb{K} = \{K_1, K_2, K_3 \ldots . K_n\}$ zbiór różnych typów jąder sesji rekomendacji. Każdemu $K_k \in \mathbb{K}$

odpowiada $J_k \subset J$ podzbiór jąder tego samego typu, innymi słowy o tych samych właściwościach funkcjonalnych np.: kategorie produktów, zamówienia, itd.…

Czyli:

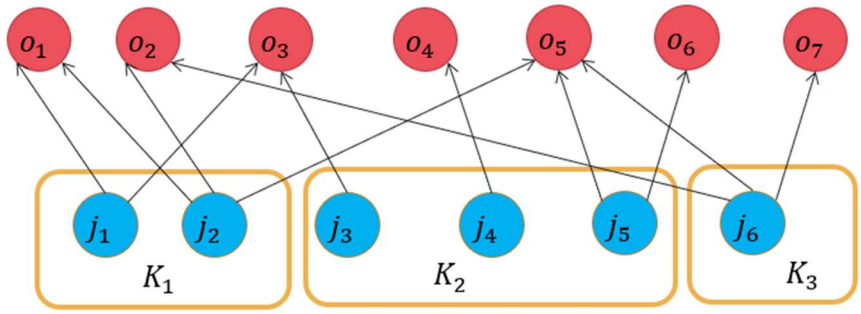
$K_k \subset J$ – klasa jest podzbiorem zbioru jąder

Gdzie klasę ($K_k \subset J$) jąder sesji mogą stanowić:

- kategorie produktów;
- zamówienia;
- listy życzeń klientów;
- eksperci;
- identyfikatory odwiedzin serwisu WWW;
- osoby.

Ponadto zakłada się spełnienie następujących ograniczeń:

- $\forall_{i \neq j}\ K_i \cap K_j = \emptyset$ – żadne jądro nie występuje w dwóch i więcej klasach
- $\forall_{j \in J}\ \exists_i\ j \in K_i$ – każde jądro należy do jakieś klasy



Rysunek 5.3. Przykład grupowania jąder $\{j_1, j_2, j_3 \ldots j_6\}$ w klasach $\{K_1, K_2, K_3\}$.
Źródło: opracowanie własne.

Zasadniczo klasy jąder można podzielić na następujące typy:

- behawioralne – jądra powstające w wyniku zachowań użytkowników. Są one zmienne w czasie jak np.: odwiedziny, zakupy lub lista życzeń;
- statyczne – jądra związane z atrybutami obiektów. Są one niezmienne w czasie jak np.: rozmiar, kolor lub przynależność do kategorii;

- mieszane – jądra powstające w wyniku oddziaływania otoczenia na obiekty, ale słabo zmienne w czasie jak np.: wskazania ekspertów lub zewnętrzne rankingi.

5.1.3. Kroki algorytmu

Algorytm ARS składa się z określonych danych wejściowych, wyjściowych i kroków (Malinowski 2020):

Dane wejściowe:

G - graf sesji rekomendacji

m - obiekt (węzeł grafu), do którego mają zostać dowiązane rekomendacje

Dane wyjściowe:

R_m - wektor rekomendacji dla obiektu m (im niższa pozycja w wektorze, tym lepsza rekomendacja dla obiektu m)

Kroki:

S01: Podanie danych wejściowych

Przykład:

$G = < N, E >$

$m = o_3$

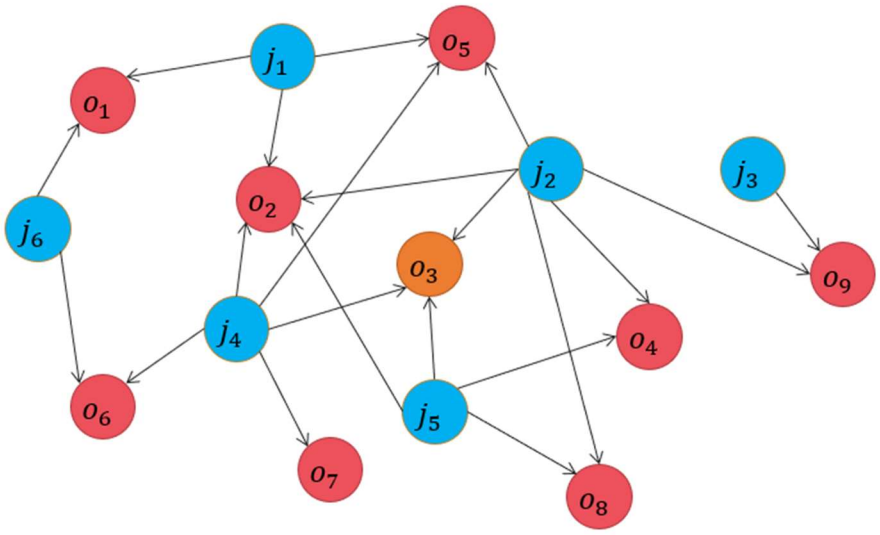
gdzie:

$N = J \cup O$

$O = \{o_1, o_2, o_3 \ldots . o_9\}$

$J = \{j_1, j_2, j_3 \ldots . j_6\}$

$E = \{< j_1, o_1 >, < j_1, o_2 >, < j_1, o_5 >, < j_2, o_2 >, < j_2, o_3 >, < j_2, o_4 >, \ldots, < j_6, o_6 >\}$



Rysunek 5.4. Wejściowy graf sesji rekomendacji G z zaznaczonym wejściowym obiektem (węzłem) m.
Źródło: opracowanie własne.

S02: Budowa podgrafu G'_m grafu G złożonego z węzła m i wszystkich węzłów sąsiadujących z węzłem m oraz łuków pomiędzy nimi a węzłem m

Podgrafu G'_m można zdefiniować ogólnie jako:

$G'_m =< N', E' >$

gdzie:

$N' = \{\mathrm{m}\} \cup J'$

$J' = \Gamma^{-1}(m)$

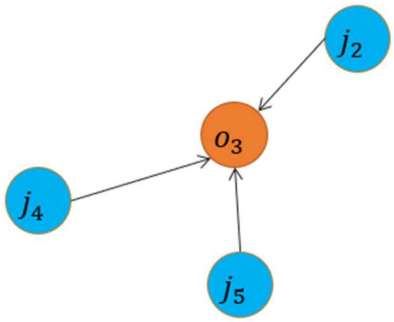
$E' = \{\langle j, m\rangle \epsilon E:\ j \epsilon \Gamma^{-1}(m)\}$

Przykład:

$m = o_3$

$J' = \{j_2, j_4, j_5\}$

$E' = \{< j_2, o_1 >, < j_4, o_1 >, < j_5, o_1 >\}$



Rysunek 5.5. Zbudowany podgraf G'_m grafu G
Źródło: opracowanie własne.

S03: Budowa podgrafu G''_m grafu G złożonego z grafu G'_m i wszystkich węzłów sąsiadujących z węzłami grafu G'_m oraz łuków pomiędzy nimi a węzłami grafu G'_m

Podgrafu G''_m można zdefiniować ogólnie jako:

$G"_m =< N", E" >$

gdzie:

$N" = N' \cup \left(\bigcup_{j \epsilon J'} \Gamma(j)\right)$

$E" = \{\langle j, m\rangle \epsilon E:\ j, m \epsilon \mathrm{N}"\}$

Przykład:

$m = o_3$

gdzie:

$O'' = \{o_3, o_2, o_4, o_5, o_6, o_7, o_8, o_9\}$

$J'' = \{j_2, j_4, j_5\}$

$E'' = \{< j_2, o_1 >, < j_4, o_1 >, < j_5, o_1 >, < j_2, o_4 >, < j_2, o_5 >, < j_2, o_9 >, < j_4, o_2 >,$
$< j_4, o_6 >, < j_4, o_7 >, < j_5, o_2 >, < j_5, o_8 >\}$

S04: Oszacowanie stopni wchodzących dla każdego węzła będącego obiektem podgrafu G''_m

Przykład:

$deg_{in}(o_3) = 3$

$deg_{in}(o_2) = 3$

$deg_{in}(o_4) = 2$

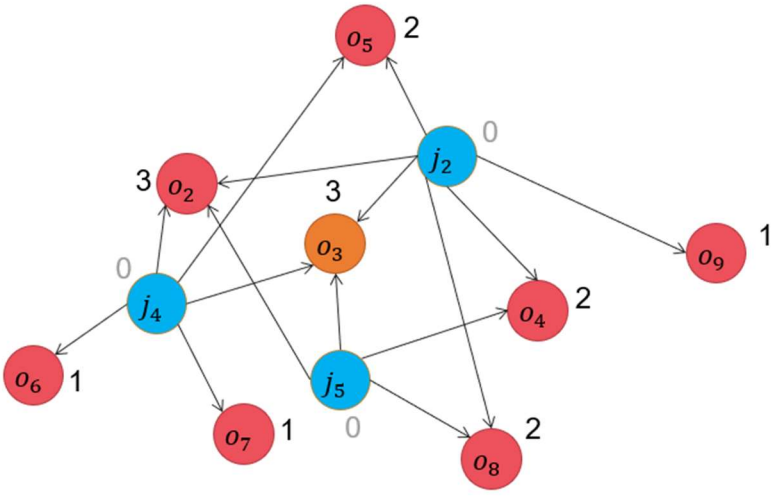
$deg_{in}(o_5) = 2$

$deg_{in}(o_6) = 1$

$deg_{in}(o_7) = 1$

$deg_{in}(o_8) = 2$

$deg_{in}(o_9) = 1$



Rysunek 5.6. Zbudowany podgraf G''_m grafu G z oszacowanymi stopniami wchodzącymi węzłów.
Źródło: opracowanie własne.

S05: Posortowanie malejąco obiektów względem stopnia wchodzącego

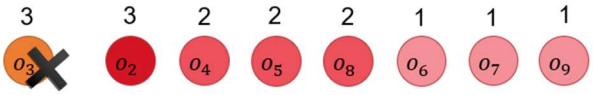
Przykład:

$o_3, o_2, o_4, o_5, o_8, o_6, o_7, o_9$

S06: Zapisanie w wektorze R_m posortowanych obiektów bez obiektu m

Przykład:

$R_m =< o_2, o_4, o_5, o_8, o_6, o_7, o_9 >$



Rysunek 5.7. Posortowane obiekty (węzły) z usuniętym obiektem m.
Źródło: opracowanie własne.

Wektor R_m stanowi dane wyjściowe, czyli wynik działania algorytmu.

Podsumowanie

Algorytm ARS składa się z następujących po sobie kroków:

S01: Podanie danych wejściowych

S02: Budowa podgrafu G'_m grafu G złożonego z węzła m i wszystkich węzłów sąsiadujących z węzłem m oraz łuków pomiędzy nimi a węzłem m

S03: Budowa podgrafu G''_m grafu G złożonego z grafu G'_m i wszystkich węzłów sąsiadujących z węzłami grafu G'_m oraz łuków pomiędzy nimi a węzłami grafu G'_m

S04: Oszacowanie stopni wchodzących dla każdego węzła będącego obiektem podgrafu G''_m

S05: Posortowanie malejąco obiektów względem stopnia wchodzącego

S06: Zapisanie w wektorze R_m posortowanych obiektów bez obiektu m

5.1.4. Oszacowanie złożoności obliczeniowej

Oszacowanie złożoności obliczeniowej algorytmu ARS bazuje na wskazaniu funkcji matematycznej odgrywającej najważniejszą rolę, czyli wpływającej najsilniej na czas lub wielkość pamięci podczas wykonywania algorytmu. Funkcja ta determinuje przynależność do klasy algorytmów i jest najczęściej określana poprzez O (Wroblewski 2015).

Ponadto w celu dokonania oszacowania wykorzystano wiedze na temat złożoności znanych algorytmów realizujących operacje na określonych strukturach (Rowell 2013) (Cormen i in. 2009).

W celu dokonania oszacowania złożoności całego algorytmu dokonano oszacowania poszczególnych kroków, gdzie:

S01: Podanie danych wejściowych.

Czas wykonania jest stały i niezależny od rozmiaru danych wejściowych

Złożoność obliczeniowa: O(1)

S02: Budowa podgrafu G'_m grafu G złożonego z węzła m i wszystkich węzłów sąsiadujących z węzłem m oraz łuków pomiędzy nimi a węzłem m

Polega na przeglądzie wszystkich łuków w celu znalezienia sąsiadujących z obiektem m jąder

Złożoność obliczeniowa: O(|E|)

gdzie:

$|E|$ - liczba łuków grafu G

S03: Budowa podgrafu G''_m grafu G złożonego z grafu G'_m i wszystkich węzłów sąsiadujących z węzłami grafu G'_m oraz łuków pomiędzy nimi a węzłami grafu G'_m

Polega na przeglądzie wszystkich łuków w celu znalezienia sąsiadujących z elementami podgrafu G'_m obiektów

Złożoność obliczeniowa: O(|E|)

S04: Oszacowanie stopni wchodzących dla każdego węzła będącego obiektem podgrafu G''_m

W skrajnym przypadku polega na przeglądzie wszystkich łuków w celu oszacowania stopni wchodzących obiektów podgrafu G''_m

Złożoność obliczeniowa: O(|E|)

S05: Posortowanie malejąco obiektów względem stopnia wchodzącego

W skrajnym przypadku polega na sortowaniu wszystkich obiektów

Złożoność obliczeniowa: O($|O|^2$)

gdzie:

$|O|$ - liczba obiektów grafu G

S06: Zapisanie w wektorze R_m posortowanych obiektów bez obiektu m

Czas wykonania jest stały i niezależny od rozmiaru danych wejściowych

Złożoność obliczeniowa: O(1)

Oszacowanie złożoności obliczeniowej dla całego algorytmu można zapisać w postaci:

O(ARS) = O(S01) +O(S02) +O(S03) +O(S04) +O(S05) +O(S06) =

= O(1)+ O($|E|$) + O($|E|$) + O($|E|$) + O($|O|^2$)+ O(1) =

= 2xO(1)+ 3xO($|E|$) + O($|O|^2$)

co daje oszacowanie złożoności obliczeniowej: O($|E|$) + O($|O|^2$)

Wyrazem najbardziej wpływającym na czas realizacji algorytmu jest wyraz o najwyższej potędze, czyli $|O|^2$, zatem jest to algorytm wielomianowy, a funkcja złożoności jest klasy O(n^2), można powiedzieć, że wielomian rozmiaru problemu jest rzędu n^2.

W praktyce przy założeniu, że liczba łuków $|E|$ jest znacznie większa od liczby obiektów $|O|$ kluczowym staje się wyrażenie O($|E|$).Ostatecznie można przyjąć, że funkcja złożoności jest klasy O(n)

Liczba łuków $|E|$ grafu G zależy od powiązań między parami węzłów złożonych z obiektów ze zboru *O* i jąder ze zbioru J. W dalszej części opracowania na podstawie badań nad algorytmem zostało wykazane, że w praktyce zbiór obiektów *O* przyrasta w bardzo niewielkim stopniu natomiast zbiór jąder J znacznie się rozrasta i to on głównie determinuje wzrost liczby łuków. W związku z powyższym można przyjąć górne oszacowanie złożoności obliczeniowej w postaci O($|J|$) gdzie $|J|$ to liczba jąder grafu G. Złożoność ta może się zmienić w zależności od implementacji algorytmu w systemie e-Commerce.

6. Implementacja algorytmów

W celu realizacji badań nad algorytmem ARS i konkurencyjnym rozwiązaniem zostały one zaimplementowane w funkcjonującym systemie informatycznym e-Commerce. Dało to możliwość gromadzenia informacji na temat funkcjonowania w czasie rzeczywistym algorytmów i podstawę do analizy podstawowych charakterystyk powstałych w wyniku implementacji systemów rekomendacji.

6.1. Badanie kroków algorytmu rekomendacji bazującego na sesjach rekomendacji

Celem pierwszego badania nad algorytmem było określenie realizowalności implementacji poszczególnych kroków algorytmu ARS na podstawie rzeczywistych danych z systemu e-Commerce. Dane zostały pozyskane z relacyjnej bazy danych systemu informatycznego klasy e-Commerce w postaci sklepu internetowego AM76[1]. Dane były gromadzone w przedziale czasu od 15 do 20 grudnia 2019r. (Malinowski 2020).

Jako dane do budowy grafu sesji rekomendacji G posłużyły takie informacje jak:

O – zbiór obiektów będący produktami o unikalnych identyfikatorach: idprodukt (*produkt.idprodukt)*

$J = K_1 \cup K_2 -$ zbiór jąder złożony z dwóch klas bazujących na zachowaniach użytkowników zgodnie ze zbiorem B zdefiniowanym w (2.1):

K_1 – kliknięcie w obiekt (produkt) sklepu posiadające unikalny identyfikator: phpsesid (*przeglad_zrodlo.phpsesid)*;

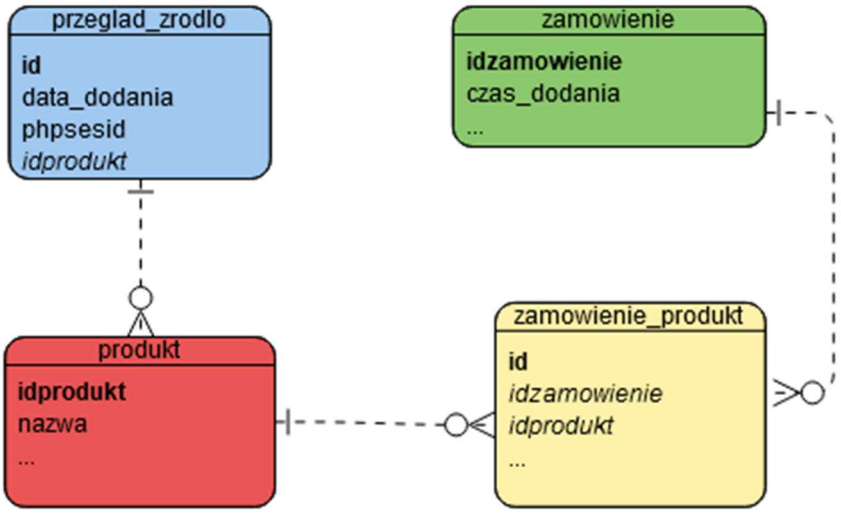
K_2 – zakup obiektu (produktu) posiadający unikalny identyfikator: idzamowienie (*zamowienie.idzamowienie)*;

[1] Implementacja funkcjonuje w czasie rzeczywistym w systemie e-Commerce dostępnym pod adresem elektronicznym http://am76.pl.

$E = E_1 \cup E_2$ – zbiór łuków złożony z:

E_1- odwiedziny produktów: phpsesid -> idprodukt

E_2- pozycje zamówień: idzamowienie -> idprodukt



Rysunek 6.1. Diagram ERD struktury istotnych dla grafu sesji rekomendacji G danych bazy systemu informatycznego AM76.
Źródło: opracowanie własne.

Opisane powyżej dane zostały wyeksportowane w postaci pliku w formacie CSV[2] do bazy grafowej Neo4j. Baza ta pozwoliła na prezentację w postaci grafu G zawartych w relacyjnej bazie MySQL[3] danych systemu informatycznego AM76 i były to:

- 1733 wierzchołki zbioru O (oznaczone kolorem czerwonym);
- 2056 wierzchołki zbioru K_1 (oznaczone kolorem niebieskim);
- 314 wierzchołki zbioru K_2 (oznaczone kolorem zielonym);
- 7459 łuki zbioru E_1;
- 964 łuki zbioru E_2.

[2] CSV to plik tekstowy w określonym formacie, który umożliwia zapisywanie danych w postaci tabeli (https://www.loc.gov/preservation/digital/formats/fdd/fdd000323.shtml)

[3] MySQL to system relacyjnej baz danych. Jest to jedno z najbardziej popularnych rozwiązań tego typu stosowanych w systemach e-Commerce (https://www.mysql.com/)

K_1 K_2

J

phpsesid	data
1624844680	2019-12-15
1601766927	2019-12-15
1372428021	2019-12-15
1630107088	2019-12-15
1539705806	2019-12-15

idzamowienie	data
89031	2019-12-15
89032	2019-12-15
89033	2019-12-15
89037	2019-12-15
89048	2019-12-15

E

phpsesid	idprodukt
1000180571	3454
1000180571	3492
1000852593	4314
1000852593	4050
1000896693	3046

idzamowienie	idprodukt
89031	4750
89031	4503
89031	5496
89031	4946
89032	3689

O

idprodukt	nazwa
2	Abalone Classic
3	Abalone Travel (edycja polska)
5	Agricola
7	Agricola: Torfowisko
8	Alhambra (edycja polska)

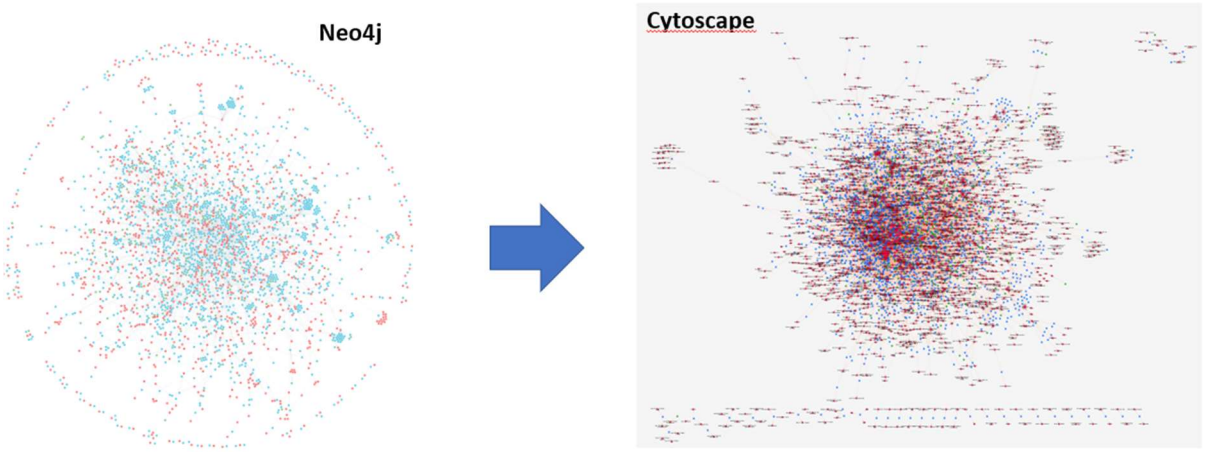
Rysunek 6.2. Przykłady danych dla poszczególnych elementów grafu G (zbiór obiektów O, zbiór jąder J oraz zbiór łuków E).
Źródło: opracowanie własne.

Do badania zostało wykorzystane oprogramowanie o nazwie Cytoscape[4]. Podstawowa wersja ww. oprogramowania została wzbogacona o plugin o nazwie CypherQueries, który to pozwala na bezpośrednie połączenie do bazy danych Neo4j[5] i edycje danych na podstawie język Cypher[6].

[4] Cytoscape to platforma do wizualizacji i analizy złożonych sieci i ich integracji z dowolnymi danymi atrybutowymi (https://cytoscape.org/)

[5] Neo4j to grafowa baza danych, w ramach której można modelować struktury grafowe (https://neo4j.com/)

[6] Cypher to język zapytań do grafu w bazie Neo4j, który pozwala na pobieranie danych z grafu i dokonywanie operacji na nich (https://neo4j.com/developer/cypher/)



Rysunek 6.3. Poglądowy obraz grafu G będącego przedmiotem badania w ramach bazy Neo4J i po migracji do oprogramowania analizy struktur grafowych Cytoscape. Źródło: opracowanie własne.

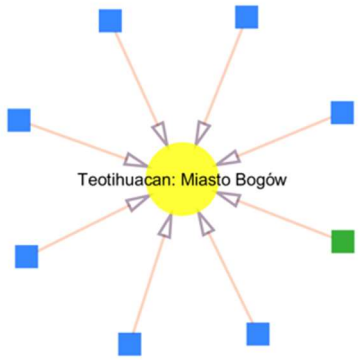
W ramach badania zostały zrealizowane kroki (S01-S06) algorytmu ARS w kolejności:

S01:

Grafem wejściowym G był graf rekomendacji sesji zaimportowany do bazy Neo4J i obiektem m był obiekt (węzeł) o identyfikatorze (idprodukt) 4537 posiadający atrybut nazwa o wartości "Teotihuacan: Miasto Bogów".

S02:

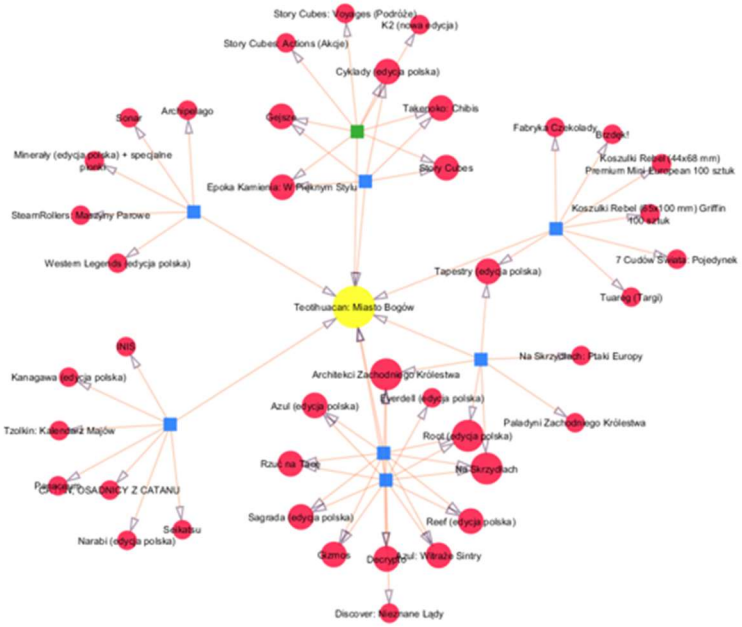
Zbudowany został podgraf G'_m grafu G. Podgraf składał się z węzła m i węzłów sąsiadujących z nim oraz łączących ich łuków. Węzłami sąsiadującymi były jądra sesji, w których występował węzeł m.



Rysunek 6.4. Zbudowany podgraf G'_m grafu G.
Źródło: opracowanie własne.

S03:

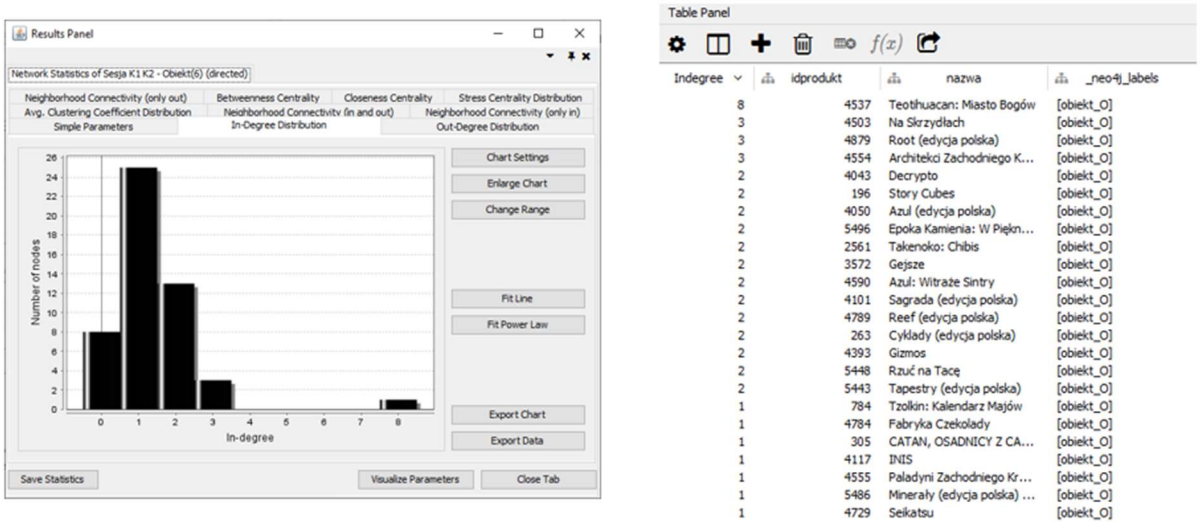
Zbudowany został podgraf G''_m grafu G. Do podgrafu G'_m zostały dodane węzły będące obiektami powiązanymi z jądrami sesji występujących w grafie.



Rysunek 6.5. Zbudowany podgraf G''_m grafu G.
Źródło: opracowanie własne.

S04 i S05:

Z wykorzystaniem modułu analitycznego oprogramowania Cytoscape zostały oszacowane stopnie wchodzące dla każdego węzła podgrafu G''_m i następnie zostały posortowane w kolejności malejącej.



Rysunek 6.6. Wynik sortowania stopni wchodzących grafu G''_m.
Źródło: opracowanie własne.

S06:

Ostatecznie wynik sortowania został zapisany w pliku CSV z usuniętym węzłem m o identyfikatorze (idprodukt) 4537. Plik ten stanowił fizyczną reprezentację wektora wynikowego R_m algorytmu ARS.

idprodukt	Indegree	nazwa	neo4j_labels
~~4537~~	~~8~~	~~Teotihuacan: Miasto Bogów~~	~~obiekt_O~~
4503	3	Na Skrzydłach	obiekt_O
4554	3	Architekci Zachodniego Królestwa	obiekt_O
4879	3	Root (edycja polska)	obiekt_O
3572	2	Gejsze	obiekt_O
5443	2	Tapestry (edycja polska)	obiekt_O
5448	2	Rzuć na Tacę	obiekt_O
196	2	Story Cubes	obiekt_O
4789	2	Reef (edycja polska)	obiekt_O
4043	2	Decrypto	obiekt_O
263	2	Cyklady (edycja polska)	obiekt_O
2561	2	Takenoko: Chibis	obiekt_O
4050	2	Azul (edycja polska)	obiekt_O
5496	2	Epoka Kamienia: W Pięknym Stylu	obiekt_O
4590	2	Azul: Witraże Sintry	obiekt_O
4101	2	Sagrada (edycja polska)	obiekt_O
4393	2	Gizmos	obiekt_O
1466	1	Archipelago	obiekt_O

Rysunek 6.7. Fizyczna reprezentacja grafu R_m będącego wynikiem działania algorytmu ARS.
Źródło: opracowanie własne.

Podsumowanie

Przedstawione badanie wykazało realizowalność poszczególnych kroków algorytmu ARS na podstawie rzeczywistych danych z systemu e-Commerce. W ramach badania były brane pod uwagę różne obiekty m względem których były szacowane rekomendacje i ich wyniki były zbieżne z przedstawionym powyżej przykładem. W ramach badania powstał graf złożony z 4103 węzłów i 8423 łuków, który można zaliczyć do klasy małych grafów w obszarze danych w systemach e-Commerce. Dodatkowo w wyniku badania, dzięki prezentacji grafu w bazie Neo4j, okazało się, że graf rekomendacji sesji G jest grafem niespójnym.

W ramach badania nie były rozpatrywane cechy wydajnościowe algorytmu w szczególności związane z czasem generowania rekomendacji, który to wpływa na opisany wcześniej czas odpowiedzi i czasem generowania grafu rekomendacji sesji G, które wynikają ze złożoności obliczeniowej rozwiązania. Charakterystyki te będą badane w ramach kolejnych etapów pracy w rozdziale 7 na podstawie zaimplementowanego algorytmu w działającym systemie e-Commerce.

6.2. Implementacja algorytmu rekomendacji bazującego na sesjach rekomendacji

Analogicznie, jak podczas pierwszego badania kroków algorytmu, implementacja została przeprowadzona w ramach rozwiązania e-Commerce w postaci platformy sklepu internetowego AM76. Jako narzędzie implementacji algorytmu przyjęty został standard SQL[7] funkcjonujący w ramach systemu zarządzania bazą danych MySQL wykorzystanego w systemie informatycznym AM76. Za tym wyborem przemawia fakt bardzo dużej popularności systemu zarządzania bazą danych MySQL w rozwiązaniach e-Commerce, która to została przedstawiona w tabeli 6.1.

[7] SQL to strukturalny oraz deklaratywny język zapytań używany do tworzenia, modyfikowania relacyjnych baz danych oraz do umieszczania i pobierania danych z tych baz.

Pozycja	DBMS	Model bazy danych	Punkty
1.	Oracle	Relational, Multi-model	1251.32
2.	**MySQL**	**Relational, Multi-model**	**1198.23**
3.	Microsoft SQL Server	Relational, Multi-model	933.78
4.	PostgreSQL	Relational, Multi-model	616.93
5.	MongoDB	Document, Multi-model	485.66
6.	Redis	Key-value, Multi-model	176.76
7.	IBM Db2	Relational, Multi-model	162.15
8.	Elasticsearch	Search engine, Multi-model	159.95
9.	Microsoft Access	Relational	135.43
10.	SQLite	Relational	132.18
11.	Cassandra	Wide column	122.14
12.	MariaDB	Relational, Multi-model	108.31
13.	Splunk	Search engine	95.36
14.	Snowflake	Relational	86.23
15.	Microsoft Azure SQL Database	Relational, Multi-model	84.68
16.	Amazon DynamoDB	Multi-model	81.80
17.	Hive	Relational	81.22
18.	Teradata	Relational, Multi-model	68.85
19.	**Neo4j**	**Graph**	**59.67**
20.	Solr	Search engine, Multi-model	59.05

Tabela 6.1 Wyniki rankingu (marzec 2022) prowadzonego przez DB-Engines pokazują różnicę pomiędzy najpopularniejszymi systemami relacyjnym (Oracle i MySQL) a najpopularniejszym systemem grafowym (Neo4j).
Źródło: https://db-engines.com/en/ranking (data dostęp: 24.03.2022).

Głównymi problemami przy implementacji algorytmu w systemie rekomendacji są (Malinowski 2021b):

- budowa grafu G w strukturach bazodanowych docelowego systemu informatycznego e-Commerce;
- implementacja kroków algorytmu z wykorzystaniem mechanizmów informatycznych danego rozwiązania e-Commerce.

6.2.1. Implementacja struktur grafowych

Znacznie wyższy poziom wykorzystania relacyjnych baz danych w stosunku do baz grafowych w obszarze przetwarzania danych powoduje, że fizycznie wykorzystywane są do reprezentacji grafów macierze i listy, które to są możliwe do implementacji w relacyjnych bazach danych. Trend ten wraz z rozwojem baz grafowych, takich jak na przykład Neo4j, najprawdopodobniej będzie się zmieniał, co pozwoli na efektywne

wykorzystanie grafów szczególnie na poziomie implementacji systemów rekomendacji oraz ich bieżącego funkcjonowania (Malinowski 2021a).

Grafy są strukturą abstrakcyjną, niemającą swojego bezpośredniego odzwierciedlenia w środowisku. Ich elementy, czyli wierzchołki i krawędzie, mogą mieć różne liczności. Ponadto model matematyczny grafów nie jest w naturalny sposób odzwierciedlany w organizacji pamięci maszyn liczących (komputerów) (Horzyk 2021). Implementując grafy, wykorzystuje się inne dobrze zdefiniowane struktury programistyczne, takie jak tablice i listy. Zasadniczo służą one do przechowywania informacji na temat sąsiednich (incydentnych) wierzchołków lub łączących je krawędzi (Goodrich i Tamassia 2002).

Grafy reprezentuje się oraz implementuje w systemach informatycznych zwykle w postaci (Cormen i in. 2009):

- macierzy sąsiedztwa (ang. adjacency matrix) – jest przedstawiana jako tablica dwuwymiarowa, gdzie indeksy wierszy i kolumn reprezentują numery wierzchołków, a wartości elementów równe 1 oznaczają krawędź łączącą wierzchołki określone numerem wiersza i kolumny;
- macierzy incydencji (ang. incidence matrix) – jest przedstawiana jako tablica dwuwymiarowa, gdzie indeksy wierszy reprezentują numery wierzchołków, a indeksy kolumn oznaczają numery krawędzi. Elementy równe 1 oznaczają krawędzie oznaczone numerami kolumn incydentnych z wierzchołkami oznaczonymi numerami wierszy;
- listy sąsiedztwa (ang. adjacency list) – jest przedstawiana jako lista, gdzie indeksy reprezentują numery wierzchołków, a każdy element tej listy jest listą numerów sąsiednich wierzchołków z numerem wierzchołka będącym w indeksie;
- listy incydencji (ang. incidence list) - jest przedstawiana jako lista, gdzie indeksy reprezentują numery wierzchołków, a każdy element tej listy jest listą numerów krawędzi incydentnych z numerem wierzchołka będącym w indeksie;
- listy krawędzi (ang. list of edges) – jest przedstawiana jako lista par numerów wierzchołków dla każdej krawędzi.

6.2.2. Budowa grafu G

O wyborze danej formy reprezentacji grafu decydują ewentualne plusy i minusy funkcjonalne danej reprezentacji lub złożoność pamięciowa (Cormen i in. 2009).

Reprezentacja grafu	Złożoność pamięciowa
macierz sąsiedztwa	O(|N|^2)
macierz incydencji	O(|N|*|E|)
lista sąsiedztwa	O(|N|+|E|)
lista incydencji	O(|N|+|E|)
lista krawędzi	O(|E|)

Tabela 6.2. Porównanie reprezentacji grafowych w pamięci komputerów. Gdzie $|N|$ to liczność zbioru wierzchołków, $|E|$ to liczność zbioru krawędzi.
Źródło: opracowanie własne.

W przypadku implementacji w ramach pracy została wybrana reprezentacja w postaci listy krawędzi wynika to z następujących faktów:

- prostej reprezentacji grafu w relacyjnej bazie danych;
- dobrych właściwości w obszarze złożoności pamięciowej;
- łatwej konwersji z danych transakcyjnych systemu e-Commerce do listy krawędzi.

Struktura tabeli w relacyjnej bazie odzwierciedlająca graf rekomendacji sesji G przyjęła postać listy krawędzi zgodnie z przedstawionym poniżej diagramem ERD.

Rysunek 6.8. Diagram ERD struktury tabeli graf_g reprezentującej łuki grafu rekomendacji sesji G w postaci listy krawędzi typu jądro (klasa)->obiekt.
Źródło: opracowanie własne.

Do implementacji grafu rekomendacji sesji G w docelowym środowisku e- Commerce zostało wykorzystanych sześć klas jąder, z czego trzy są związane z zachowaniami użytkowników i są analogiczne do zdefiniowanych w ramach danych transakcyjnych Analizy Asocjacji oraz trzy związane z atrybutami obiektów i są to:

K_1 – zakup obiektu (zachowanie);

K_2 – kliknięcie w obiekt (zachowanie);

K_3 – kategoria obiektu (atrybut);

K_4 – seria obiektu (atrybut);

K_5 – wyróżnienie (przez użytkownika) obiektu (zachowanie);

K_6 – polecanie (przez eksperta) obiektu (atrybut).

Graf G budowany jest co godzinę na podstawie bieżących danych systemu informatycznego AM76. Powstaje on w wyniku działania sekwencji zapytań SQL w postaci:

Zapytanie 1 – usunięcie wcześniejszej implementacji grafu G

```
TRUNCATE TABLE graf_g;
```

Zapytanie 2 – budowa nowej wersji grafu G

```
INSERT INTO graf_g SELECT * FROM (
# krawędzie związane z jądrami klasy K1
SELECT zamowienie_produkt.idzamowienie AS jadro, zamowienie_produkt.idprodukt AS obiekt,
1 AS klasa FROM zamowienie_produkt
INNER JOIN zamowienie ON zamowienie.id=zamowienie_produkt.idzamowienie
INNER JOIN produkt ON produkt.id=zamowienie_produkt.idprodukt
UNION
# krawędzie związane z jądrami klasy K2
SELECT produkt_statystyka.phpsessid AS jadro, produkt_statystyka.idprodukt AS obiekt,
2 AS klasa FROM produkt_statystyka
INNER JOIN produkt ON produkt.id=produkt_statystyka.idprodukt
UNION
# krawędzie związane z jądrami klasy K3
SELECT kategoria_produkt.idkategoria AS jadro, kategoria_produkt.idprodukt AS obiekt,
3 AS klasa FROM kategoria_produkt
INNER JOIN produkt ON produkt.id=kategoria_produkt.idprodukt
UNION
```

krawędzie związane z jądrami klasy K4

SELECT idseria AS jadro, id AS obiekt,
4 AS klasa FROM produkt

WHERE product.idseria>0

UNION

krawędzie związane z jądrami klasy K5

SELECT lista_zyczen.idkontrahent AS jadro, lista_zyczen.idprodukt AS obiekt,

5 AS klasa FROM lista_zyczen

INNER JOIN produkt ON produkt.id=lista_zyczen.idprodukt

UNION

krawędzie związane z jądrami klasy K6

SELECT 900000 AS jadro, produkt.id AS obiekt,

6 AS klasa FROM produkt

WHERE produkt.HIT='tak'

zamknięcie zapytania

) AS graph_g_view;

Zapytanie 3 – optymalizacja danych grafu G

OPTIMIZE TABLE graph_g;

Podsumowanie

Elementy, z których zbudowany jest graf rekomendacji sesji G w szczególności klasy jąder, wynikają z właściwości i możliwości danego systemu e-Commerce. Jak pokazano powyżej są one uniwersalne. Łączą w sobie zarówno zachowania użytkowników, jak i atrybuty obiektów.

6.2.3. Implementacja kroków algorytmu

Na bazie zbudowanego grafu rekomendacji sesji G w postaci tabeli *graf_g* poszczególne kroki algorytmu ARS przyjmują postać:

S01: Podanie danych wejściowych

m – identyfikator obiektu, do którego mają zostać dowiązane rekomendacje. Identyfikator pozyskiwany jest na podstawie zachowania użytkownika $'kliknięcie'$ powiązanego ze stroną WWW produktu

G – wykorzystanie tabeli *graf_g*

S02: Budowa podgrafu G'_m grafu G złożonego z węzła m i wszystkich węzłów sąsiadujących z węzłem m oraz łuków pomiędzy nimi a węzłem m

Wykonanie zapytania SQL w postaci:

SELECT jadro, obiekt FROM graf_g WHERE obiekt=m

S03: Budowa podgrafu G''_m grafu G złożonego z grafu G'_m i wszystkich węzłów sąsiadujących z węzłami grafu G'_m oraz łuków pomiędzy nimi a węzłami grafu G'_m

Wykonanie zapytania SQL w postaci:

SELECT jadro, obiekt FROM graf_g

WHERE jadro IN (SELECT jadro FROM graf_g WHERE obiekt=m)

S04: Oszacowanie stopni wchodzących dla każdego węzła będącego obiektem podgrafu G''_m

Wykonanie zapytania SQL w postaci:

SELECT obiekt, count(*) AS stopien_in FROM graf_g

WHERE jadro IN (SELECT jadro FROM graf_g WHERE obiekt=m)

S05: Posortowanie malejąco obiektów względem stopnia wchodzącego

Do zapytania SQL z S04 zostaje dodany kod SQL w postaci:

(…) ORDER BY stopien_in DESC

S06: Zapisanie w wektorze R_m posortowanych obiektów bez obiektu m

Do zapytania SQL z S05 zostaje dodany warunek SQL w postaci:

(…) WHERE obiekt<>m

oraz zostaje zwrócony do systemu e-Commerce wywołującego algorytm ARS wynik powstały w na bazie kroków od S01 do S06

Kroki algorytmu od S02 do S06 można zapisać w ramach jednego zapytania SQL w postaci:

```
SELECT obiekt, count(*) AS stopien_in FROM graf_g
WHERE obiekt<>m
AND jadro IN (SELECT jadro FROM graf_g WHERE obiekt=m)
GROUP BY obiekt
ORDER BY stopien_in DESC
```

Powyższe zapytanie SQL integruje w sobie kroki algorytmu ARS i zwraca wynik jego działania.

Podsumowanie

Rezultatem implementacji algorytmu ARS z wykorzystaniem metody jego budowy na podstawie standardu SQL w ramach relacyjnej bazy danych, jest w pełni funkcjonalny system rekomendacji możliwy do zaadaptowania w różnego rodzaju systemach informatycznych z obszaru e-Commerce.

6.2.4. Oszacowanie złożoności obliczeniowej implementacji

Implementacja algorytmu ARS i budowa grafu rekomendacji sesji G na bazie standardu SQL i na podstawie relacyjnego systemu zarządzania bazą danych MySQL wymaga ponownego podejścia do kwestii oszacowania złożoności obliczeniowej zaimplementowanego algorytmu.

Graf G reprezentuje lista krawędzi zapisana w tabeli *graf_g,* która przechowuje informacje o $|E|$ krawędziach grafu G w postaci rekordów bazy danych o strukturze kolumn *{jadro, obiekt, klasa}.* Każda kolumna tabeli *graf_g* jest indeksowana. Indeksy te, w ramach systemu zarządzania bazą danych MySQL, bazują na strukturze klasy B-drzewa (drzewo binarne, ang. B-tree) (Oracle 2022a).

Struktura B-drzewa charakteryzuje się następującymi oszacowaniami złożoności obliczeniowej dla operacji bazodanowych (Niklaus 2002):

- wyszukiwanie (przegląd) – złożoność obliczeniowa O(log n);
- dodawanie – złożoność obliczeniowa O(log n);

- usuwanie – złożoność obliczeniowa O(log n)**.**

Sortowanie odbywa się w oparciu o zaimplementowany w systemie bazy danych algorytm QuickSort (Oracle 2022b) dla którego złożoność obliczeniowa jest klasy $O(n^2)$) (Wroblewski 2015).

W oparciu o powyższe informacje, ponowne oszacowanie złożoności obliczeniowej kroków algorytmu przyjmuje postać:

S01: O(1) - czas wykonania jest stały i niezależny od rozmiaru danych wejściowych;

S02: O(log|E|) - przegląd wszystkich łuków w celu znalezienia sąsiadujących z obiektem m jąder;

S03: O(log|E|) - przegląd wszystkich łuków w celu znalezienia sąsiadujących wierzchołków z węzłami podgrafu G'_m ;

S04: O(log|E|) - przegląd wszystkich łuków w celu oszacowania stopni wchodzących podgrafu G''_m ;

S04: O(log|E|) - przegląd wszystkich łuków w celu oszacowania stopni wchodzących podgrafu G''_m ;

S05: O($|O|^2$) - sortowanie węzłów;

S05: O(1) - czas wykonania jest stały i niezależny od rozmiaru danych wejściowych.

Na bazie złożoności obliczeniowej poszczególnych kroków oszacowanie złożoności obliczeniowej dla całego algorytmu przyjmuje postać: O(log|E|) + O($|O|^2$)

Analogicznie, jak przy wcześniejszym oszacowaniu, wyrazem o najwyższej potędze jest $|O|^2$, zatem jest to algorytm wielomianowy, a funkcja złożoności jest klasy $O(n^2)$, można powiedzieć, że wielomian rozmiaru problemu jest rzędu n^2.

Jednakże, przy założeniu, że liczba łuków |E| jest znacznie większa od liczby obiektów |O| oraz w oparciu o obserwację, że w praktyce zbiór obiektów *O* przyrasta w bardzo niewielkim stopniu. Natomiast zbiór jąder J znacznie wzrasta, to on głównie determinuje wzrost liczby łuków, w efekcie można przyjąć oszacowanie złożoności obliczeniowej, w przypadku ww. implementacji, w postaci O(log |J|).

6.3. Implementacja algorytmu bazującego na regułach asocjacji

Głównymi problemami przy implementacji algorytmu bazującego na regułach asocjacji są:

- ekstrakcja danych transakcyjnych z bazy danych docelowego systemu informatycznego e-Commerce;
- implementacja kroków algorytmu odkrywania reguł asocjacji;
- implementacja kroków algorytmu rekomendacji.

Tak jak implementacja algorytmu ARS, tak ww. implementacja zostały przeprowadzona na bazie rozwiązania e-Commerce w postaci platformy sklepu internetowego AM76. W tym wypadku jako narzędzia implementacji oprócz skryptów SQL użyto języka skryptowego PHP[8] funkcjonującego w ramach obszaru aplikacyjnego systemu informatycznego AM76. Ocenia się, że 77,5% wszystkich stron internetowych[9] wykorzystuje elementy zbudowane na bazie tego języka.

6.3.1. Ekstrakcja danych transakcyjnych

Do ekstrakcji danych transakcyjnych w docelowym środowisku e-Commerce zostały wykorzystane grupy informacji związane z zachowaniami użytkowników takie jak:

- zakup obiektu;
- kliknięcie w obiekt;
- wyróżnienie (przez użytkownika) obiektu;

Strukturę tabeli w relacyjnej bazie odzwierciedlającą dane zgromadzone podczas ekstrakcji przedstawia poniższy diagram ERD.

[8] PHP to skryptowy język programowania zaprojektowany do generowania stron internetowych i budowania aplikacji webowych (https://www.php.net/)

[9] W3Techs - World Wide Web Technology Surveys https://w3techs.com/technologies/details/pl-php [dostęp: 29-04-2022]

Rysunek 6.9. Diagram ERD struktury tabeli transakcja zapis transakcji w postaci id(transakcji)->obiekt.
Źródło: opracowanie własne.

Ekstrakcja transakcji następuje co godzinę na podstawie bieżących danych systemu informatycznego AM76. Powstaje ona w wyniku działania sekwencji zapytań SQL w postaci:

Zapytanie 1 – usunięcie wcześniejszych transakcji

```
TRUNCATE TABLE transakcja;
```

Zapytanie 2 – budowa nowej wersji ekstrakcji transakcji

```
INSERT INTO transakcja SELECT * FROM (
# zakupy
SELECT zamowienie_produkt.idzamowienie AS id, zamowienie_produkt.idprodukt AS obiekt
FROM zamowienie_produkt
INNER JOIN zamowienie ON zamowienie.id=zamowienie_produkt.idzamowienie
INNER JOIN produkt ON produkt.id=zamowienie_produkt.idprodukt
UNION
# kliknięcie w obiekt
SELECT produkt_statystyka.phpsessid AS id, produkt_statystyka.idprodukt AS obiekt
FROM produkt_statystyka
INNER JOIN produkt ON produkt.id=produkt_statystyka.idprodukt
UNION
# wyróżnienia (przez użytkownika) obiektów
SELECT lista_zyczen.idkontrahent AS id, lista_zyczen.idprodukt AS obiekt
FROM lista_zyczen
INNER JOIN produkt ON produkt.id=lista_zyczen.idprodukt
UNION
```

zamknięcie zapytania

) AS transakcja_view;

Zapytanie 3 – odfiltrowanie transakcji jednoelementowych i usunięcie ich

Dla każdej *SELECT transakcja.id FROM transakcja*

GROUP BY transakcja.id HAVING count(*) =1

Wykonaj *DELETE FROM transakcja WHERE id=transakcja.id*

Zapytanie 4 – optymalizacja danych grafu G

OPTIMIZE TABLE transakcja;

Podsumowanie

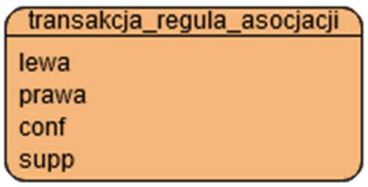
Dodanie kroku związanego z odfiltrowaniem transakcji jednoelementowych i usunięciem ich na etapie ekstrakcji jest konieczne, aby zmniejszyć liczbę danych wejściowych dla obliczeń algorytmu odpowiedzialnego za odkrywanie asocjacji. Szczególnie, że liczba takich transakcji stanowi około 25% wszystkich rozpatrywanych transakcji.

6.3.2. Implementacja kroków algorytmu odkrywania reguł asocjacji

Jako podstawa do implementacji algorytmu odkrywania asocjacji w ramach systemu e-Commerce została wykorzystana otwarta biblioteka „class.apriori.php" zawierająca implementację opisanego w rozdziale 4.2 algorytmu Apriori. Biblioteka jest dostępna w sieci Internet pod adresem elektronicznym https://github.com/VTwo-Group/Apriori-Algorithm w ramach biblioteki GitHub[10]. O wyborze tego konkretnego rozwiązania zdecydował fakt wykorzystania przez bibliotekę tej samej technologii implementacyjnej w postaci języka skryptowego PHP co docelowy system e-Commerce.

[10] GitHub to serwis internetowy przeznaczony do projektów aplikacji webowych. Posiadający repozytorium gotowych implementacji algorytmów przydatnych w budowie systemów informatycznych (https://github.com/)

Wybrana biblioteka została dostosowana do odkrywania asocjacji w ramach transakcji przygotowanych podczas ekstrakcji danych opisanych wcześniej. Kolejna modyfikacja biblioteki polegała na dostosowaniu jej do zapisu odkrytych reguł asocjacji w bazie danych systemu e-Commerce. Reguły te są zapisywane w tabeli *transakcja_regula_asocjacji*, której budowa jest przedstawiona na poniższym diagramie ERD.



Rysunek 6.10. Diagram ERD struktury tabeli transakcja_regula_asocjacji przechowującej informacje o odkrytych regułach asocjacji.
Źródło: opracowanie własne.

Tabela ta w rekordzie przechowuje informacje o pojedynczej regule asocjacji typu $A \rightarrow B$ w postaci:

- *lewa* – poprzednik A;
- *prawa* – następnik B;
- *conf* – ufność $conf(A \rightarrow B)$;
- *supp* – wsparcie $supp(A \rightarrow B)$.

6.3.3. Implementacja kroków algorytmu rekomendacji

Jako narzędzie implementacji algorytmu rekomendacji bazującego na regułach asocjacji został również wykorzystany język skryptowy PHP. Powodem tego podobnie jak przy algorytmie Apriori jest fakt, że na bazie tego języka jest zbudowana część aplikacyjna rozpatrywanego systemu e-Commerce.

Opis i zasada działania tego algorytmu w postaci poszczególnych kroków zostały szczegółowo opisane w rozdziale 4.3. W przypadku tej implementacji jako dane wejściowe w postaci zestawu silnych reguł asocjacji (AM) są wykorzystane reguły odkryte przez algorytm Apriori i zapisane w tabeli *transakcja_regula_asocjacji* oraz jako produkty, względem których następuje rekomendacja (IF) jest wykorzystany identyfikator pozyskany na podstawie zachowania użytkownika $'kliknięcie'$

powiązanego ze stroną WWW produktu. Dane wyjściowe natomiast stanowi tablica rekomendowanych produktów (RF), która jest wyświetlana na stronie WWW produktu w ramach systemu e-Commerce.

7. Badanie algorytmu

Niniejszy rozdział opisuje eksperymenty, które zostały przeprowadzone w celu zbadania algorytmu ARS w ramach funkcjonującego systemu e-Commerce oraz konkurencyjnego rozwiązania bazującego na analizie asocjacji. Szczegółowy opis implementacji algorytmów został opisany w rozdziale 6.2 i 6.3 pracy.

7.1. Wprowadzenie do badań

Zasadniczym celem badań było porównanie systemu rekomendacji zbudowanego na bazie algorytmu ARS w funkcjonującym online rozwiązaniu e-Commerce z konkurencyjnym rozwiązaniem opartym na analizie asocjacji. Do porównania wybrano tą technikę, ponieważ spełnia ona warunek bazowania na zachowaniach użytkowników w systemie bez gromadzenia innych danych ich dotyczących.

Jako punkt odniesienia, dla obu badanych algorytmów, dokonano oceny systemu rekomendacji bazującego na algorytmie losowych rekomendacji ze zbioru obiektów (produktów) będących w bazie danych systemu e-Commerce. Metoda ta bazuje na braku wszelkiej informacji o użytkownikach oraz obiektach.

W celu porównania algorytmów wybrano charakterystyki definiujące skuteczność oraz wydajność badanych mechanizmów. Szczegółowo kryteria oceny zostaną opisane w dalszej części rozdziału.

Do badań wykorzystano obserwację systemu rekomendacji i analizę danych rzeczywistych pochodzących z funkcjonującego online produkcyjnie systemu e-Commerce w postaci systemu informatycznego sklepu internetowego AM76. Techniki te były wykorzystane podczas eksperymentów ujętych w planie badań.

Eksperymenty przeprowadzone podczas badań bazowały na zachowaniach użytkowników systemu e-Commerce i oraz obiektach (produktach) zgromadzonych w bazie danych tego systemu i były realizowane na podstawie eksperymentów online. Eksperymenty tego typu dają możliwość badań na dużą skalę na wdrożonym systemie. Oceniają one działanie rekomendacji na prawdziwych użytkownikach, którzy nie są świadomi przeprowadzanego eksperymentu (Ricci i in. 2015).

7.1.1. Kryteria oceny algorytmów

Jako kryteria oceny algorytmów zaimplementowanych w systemach rekomendacji, spośród miar określonych w podrozdziale 3.5, w ramach badań zostały wybrane charakterystyki istotne z punktu widzenia użyteczności oraz wydajności systemu e- Commerce oraz możliwe do określenia na podstawie danych rejestrowanych w systemie. Są to następujące charakterystyki::

- czas generowania rekomendacji (s) – jako czas od momentu zainicjowania procesu oszacowania rekomendacji do momentu wygenerowania rekomendacji. Charakterystyka ta ma związek z opisanym w rozdziale 2.6 czasem odpowiedzi systemu e-Commerce oraz wynika ze złożoności obliczeniowej badanych algorytmów;
- pokrycie (%) – jako iloraz obiektów mogących pojawić się w rekomendacjach do wszystkich obiektów będących w bazie danych systemu e-Commerce;
- CTR (%) (współczynnik kliknięć) – jako iloraz liczby kliknięty rekomendacji (nastąpiło przejście do strony WWW obiektu zaprezentowanego w ramach rekomendacji) do liczby wyświetlonych rekomendacji w systemie e-Commerce.

Dodatkowo badano charakterystyki związane z danymi wejściowymi dla poszczególnych algorytmów takie jak:

- czas generowania grafu G (s) – jako czas powstawania grafu rekomendacji G na podstawie danych z bazy danych systemu e-Commerce. Graf generowany był co godzinę na podstawie rzeczywistych danych;
- czas generowania reguł asocjacji (s) – jako czas powstania reguł asocjacji na podstawie danych z bazy danych systemu e-Commerce. Reguły były szacowane co godzinę na podstawie rzeczywistych danych. Przyjęto taki cykl wyznaczania reguł, aby aktualność reguł była porównywalna z aktualnością grafu rekomendacji G. Jednak, można byłoby przyjąć również inny cykl identyfikacji reguł asocjacyjnych np. raz dziennie;
- wielkość grafu G – jako liczba węzłów i liczba łuków z podziałem na typy węzłów (jądra i obiekty);
- liczba reguł asocjacji – jako liczba oszacowanych silnych reguł asocjacji;
- liczba transakcji – gdzie jako transakcje były uważane podzbiory obiektów związane z zachowaniami użytkowników.

7.1.2. Wartości bazowe charakterystyk

Dla wyżej wymienionych charakterystyk funkcjonowania systemu rekomendacji dokonano oszacowania ich bazowych wartości względem, których odnoszono się do opisu wyników eksperymentów.

Ich oszacowania dokonano na podstawie wyników eksperymentu związanego z systemem rekomendacji zbudowanym na bazie algorytmu losowych rekomendacji. Wykorzystanie losowych rekomendacji jako bazowych jest stosowaną metodą do badań wybranych charakterystyk systemów rekomendacji (Beel i in. 2017). Przede wszystkim w tym wypadku wyznaczono minimalną bazową wartość dla miary CTR.

Metodą na wyznaczenie maksymalnego, dopuszczalnego dla użytkowników, czasu generowania rekomendacji wpływającego bezpośrednio na czas odpowiedzi systemu (serwera) było wykorzystanie informacji zawartych w badaniach dotyczących satysfakcji z korzystania z systemów e-Commerce (Poggi i in. 2014) (Loisel 2001). Jednoznacznie wskazują one, że im mniejsza wartość tej charakterystyki tym system jest bardziej użyteczny dla jego odbiorców. Przyjęto 1 sekundę jako górną granicę wartości miary czasu generowania rekomendacji.

Jeśli chodzi o wartość bazową dla miary pokrycie to na podstawie zasady Pareto (Grachev 2020) przyjęto minimalną akceptowalną wartość dla systemów e-Commerce wynoszącą 80%. Z przeprowadzonych przez Yinggui Wang, Ben Wang oraz Yuxin Huang badań nad występowaniem zasady Pareto w powiązaniu ze zbiorem zidentyfikowanych zachowań użytkowników najczęściej dotyczą one 80% obiektów zgromadzonych w bazie danych systemu. Pozostałe 20% obiektów było związane z zachowaniami użytkowników bardzo rzadko (Y. Wang i in. 2020).

Istotnym z punktu widzenia funkcjonowania systemu rekomendacji są: czas generowania grafu G i czas generowania reguł asocjacji. Ich wartości były determinowane od góry przez maksymalny czas wykonywania skryptu w środowisku badawczym i wartość ta wyniosła w przypadku języka skryptowego PHP 60 sekund. Wartość ta jest ustawiana systemowo dla środowiska testowego bez możliwości jej zwiększenia. Ma to związek z faktem, że większy czas powodował zbyt duże obciążenie serwera WWW, który jako rozwiązanie webowe musiał zapewnić dostęp dla wielu użytkowników, czyli działanie w tym samym czasie wielu skryptów bez

konieczności kolejkowania zadań. Ewentualne kolejkowanie wpłynęło by negatywnie w postaci zwiększenia wartości (opóźnienie) czasu generowania rekomendacji.

7.1.3. Środowisko badawcze

Środowisko badawcze stanowiło rozwiązanie e-Commerce w postaci systemu informatycznego sklepu internetowego AM76 złożonego z aplikacji i bazy danych. Dodatkowo system ten został wzbogacony o moduły umożliwiające pomiar, w czasie rzeczywistym, istotnych z punktu widzenia badania charakterystyk. Tymi modułami były:

- moduł gromadzenia danych na temat pojedynczych rekomendacji;
- moduł gromadzenia danych dotyczących grafu rekomendacji G;
- moduł gromadzenia danych dotyczących transakcji i reguł asocjacji.

W skład każdego modułu wchodziły dwa elementy:

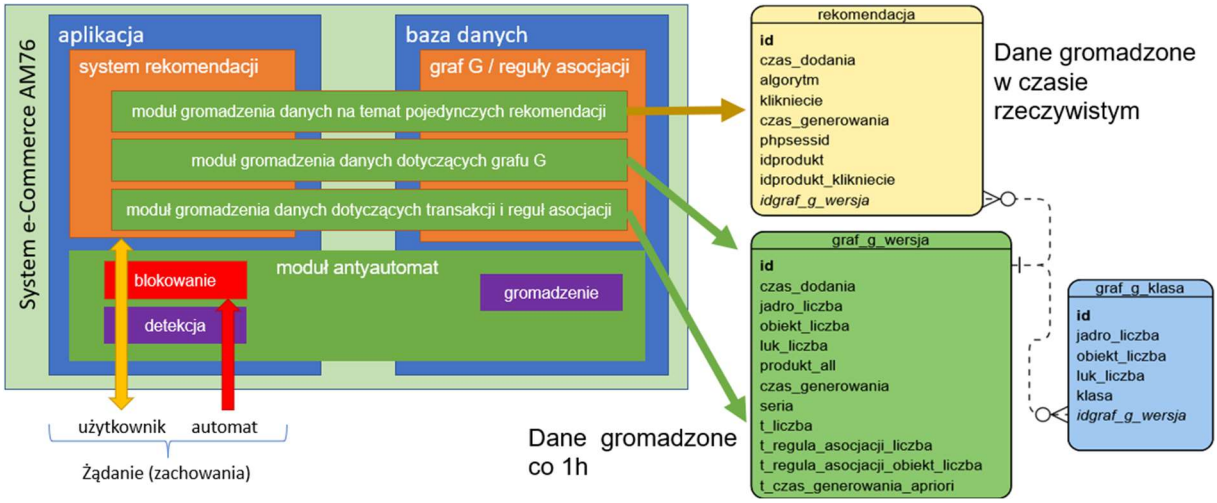
- aplikacyjny – kody zaimplementowany w ramach systemu e-Commerce umożliwiający pozyskiwanie danych w czasie rzeczywistym dotyczących badanego obszaru;
- bazodanowy – część bazy danych systemu e-Commerce, w której gromadzone były pozyskane przez element aplikacyjny dane.

Dane były gromadzone w próbkach co godzinę. Z każdą próbką związane były:

- unikalny numer (identyfikator);
- godzina powstania;
- zbiór danych dotyczący badanych charakterystyk z przedziału jednej godziny:
    - liczba klikniętych rekomendacji;
    - liczba wyświetlonych rekomendacji;
    - liczba wszystkich obiektów będących w bazie danych systemu e-Commerce;
    - liczba obiektów w grafie G;
    - liczba jąder w grafie G;
    - liczba łuków w grafie G;
    - liczba silnych reguł asocjacji;
    - liczba transakcji;

- liczba obiektów w transakcjach;
- średni czas generowanie rekomendacji;
- średni czas generowania grafu G;
- średni czas generowania reguła asocjacji;

Zgromadzone w próbkach dane były poddawane agregacji i szacowaniu w ramach oprogramowania Microsoft Excel.



Rysunek 7.1. Schemat blokowy środowiska badawczego.
Źródło: opracowanie własne

7.1.4. Plan badań

Aby osiągnąć cel badań, przeprowadzono w ramach pracy eksperymenty online, które miały zapewnić uzyskanie danych do wyznaczenia charakterystyk dla:

- systemu rekomendacji zbudowanego na bazie algorytmu losowych rekomendacji;
- systemu rekomendacji zbudowanego na bazie algorytmu ARS w oparciu o zachowania użytkowników;
- systemu rekomendacji zbudowanego na bazie algorytmu ARS w oparciu o atrybuty obiektów;
- systemu rekomendacji zbudowanego na bazie algorytmu ARS w oparciu łącznie o zachowania użytkowników i atrybuty obiektów;
- systemu rekomendacji zbudowanego na bazie algorytmu rekomendacji bazującego na regułach asocjacji;

Ponadto podczas eksperymentów analizowane były zagadnienia dotyczące:

- problemu zimnego startu;
- problemu rzadkości danych;
- problemu skalowalności.

Problemy te zostały opisane dla każdego badanego systemu w ramach analizy wyników poszczególnych eksperymentów.

Badania polegały na stosowaniu w przedziałach czasu określonych systemów rekomendacji i wyznaczania na podstawie zgromadzonych danych wartości charakterystyk.

7.1.5. Związek badań z problemem badawczym

Istotą problemu badawczego, opisanego w rozdziale 3.4, dla rekomendacji jest wyznaczenie podzbioru obiektów, które maksymalizują użyteczność dla użytkownika. Zaimplementowane algorytmy rekomendacji w systemie e-Commerce wyznaczają przedmiotowy zbiór obiektów i prezentują go użytkownikom.

Jako, że eksperymenty są prowadzone online gdzie użytkownicy nie są świadomi prowadzonych badań oraz nie jest posiadany wiarygodny wzorzec ich decyzji co do użyteczności obiektów, przyjęto w oparciu o zasady badania użyteczności w obszarze e-Commerce wykorzystanie charakterystyki CTR jako wyznacznika jakości rozwiązania problemu badawczego przez badane algorytmy (Beel i Langer 2015) (Kuanr i Mohapatra 2021).

7.1.6. Forma prezentacji wyników

W ramach badań na podstawie danych zebranych podczas eksperymentów przedstawiono w postaci wykresów zmiany wartości charakterystyk w czasie funkcjonowania zaimplementowanych rozwiązań w systemie e-Commerce.

Dodatkowo wzbogacono je o linie (przerywane) reprezentujące modele trendu i ich funkcje w postaci $y = f(x)$. Przy każdym wykresie były rozpatrywane modele trendu: liniowy, potęgowy, logarytmiczny i wykładniczy. O wyborze danego modelu trendu decydowała większa wartość współczynnika dopasowania R^2. Dzięki czemu możliwe było prognozowanie wartości badanych charakterystyk.

Ponadto do wyników badań dodano podmacierze korelacji pomiędzy wybranymi charakterystykami co pozwoliło na odkrycie powiązań pomiędzy nimi.

7.1.7. Czas realizacji badań

Pierwsze badania były prowadzone od 1 czerwca 2021r. do początku stycznia 2022r. Podczas eksperymentów okazało się, że sesje a w ramach nich zachowania ewidencjonowane w systemie pochodziły nie tylko od użytkowników, ale w dużej mierze od automatów w postaci robotów (botów): indeksujących strony WWW np. Google, poszukujących luk w systemach bezpieczeństwa lub automatycznych porównywarek cen. Automaty generowały fałszywe, z punktu widzenia pracy dane, które nie były związane z użytkownikami rozumianymi jako osoby korzystające z systemu e-Commerce, co w konsekwencji wpływało na nieprawdziwe wyniki badań. Stan ten wymusił zbudowanie w ramach środowiska badawczego, modułu „antyautomat" złożonego z elementów:

- detekcji automatycznych sesji;
- blokowania automatycznych sesji;
- gromadzenia informacji o robotach inicjujących automatyczne sesje.

Dopiero po wdrożeniu produkcyjnym ww. modułu można było przystąpić do zasadniczego etapu badań. Zakres przedmiotowych badań obejmował czas od 9 stycznia 2022r. do dnia 18 maja 2022 r.

Podczas zasadniczego etapu badań w ramach przeprowadzonych eksperymentów zgromadzono dane dotyczące poszczególnych elementów systemu e-Commerce zgodnie z poniższą tabelą:

Liczba elementów	Nazwa zbioru
2 610	Produkty (obiekty)
112 337	Sesje
213 437	Zachowania użytkowników w postaci kliknięć
6 108	Zachowania użytkowników w postaci zakupów
1 071	Zachowania użytkowników w postaci wyróżnień (przez użytkowników)
119	Atrybuty obiektów w postaci kategorii
430	Atrybuty obiektów w postaci serii
63	Atrybuty obiektów w postaci polecanych (przez eksperta)
8 857	Wersje grafu rekomendacji G
98 657	Łuki grafu rekomendacji G

8 456	Transakcje w rozumieniu analizy asocjacji
7 561	Reguły asocjacji
265 932	Wygenerowane rekomendacje (łącznie dla badanych algorytmów)
12 584	Skuteczne rekomendacje (łącznie dla badanych algorytmów)

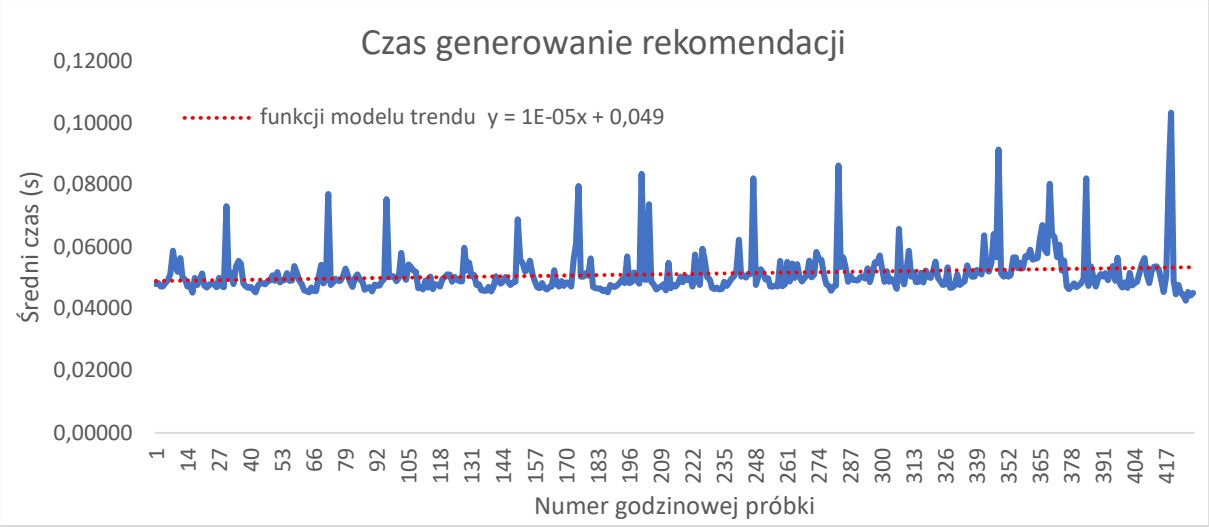
Tabela 7.1. Zakres ilościowy zgromadzonych podczas badań danych dla wszystkich eksperymentów.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

7.2. Wyniki badań

Poniżej są zamieszczone wyniki eksperymentów prowadzonych zgodnie z przedstawionym powyżej planem badań.

7.2.1. System rekomendacji zbudowany na bazie algorytmu losowych rekomendacji

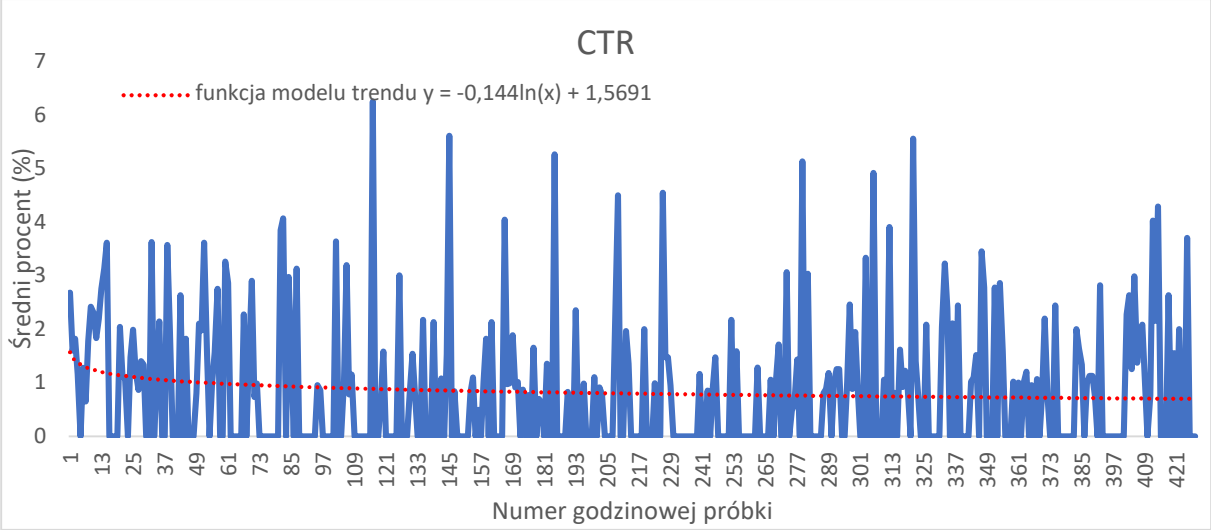
W celu oszacowania bazowych wartości charakterystyk miar funkcjonowania systemów rekomendacji, względem których prowadzono analizę wyników kolejnych badań, przeprowadzono eksperyment w ramach którego w środowisku badawczym zaimplementowano w systemie rekomendacji mechanizm losowych rekomendacji bazujących na wszystkich obiektach systemu e-Commerce. Eksperyment przeprowadzono w przedziale czasu od 13 marca 2022r. do 3 kwietnia 2022r. W tym czasie zostały zgromadzone dane dotyczące 428 godzinowych próbek.



Wykres 7.1. Wykres dla charakterystyki czas generowanie rekomendacji.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

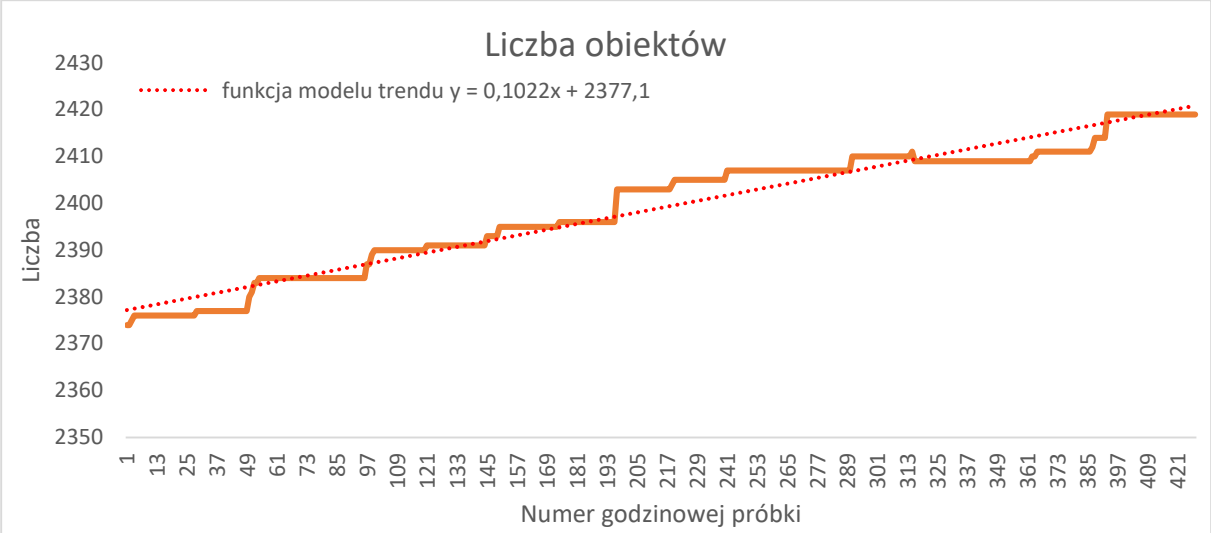
Eksperyment wykazał, że wartości charakterystyki czasu generowania rekomendacji wahały się w przedziale od 0,0426s do 0,1033s i średnio wynosiły 0,05122s. Są to wartości bardzo dobre z punktu widzenia systemów e-Commerce i wynoszą znacznie mniej niż przyjęta górna granica wartości miary czasu generowania rekomendacji wynosząca 1 sekundę. Nieznaczny wzrost wartości charakterystyki w czasie wykazany na bazie funkcji modelu trendu był związany ze stopniowo rosnącą wielkością danych (liczbą obiektów w systemie e-Commerce) pokazaną na kolejnym wykresie. Czas generowania rekomendacji jest bardzo ważny z punktu widzenia obszaru e-Commerce i wpływa bezpośrednio na czas odpowiedzi systemu (serwera), który został opisany we wcześniejszej części pracy. Decyduje on o tym, po jakim czasie użytkownik od wysłania żądania do środowiska serwerowego systemu otrzyma odpowiedź, czyli kod strony WWW będzie dostępny w przeglądarce. Tak jak zostało wcześniej powiedziane „milisekundy warte są miliony” (Glynn i in. 2020). Dodatkowo wartość czasu generowania rekomendacji wynika wprost ze złożoności obliczeniowej zastosowanego algorytmu w ramach systemu rekomendacji.

Charakterystyka pokrycie, definiowana jako iloraz obiektów mogących pojawić się w rekomendacjach do wszystkich obiektów będących w bazie danych systemu e-Commerce w przypadku przeprowadzonego eksperymentu bazującego na wszystkich obiektach (produktach) systemu e-Commerce przyjmowała wartość 100% przez cały czas trwania eksperymentu. Co z punktu widzenia rozwiązań e-Commerce jest wartością najlepszą z możliwych do uzyskania.



Wykres 7.2. Wykres dla charakterystyki CTR.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Kolejną istotną z punktu widzenia e-Commerce charakterystyką jest CTR (współczynnik kliknięć). Charakterystyka ta jest kluczowa dla porównania różnych systemów rekomendacji. W przypadku badanego mechanizmu losowych rekomendacji na 39107 przeprowadzonych rekomendacji charakterystyka ta przyjęła średnią wartość 0,89754% w przybliżeniu to 0,90%. Wartość ta dla kolejnych eksperymentów została uznana za minimalną bazową wartość dla miary CTR.



Wykres 7.3. Wykres dla charakterystyki liczba obiektów.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Liczba obiektów, czyli produktów dostępnych dla użytkowników w systemie e-Commerce podczas eksperymentu była od 2374 do 2419 czyli przyrost wynosił 1,86%. Średnio było to 2399 obiektów z tendencją wzrostową określoną przez liniową funkcję modelu trendu ze współczynnikiem dopasowania R^2 (>0,9). Na podstawie funkcji modelu trendu dokonano oszacowania miesięcznego przyrostu obiektów $\Delta(y)$ w bazie danych. W oparciu o założenie, że próbkowanie i szacowanie wartości liczby obiektów odbywało się co godzinę, obliczono $\Delta(y) = |f(x') - f(0)|$ gdzie $x' = 720$ (liczba godzin w miesiącu) dla $f(x) = 0{,}1022x + 2377{,}1$ oszacowanie dało wynik $\Delta(y) = 74$.

Charakterystyka w postaci liczby obiektów, w przypadku mechanizmu losowych rekomendacji, była interpretowana jako wielkość danych i wpływała na wartość czasu generowania rekomendacji.

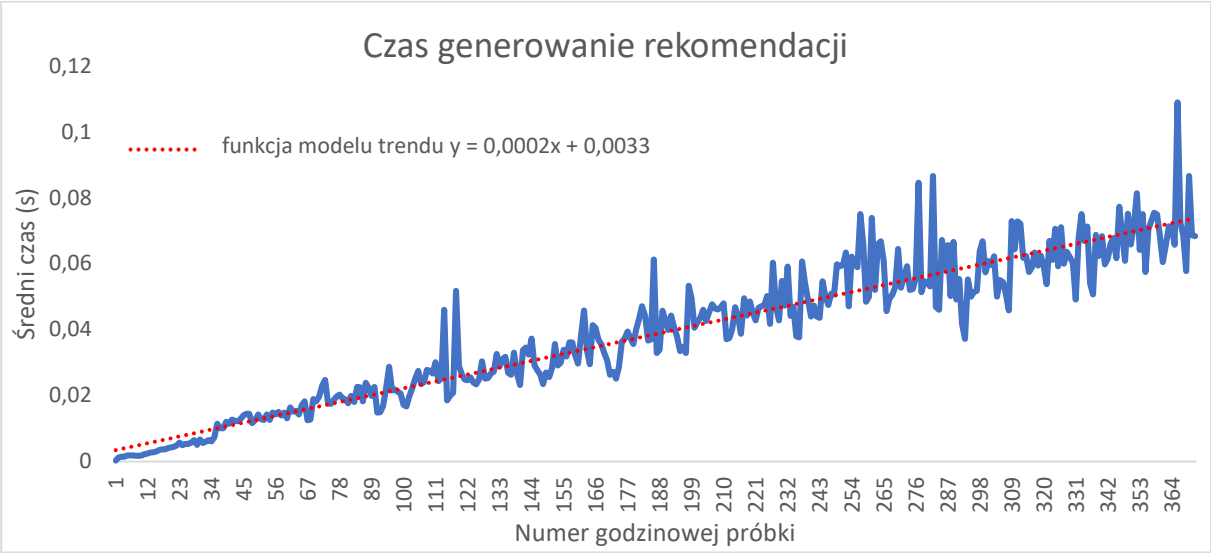
7.2.2. System rekomendacji zbudowany na bazie algorytmu ARS w oparciu o zachowania użytkowników

Eksperyment przeprowadzono w przedziale czasu od 6 kwietnia 2022r. do 21 kwietna 2022r. W tym czasie zostały zgromadzone dane dotyczące 370 godzinowych próbek. Eksperyment polegał na tym, że w środowisku badawczym zaimplementowano w systemie rekomendacji algorytm ARS. Specyfiką tego rozwiązania było to, że graf rekomendacji sesji G składał się tylko z klas jąder związanych z zachowaniami użytkowników, czyli takich jak:

K_1 – zakup obiektu;

K_2 – kliknięcie w obiekt;

K_5 – wyróżnienie (przez użytkownika) obiektu.

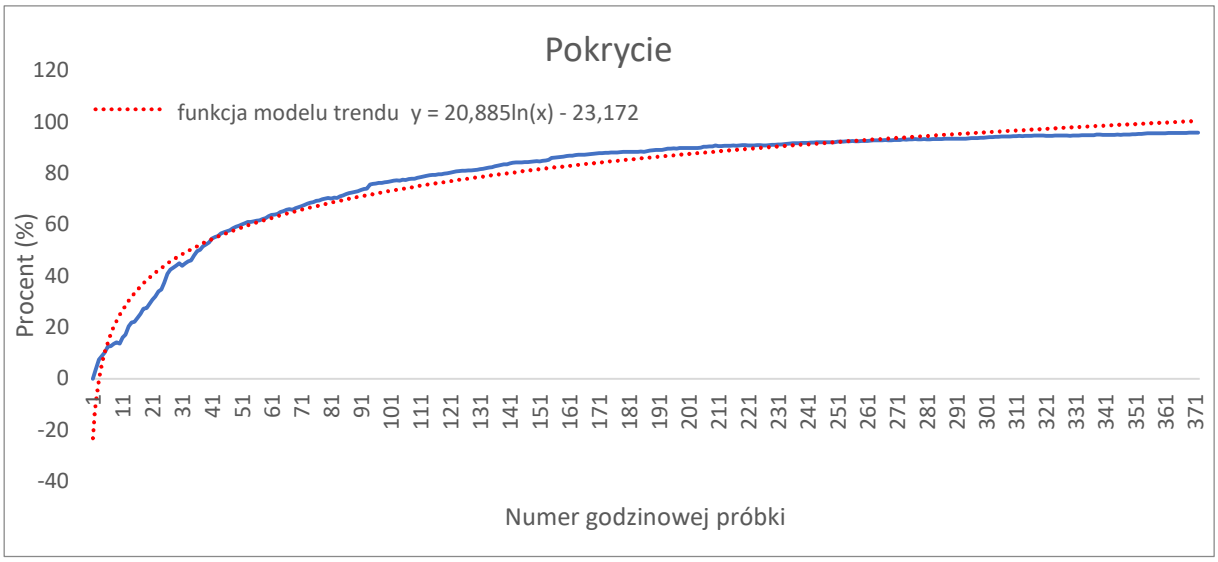


Wykres 7.4. Wykres dla charakterystyki czas generowanie rekomendacji. Źródło: opracowanie własne na podstawie przeprowadzonych badań

Eksperyment wykazał, że wraz z upływem czasu funkcjonowania systemu rekomendacji, wartości charakterystyki czasu generowania rekomendacji wzrastały do 0,10910s i średnio wynosiły 0,03868s. Są to wartości bardzo dobre z punktu widzenia

systemów e-Commerce i wynoszą mniej niż przyjęta górna granica wartości miary czasu generowania rekomendacji wynosząca 1 sekundę.

Na podstawie funkcji modelu trendu, która posiada wysoki współczynnik dopasowania R^2 (>0,9) dokonano oszacowania po jakim czasie system rekomendacji mógł wpłynąć negatywnie, z punktu widzenia e-Commerce, na czas odpowiedzi systemu. Przyjęto jako górną graniczną wartość opóźnienia 1 sekundę i wykorzystano dla funkcji modelu trendu $f(x) = 0{,}0002x + 0{,}0033$ postać funkcji odwrotnej $f^{-1}(y) = -5000(0{,}0033 - x)$. Okazało się, że system potrzebowałby 4983 godzin pracy, czyli 208 dni na osiągnięcie wartości granicznej miary czasu generowanie rekomendacji.



Wykres 7.5. Wykres dla charakterystyki pokrycie.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

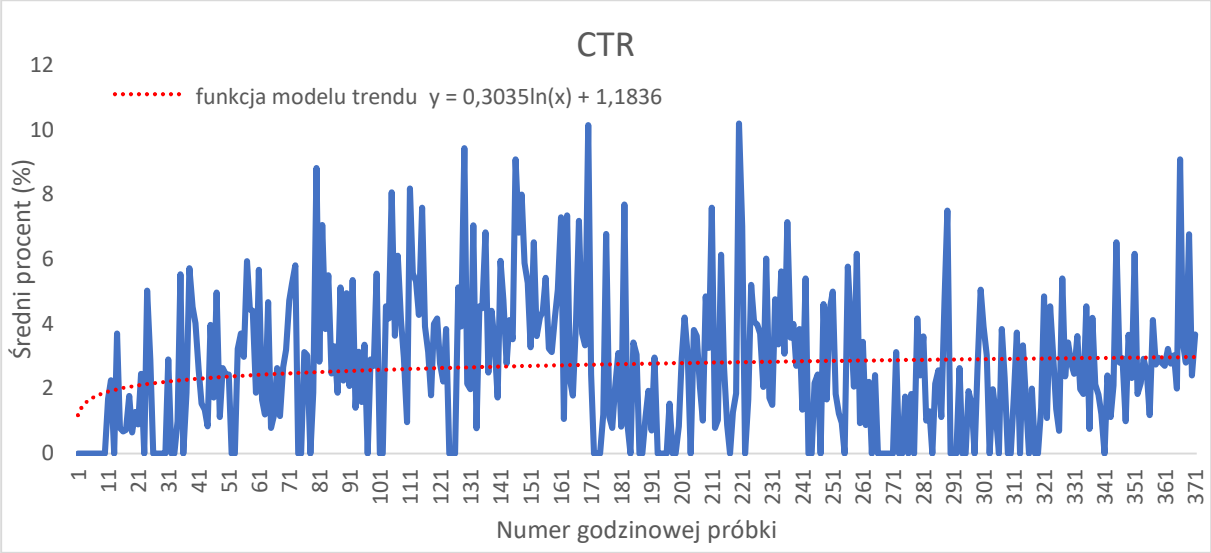
Drugą istotną z punktu widzenia e-Commerce miarą jest pokrycie. Teoretycznie algorytm rekomendacji powinien dokonywać rekomendacji dla wszystkich obiektów (produktów, usług) będących w bazie danych systemu e-Commerce. W praktyce jednak nie dla wszystkich obiektów tego typu są dostępne wystarczające informacje, jest to związane z takimi problemami, jak: zimny start dla początkowego czasu działania systemu (Burke 2007) i rzadkość danych dla nowych obiektów (Sarwar i in. 2000). Stąd też jako minimalną wartość bazową dla miary pokrycie przyjęto 80%.

Jak widać na przedstawionym powyżej wykresie, w przypadku implementacji algorytmu ARS w oparciu o zachowania użytkowników, pokrycie dążyło do 100% w oparciu o logarytmiczną funkcję modelu trendu, ale go nie osiągało. Stan taki

utrzymywał się tak długo, jak długo w systemie e-Commerce pojawiały się nowe obiekty. Z każdym nowym obiektem w początkowym czasie jego istnienia w bazie danych nie było związanych zachowań użytkowników, czyli nie było informacji, na podstawie których można było dokonać oszacowania rekomendacji. Poniższa tabela przedstawia czasy po jakich osiągano progi wartości charakterystyki pokrycie podczas trwania eksperymentu. Co jest istotne, graniczna wartość pokrycia 80% została osiągnięta po 5 dniach. Ponadto średnia wartość pokrycia podczas eksperymentu wynosiła 79,78%

Czas eksperymentu	**Pokrycie**
5h	10%
13h	20%
21h	30%
26h	40%
40h	50%
52h	60%
79h	70%
121h	**80%**
209h	**90%**
∞	100%

Tabela 7.2. Osiąganie progów charakterystyki pokrycie podczas trwania eksperymentu.
Źródło: opracowanie własne na podstawie przeprowadzonych badań



Wykres 7.6. Wykres dla charakterystyki CTR.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

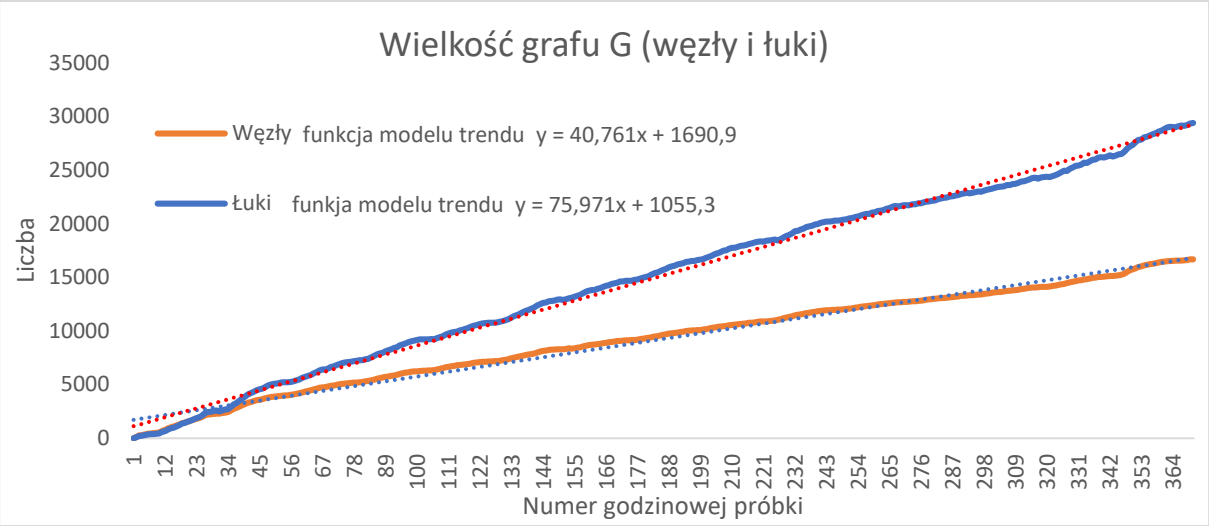
Podczas trwanie eksperymentu okazało się, że charakterystyka CTR jest bardzo zmienna w czasie i praktycznie nie jest możliwe wyznaczenie dla niej dobrej funkcji modelu trendu. Najlepsza jaką udało się wyznaczyć miała bardzo niski współczynnik dopasowania R^2 wynoszący zaledwie 0,0407. W związku z powyższym, podczas eksperymentu, oszacowanie średniej wartości tej charakterystyki dokonano na podstawie pozyskanych danych zawartych w poniższej tabeli i charakterystyka ta przyjęła średnią wartość 2,90%.

	Wartość
Liczba rekomendacji	38111
Liczba kliknięć	1106
CTR	**2,90205%**

Tabela 7.3. Wartość parametru „efektywność rekomendacji" dla eksperymentu. Źródło: opracowanie własne na podstawie przeprowadzonych badań

Wartość ta była wyższa od przyjętej minimalnej wartości bazowej dla charakterystyki CTR oszacowanej na bazie funkcjonowania systemu rekomendacji bazującego na algorytmie losowym, która wynosiła 0,90%. Co oznacza, że zaimplementowane rozwiązanie może być skutecznie stosowane w rozwiązaniach e-Commerce.

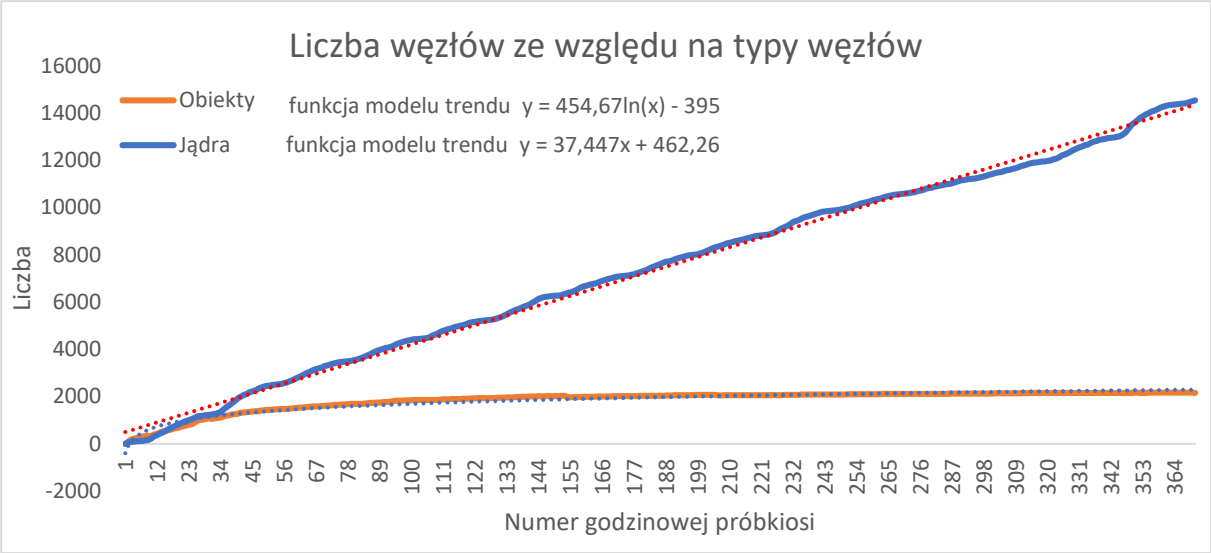
Oprócz miar istotnych z punktu widzenia systemu e-Commerce zbadano podczas eksperymentu dodatkowe charakterystyki związane z algorytmem ARS. Wyniki zostały przedstawione poniżej.



Wykres 7.7. Wykres d*la charakterystyki wielkość grafu G.*

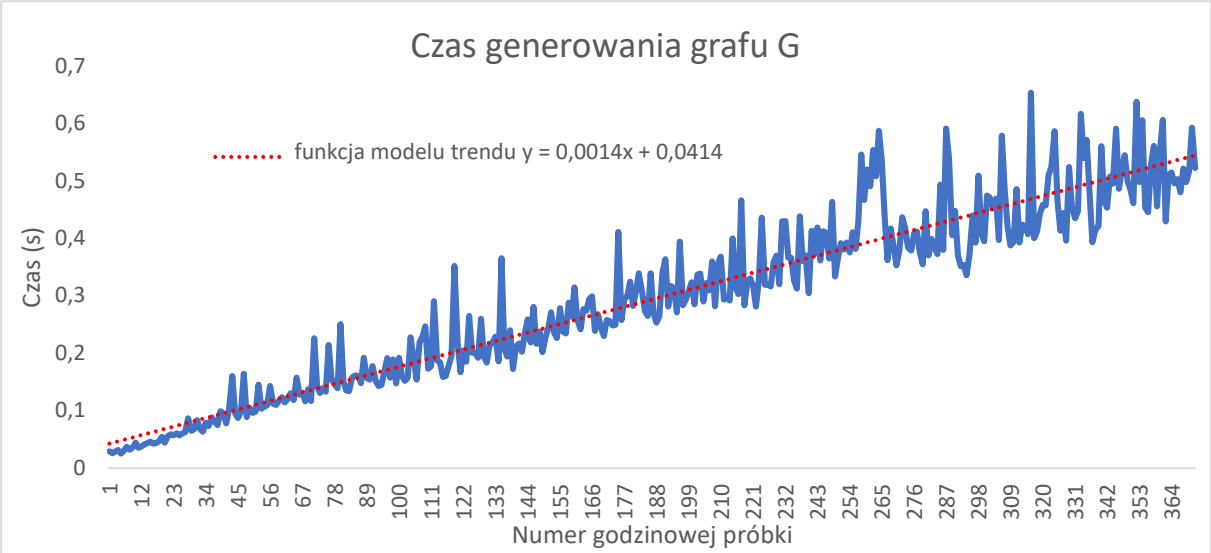
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Graf rekomendacji G zgodnie ze swoją definicją składa się z węzłów oraz łuków i teoretycznie to te dwa zbiory mają znaczenie dla jego wielkości. W praktyce ze względu na przyjętą implementację struktury grafowej w postaci listy krawędzi na wielkość grafu G wpływa przede wszystkim liczba łuków. Można przyjąć, że liczba łuków jest wyznacznikiem wielkości danych na jakich bazuje algorytm ARS. Na podstawie wysokiej wartości współczynnika dopasowania R^2 (>0,9) modelu trendu dla charakterystyki liczba łuków wskazano, że przyrost danych ma postać liniową w postaci $f(x) = 75{,}971x + 1055{,}3$.



Wykres 7.8. Wykres dla charakterystyk liczba obiektów i liczba jąder.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Węzły grafu rekomendacji G podzielone są na dwa zbiory: obiekty i jądra. Podczas eksperymentu okazało się, że liczba jąder z czasem jest znacznie większa niż liczba obiektów. Obserwacja ta potwierdza przedstawione podczas szacowania złożoności obliczeniowej założenie, że zbiór obiektów przyrasta w bardzo niewielkim stopniu, natomiast znaczący jest przyrost liczby jąder. Maksymalnie charakterystyki te przyjęły wartości: liczba jąder 14 592 i liczba obiektów 2 168.



Wykres 7.9. Wykres dla charakterystyki czas generowania grafu G.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Dla czasu generowania grafu G również jak dla wcześniejszej charakterystyki liczby łuków, funkcja modelu trendu przyjmuje postać liniową z wysokim współczynnikiem dopasowania R^2 (>0,9). Wartość charakterystyki czas generowania grafu G podczas trwania eksperymentu nie przekroczyła 0,65380s i wynosiła średnio 0,29376s. Wartości te były mniejsze niż przyjęta górna granica wartości miary czasu generowania grafu G wynosząca 60 sekund. W oparciu o funkcję modelu trendu $f(x) = 0{,}0014x + 0{,}0414$ i postać funkcji odwrotnej do niej w postaci $f^{-1}(y) = -714{,}286(0{,}0144 - y)$ oszacowano, że system potrzebuje 703 godzin pracy, czyli 29 dni na osiągnięcie górnej granicy wartości miary czasu generowania grafu G. Z punktu widzenia systemu e-Commerce to dobry wynik i wskazuje na dobrą skalowalność systemu. Wynika, to z faktu że system rekomendacji może funkcjonować efektywnie przy zwiększającej się z czasem wielkości danych wymagających oszacowania w ramach algorytmu.

Miejscami zwiększone wartości tej charakterystyki podczas trwania eksperymentu wynikały z faktu, że eksperyment był prowadzony w produkcyjnym środowisku online systemu e-Commerce, na który oddziaływały różne czynniki zewnętrzne jak na przykład wzmożone zainteresowanie użytkowników lub cyberataki, które pochłaniały zasoby obliczeniowe środowiska serwerowego.

W ramach analizy zebranych podczas eksperymentu danych dokonano oszacowania korelacji pomiędzy badanymi charakterystykami a wielkością grafu rekomendacji G definiowaną jako liczba łuków. Obliczeń korelacji dokonano względem tej charakterystyki, gdyż jest ona wyznacznikiem wielkości danych na jakich bazuje algorytm ARS. Dodatkowo dokonano oszacowania korelacji przedmiotowych charakterystyk a czasem generowania grafu G.

Wyniki oszacowania korelacji zostały przedstawione w postaci podmacierzy korelacji (pełna macierz korelacji powinna zawierać oszacowania korelacji dla wszystkich charakterystyk względem siebie).

	Wielkość grafu G (liczba łuków)	**Czas generowania grafu G**
Liczba obiektów	0,81506	0,77071
Liczba jąder	0,99982	0,95203
Czas generowanie grafu G	0,95207	1
Czas generowanie rekomendacji	0,95248	0,92870
CTR	0,01139	0,01119

Tabela 7.4. Podmacierz korelacji
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Na bazie podmacierzy korelacji zostało pokazane, że charakterystyki wielkość grafu G i czas generowania grafu G są silnie dodatnio skorelowane z liczbą jąder i czasem generowania rekomendacji. Ponadto wielkość grafu G silnie dodatnio koreluje z czasem generowania grafu G. Jednakże oszacowanie korelacji wskazało, że jest bardzo słaba korelacja pomiędzy wielkością grafu G a CTR.

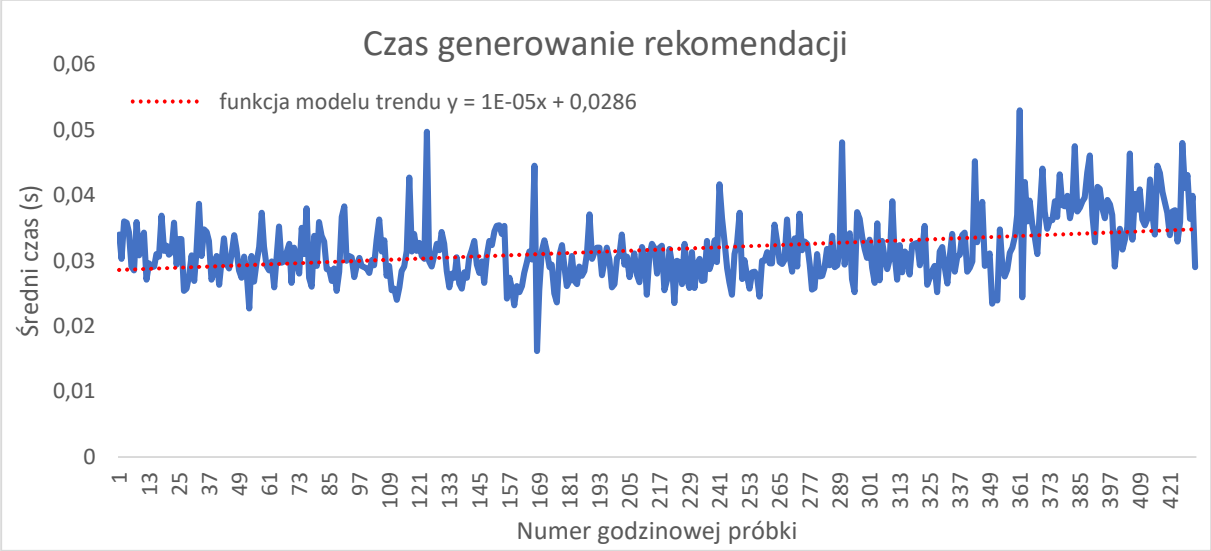
7.2.3. System rekomendacji zbudowany na bazie algorytmu ARS w oparciu o atrybuty obiektów

Eksperyment przeprowadzono w przedziale czasu od 21 kwietnia 2022r. do 9 maja 2022r. W tym czasie zostały zgromadzone dane dotyczące 430 godzinowych próbek. Eksperyment polegał na tym, że w środowisku badawczym zaimplementowano w systemie rekomendacji algorytm ARS. Specyfiką tego rozwiązania było to, że graf rekomendacji sesji G składał się tylko z klas jąder związanych z atrybutami obiektów, czyli takich jak:

K_3 – kategoria obiektu;

K_4 – seria obiektu;

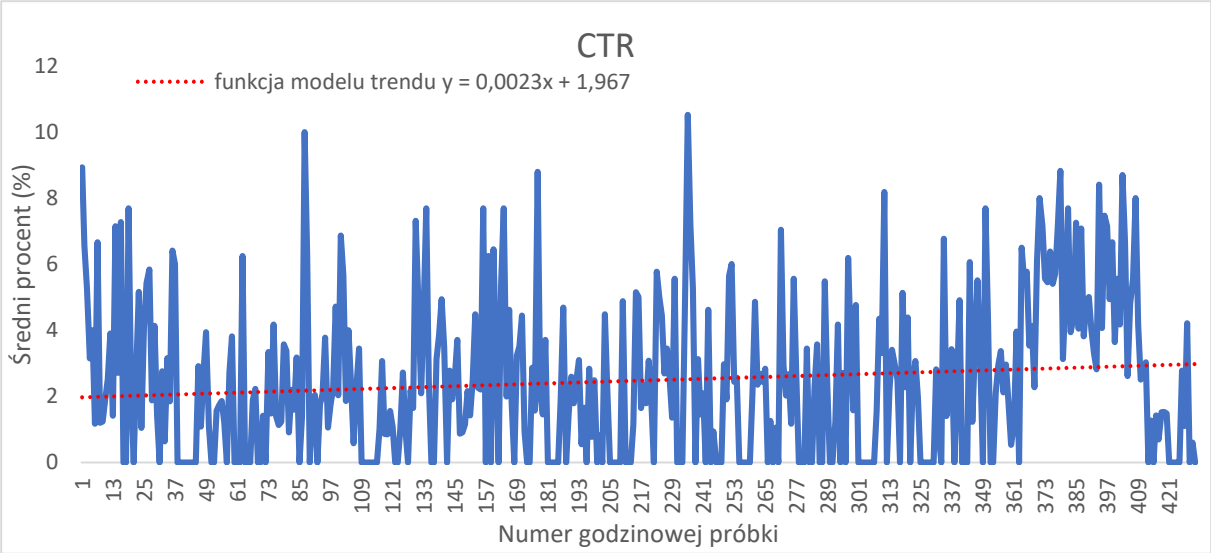
K_6 – polecanie (przez eksperta) obiektu.



Wykres 7.10. Wykres dla charakterystyki czas generowanie rekomendacji. Źródło: opracowanie własne na podstawie przeprowadzonych badań

Eksperyment wykazał, że wraz z upływem czasu funkcjonowania systemu rekomendacji, wartości charakterystyki czasu generowania rekomendacji wahały się w przedziale od 0,01620s do 0,05300s i średnio wynosiły 0,03171s. Są to wartości bardzo dobre z punktu widzenia systemów e-Commerce i stanowią mniej niż przyjęta górna granica wartości miary czasu generowania rekomendacji wynosząca 1 sekundę.

Charakterystyka pokrycie w przypadku przeprowadzonego eksperymentu miała wartość 100% przez cały czas trwania eksperymentu. Co z punktu widzenia rozwiązań e-Commerce jest wartością najlepszą z możliwych do uzyskania. Wynikało to z faktu, że każdy obiekt będący w bazie danych systemu e-Commerce posiadał choćby jeden związany z nim atrybut. Oznacza to, że w badanym podczas eksperymentu systemie rekomendacji nie występują problemy zimnego startu i rzadkości danych.



Wykres 7.11. Wykres dla charakterystyki CTR.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

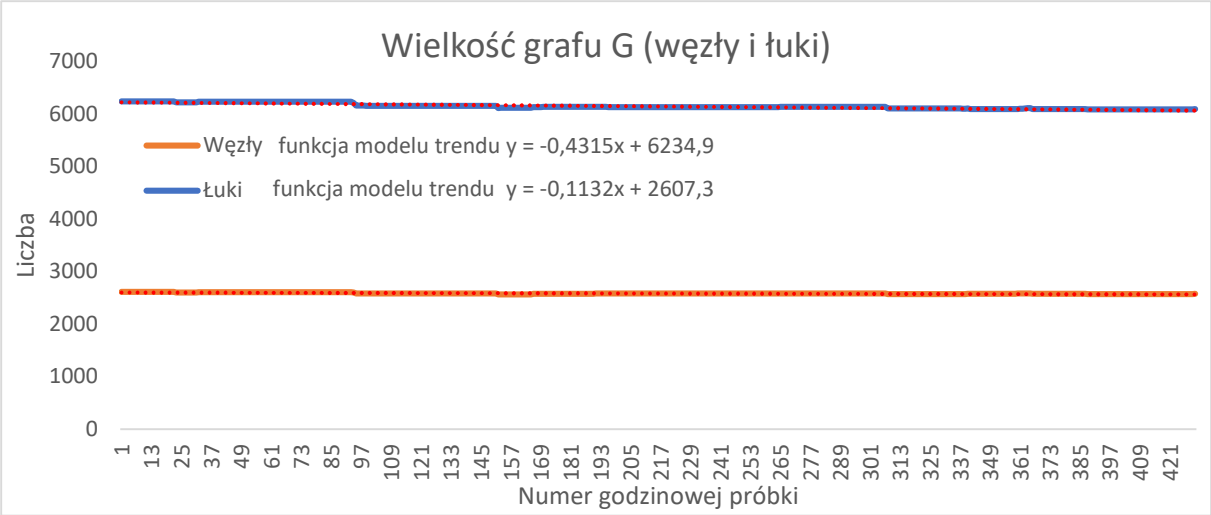
Podczas trwania eksperymentu okazało się, że charakterystyka CTR jest bardzo zmienna w czasie i praktycznie nie jest możliwe wyznaczenie dla niej dobrej funkcji modelu trendu. Najlepsza jaką udało się wyznaczyć miała bardzo niski współczynnik dopasowania R^2 wynoszący zaledwie 0,0151. W związku z powyższym, podczas eksperymentu, oszacowanie średniej wartości tej charakterystyki dokonano na podstawie pozyskanych danych zawartych w poniższej tabeli i charakterystyka ta przyjęła średnią wartość 2,63%.

	Wartość
Liczba rekomendacji	38724
Liczba kliknięć	1019
CTR	**2,63144%**

Tabela 7.5. Wartość parametru „efektywność rekomendacji" dla eksperymentu.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

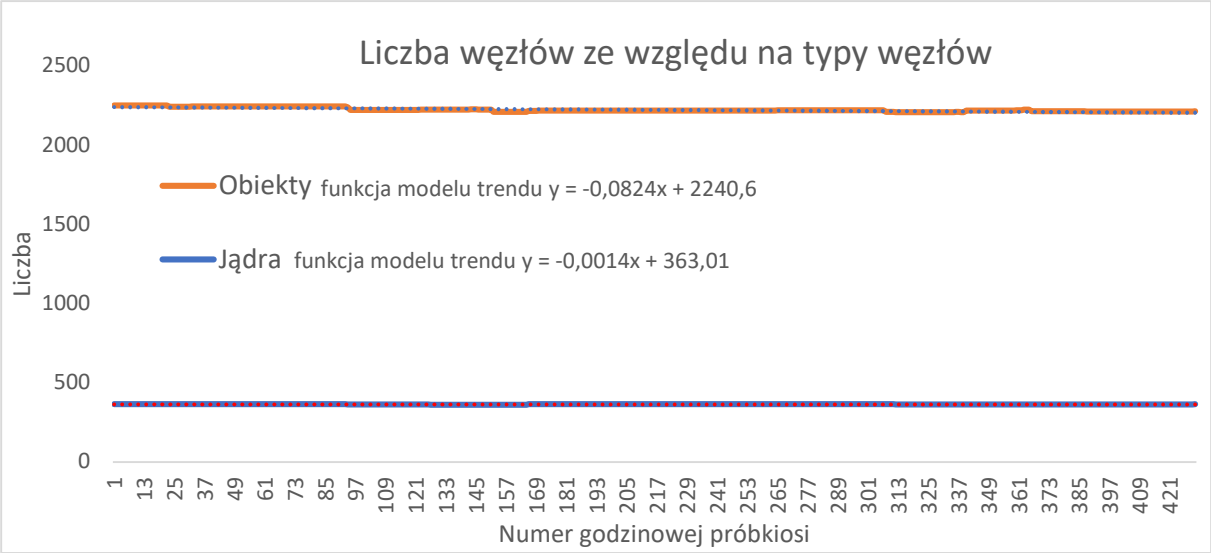
Wartość ta była wyższa od przyjętej minimalnej wartości bazowej dla charakterystyki CTR wynoszącej 0,90%. Co oznacza, że zaimplementowane rozwiązanie może być skutecznie stosowane w rozwiązaniach e-Commerce.

Oprócz miar istotnych z punktu widzenia systemu e-Commerce zbadano podczas eksperymentu dodatkowe charakterystyki związane z algorytmem ARS. Wyniki zostały przedstawione poniżej.



Wykres 7.12. Wykres d*la charakterystyki wielkość grafu G.*
Źródło: opracowanie własne na podstawie przeprowadzonych badań

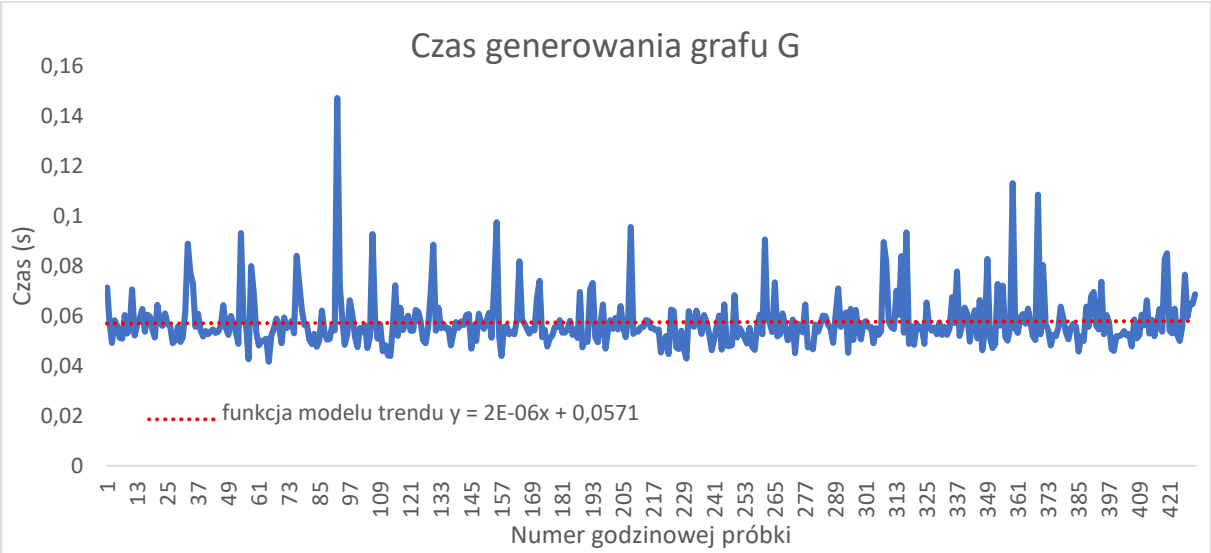
Analogicznie jak we wcześniejszym eksperymencie wielkość grafu G, a dokładnie liczba łuków jest wyznacznikiem wielkości danych na jakich bazuje algorytm ARS. W przypadku rozpatrywanego eksperymentu liczba ta była praktycznie stała i wynosiła od 6090 do 6245 łuków oraz średnio 6148 łuki.



Wykres 7.13. Wykres dla charakterystyk liczba obiektów i liczba jąder.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Podczas eksperymentu okazało się, że liczba jąder i obiektów jest również praktycznie stała. Różnice liczby obiektów wynosiła niecałe 2% podobnie dla jąder. Wynikało to z faktu, że liczba aktywnych obiektów w systemie e-Commerce zmienia

się. Są dodawane do bazy nowe obiekty i dezaktywowane stare nieatrakcyjne dla użytkowników.



Wykres 7.14. Wykres dla charakterystyki czas generowanie grafu G.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Dla czasu generowania grafu G również jak dla wcześniejszej charakterystyki liczby łuków, funkcja modelu trendu przyjmuje postać liniową. Z reguły wartości charakterystyki czas generowania grafu G podczas trwania eksperymentu były w przedziale od 0,04180s do 0,14750s i wynosiły średnio 0,05756s. Wartości te były mniejsze niż przyjęta górna granica wartości miary czasu generowania grafu G wynosząca 60 sekund.

W ramach analizy zebranych podczas eksperymentu danych dokonano analogicznie jak we wcześniejszym eksperymencie oszacowania korelacji pomiędzy badanymi charakterystykami.

	Wielkość grafu G (liczba łuków)	**Czas generowania grafu G**
Liczba obiektów	0,96390	0,00391
Liczba jąder	0,35268	-0,04524
Czas generowanie grafu G	-0,00725	1
Czas generowanie rekomendacji	-0,24720	0,15759
CTR	-0,12191	-0,00161

Tabela 7.6. Podmacierz korelacji
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Na bazie podmacierzy korelacji zostało pokazane, że charakterystyki wielkość grafu G jest silnie dodatnio skorelowana z liczbą obiektów. Pozostałe charakterystyki są bardzo słabo skorelowane pomiędzy sobą.

7.2.4. System rekomendacji zbudowany na bazie algorytmu ARS w oparciu łącznie o zachowania użytkowników i atrybuty obiektów

Eksperyment przeprowadzono w przedziale czasu od 9 stycznia 2022r. do 30 stycznia 2022r. W tym czasie zostały zgromadzone dane dotyczące 496 godzinowych próbek. Eksperyment polegał na tym, że w środowisku badawczym zaimplementowano w systemie rekomendacji algorytm ARS. Specyfiką tego rozwiązania było to, że graf rekomendacji sesji G składał się zarówno z klas jąder związanych z zachowaniami użytkowników, jak i atrybutami obiektów, czyli takich jak:

K_1 – zakup obiektu (zachowanie);

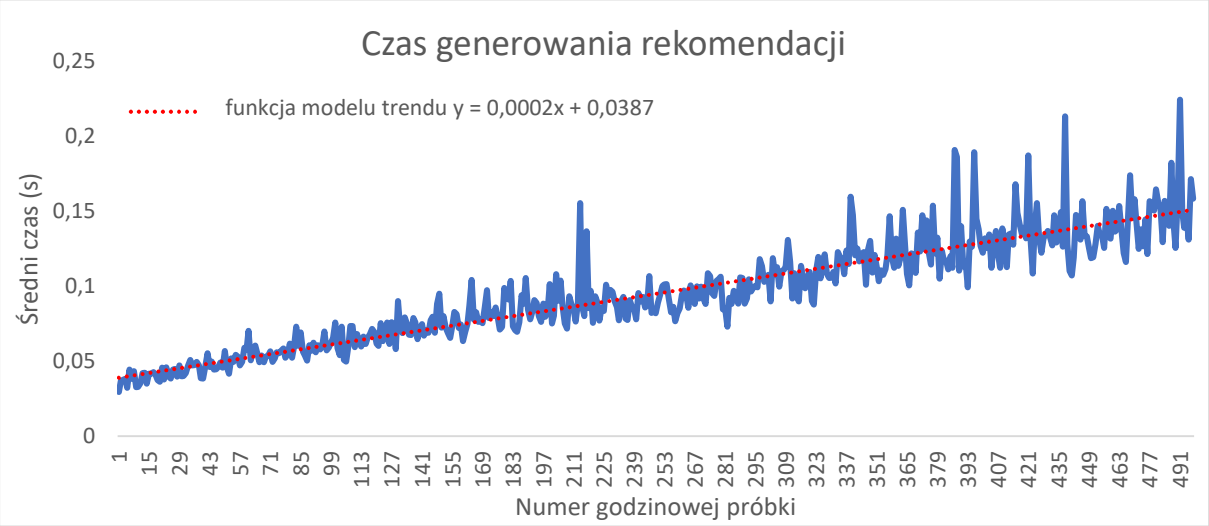
K_2 – kliknięcie w obiekt (zachowanie);

K_3 – kategoria obiektu (atrybut);

K_4 – seria obiektu (atrybut);

K_5 – wyróżnienie (przez użytkownika) obiektu (zachowanie);

K_6 – polecanie (przez eksperta) obiektu (atrybut).

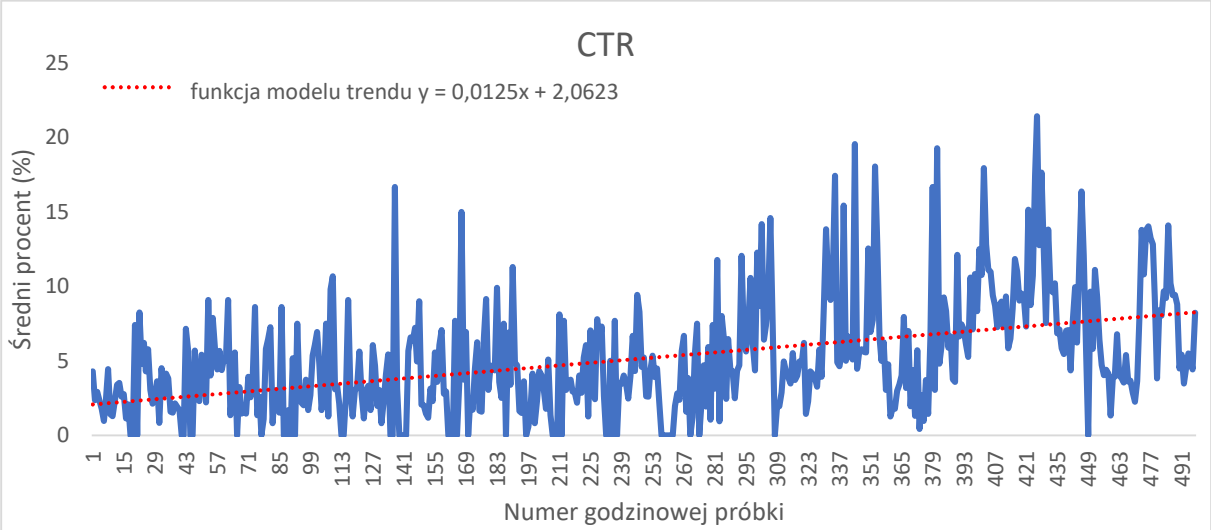


Wykres 7.15. Wykres dla charakterystyki czas generowanie rekomendacji. Źródło: opracowanie własne na podstawie przeprowadzonych badań

Eksperyment wykazał, że wraz z upływem czasu funkcjonowania systemu rekomendacji, wartości charakterystyki czasu generowania rekomendacji wzrastały i wahały się w przedziale od 0,00030s do 0,10910s i średnio wynosiły 0,03868s. Są to wartości bardzo dobre z punktu widzenia systemów e-Commerce i wynoszą mniej niż przyjęta górna granica wartości miary czasu generowania rekomendacji wynosząca 1 sekundę.

Na podstawie funkcji modelu trendu, który posiadał wysoki współczynnik dopasowania R^2 (>0,8) dokonano oszacowania po jakim czasie system rekomendacji mógł wpłynąć negatywnie, z punktu widzenia e-Commerce, na czas odpowiedzi systemu. Przyjęto jako graniczną wartość opóźnienia 1 sekundę i wykorzystano dla funkcji modelu trendu $y = 0{,}0002x + 0{,}0387$ postać funkcji odwrotnej $f^{-1}(y) = -5000(0{,}0387 - x)$. Okazało się, że system potrzebowałby 4806 godzin pracy, czyli 200 dni na osiągnięcie wartości granicznej miary czasu generowanie rekomendacji.

Charakterystyka pokrycie analogicznie do wcześniejszego eksperymentu miała wartość 100% przez cały czas trwania eksperymentu. Oznacza to, że w badanym podczas eksperymentu systemie rekomendacji nie występują problemy zimnego startu i rzadkości danych.



Wykres 7.16. Wykres dla charakterystyki CTR.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Podczas trwania eksperymentu okazało się, że charakterystyka CTR analogicznie jak przy wcześniejszych eksperymentach jest bardzo zmienna w czasie i praktycznie

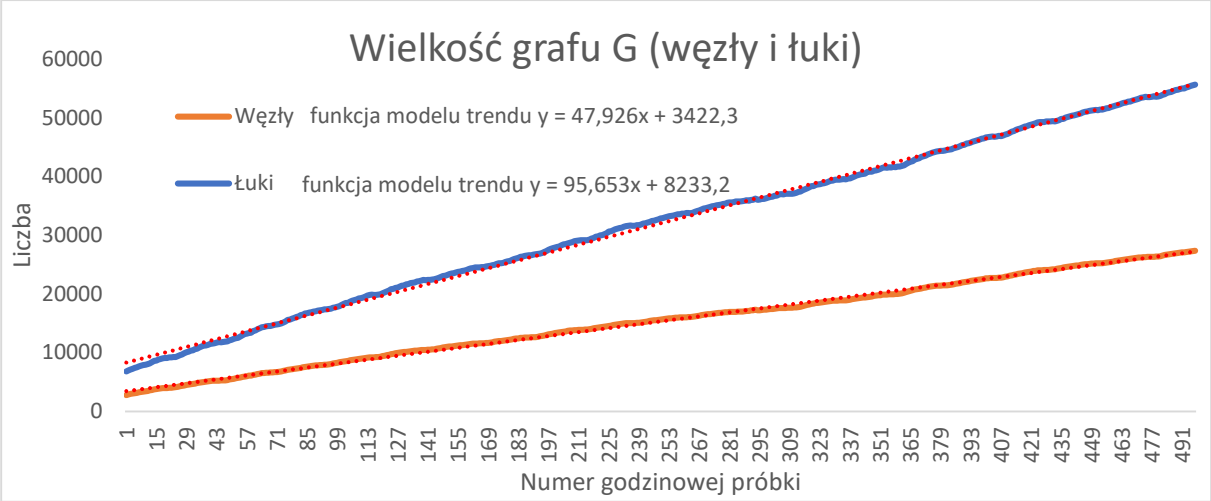
nie jest możliwe wyznaczenie dla niej dobrej funkcji modelu trendu. Najlepsza jaką udało się wyznaczyć miała bardzo niski współczynnik dopasowania R^2 wynoszący zaledwie 0,2082. W związku z powyższym, podczas eksperymentu, oszacowanie średniej wartości tej charakterystyki dokonano na podstawie pozyskanych danych zawartych w poniższej tabeli i charakterystyka ta przyjęła średnią wartość 5,21%.

	Wartość
Liczba rekomendacji	65122
Liczba kliknięć	3394
CTR	**5,21176%**

Tabela 7.7 Wartość parametru „efektywność rekomendacji" dla eksperymentu.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Wartość ta była wyższa od przyjętej minimalnej wartości bazowej dla charakterystyki CTR która wynosi 0,90%. Co oznacza, że zaimplementowane rozwiązanie może być skutecznie stosowane w rozwiązaniach e-Commerce.

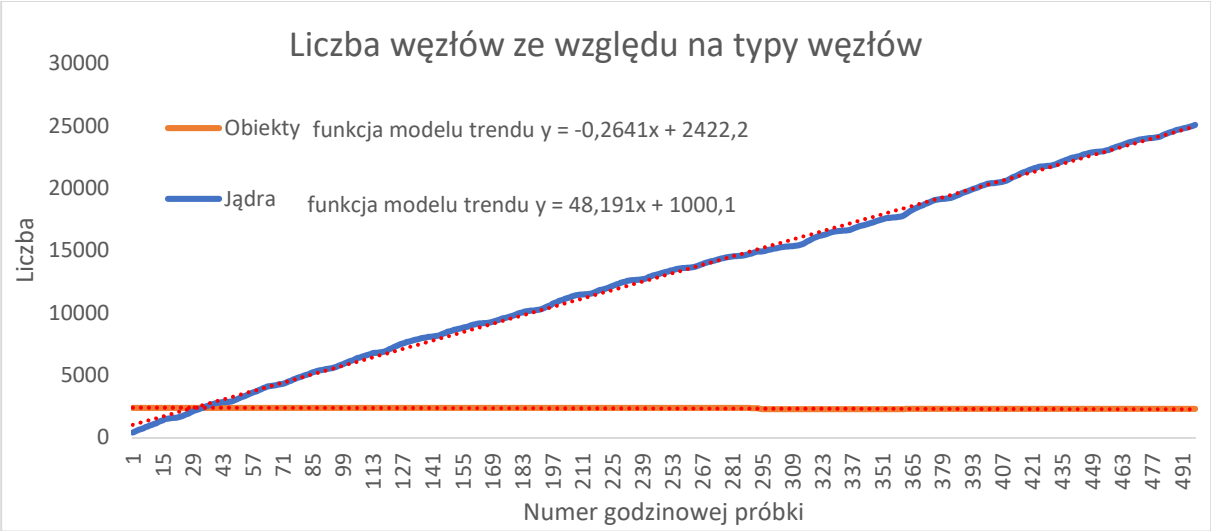
Oprócz miar istotnych z punktu widzenia systemu e-Commerce zbadano podczas eksperymentu dodatkowe charakterystyki związane z algorytmem ARS. Wyniki zostały przedstawione poniżej.



Wykres 7.17. Wykres d*la charakterystyki wielkość grafu G.*
Źródło: opracowanie własne na podstawie przeprowadzonych badań

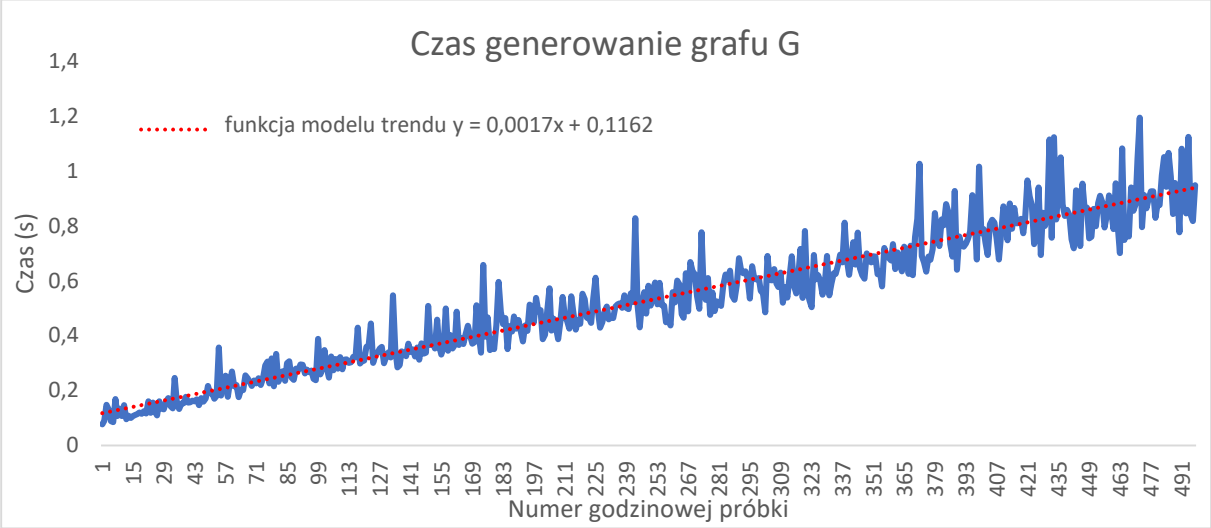
Na wielkość grafu G wpływa przede wszystkim liczba łuków. Można przyjąć, że liczba łuków jest wyznacznikiem wielkości danych na jakich bazuje algorytm ARS. Na podstawie wysokiej wartości współczynnika dopasowania R^2 (>0,9) modelu trendu dla

charakterystyki liczba łuków wskazano, że przyrost danych ma postać liniową w postaci $f(x) = 95{,}653x + 8233{,}2$.



Wykres 7.18. Wykres dla charakterystyk liczba obiektów i liczba jąder.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Węzły grafu rekomendacji G podzielone są na dwa zbiory: obiekty i jądra. Podczas eksperymentu okazało się, że liczba jąder z czasem jest znacznie większa niż liczba obiektów. Obserwacja ta potwierdza przedstawione podczas szacowania złożoności obliczeniowej założenie, że liczba obiektów zmienia się w bardzo niewielkim stopniu, natomiast znaczący jest przyrost liczby jąder. Maksymalnie charakterystyki te przyjęły wartości: liczba jąder 13 000 i liczba obiektów 2 356.



Wykres 7.19. Wykres dla charakterystyki czas generowania grafu G.

Źródło: opracowanie własne na podstawie przeprowadzonych badań

Dla czasu generowania grafu G również jak dla wcześniejszej charakterystyki liczby łuków, funkcja modelu trendu przyjmuje postać liniową ze współczynnikiem dopasowania R^2 (>0,9). Zasadniczo wartość charakterystyki czas generowania grafu G podczas trwania eksperymentu nie przekroczyła 1,19520s i wynosiła średnio 0,52906s. Wartości te były mniejsze niż przyjęta górna granica wartości miary czasu generowania grafu G wynosząca 60 sekund. W oparciu o funkcję modelu trendu $f(x) = 0{,}0017\mathrm{x} + 0{,}1162$ i postać funkcji odwrotnej do niej w postaci $f^{-1}(y) = -588{,}235(0{,}1162 - y)$ oszacowano, że system potrzebuje 519 godzin pracy, czyli 21 dni na osiągnięcie górnej granicy wartości miary czasu generowania grafu G. Z punktu widzenia systemu e-Commerce to dobry wynik i wskazuje na dobrą skalowalność systemu.

W ramach analizy zebranych podczas eksperymentu danych dokonano oszacowania korelacji pomiędzy badanymi charakterystykami a wielkością grafu rekomendacji G definiowaną jako liczba łuków. Obliczeń korelacji dokonano względem tej charakterystyki, gdyż jest ona wyznacznikiem wielkości danych na jakich bazuje algorytm ARS. Dodatkowo dokonano oszacowania korelacji przedmiotowych charakterystyk a czasem generowania grafu G. Wyniki oszacowania korelacji zostały przedstawione w postaci podmacierzy korelacji.

	Wielkość grafu G (liczba łuków)	**Czas generowania grafu G**
Liczba obiektów	-0,81650	-0,78439
Liczba jąder	0,99982	0,96139
Czas generowanie grafu G	0,96104	1
Czas generowanie rekomendacji	0,92633	0,91050
CTR	0,45160	0,46142

Tabela 7.8. Podmacierz korelacji
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Na bazie podmacierzy korelacji zostało pokazane, że charakterystyki wielkość grafu G i czas generowania grafu G są silnie dodatnio skorelowane z liczbą jąder i czasem generowania rekomendacji. Ponadto wielkość grafu G silnie dodatnio koreluje z czasem generowania grafu G.

7.2.5. System rekomendacji zbudowany na bazie algorytmu rekomendacji bazującego na regułach asocjacji

W celu porównania funkcjonowania systemów rekomendacji bazujących na różnych odmianach algorytmu ARS (wcześniejsze eksperymenty) z konkurencyjnym rozwiązaniem, przeprowadzono eksperyment w ramach którego, w środowisku badawczym zaimplementowano w systemie rekomendacji algorytm bazujący na regułach asocjacji. Algorytm wyszukiwał silne reguły asocjacji z minimalny poziom wsparcia wynoszącym 0,2% oaz minimalny poziom ufności równym 20%. Eksperyment składał się z trzech serii przeprowadzonych w następujących przedziałach czasu:

- seria A - od 9 maja 2022r. do 11 maja 2022r. (44 godzinowe próbki);
- seria B - od 11 maja 2022r. do 13 maja 2022r. (41 godzinowych próbek);
- seria C - od 16 maja 2022r. do 18 maja 2022r. (42 godzinowe próbek).

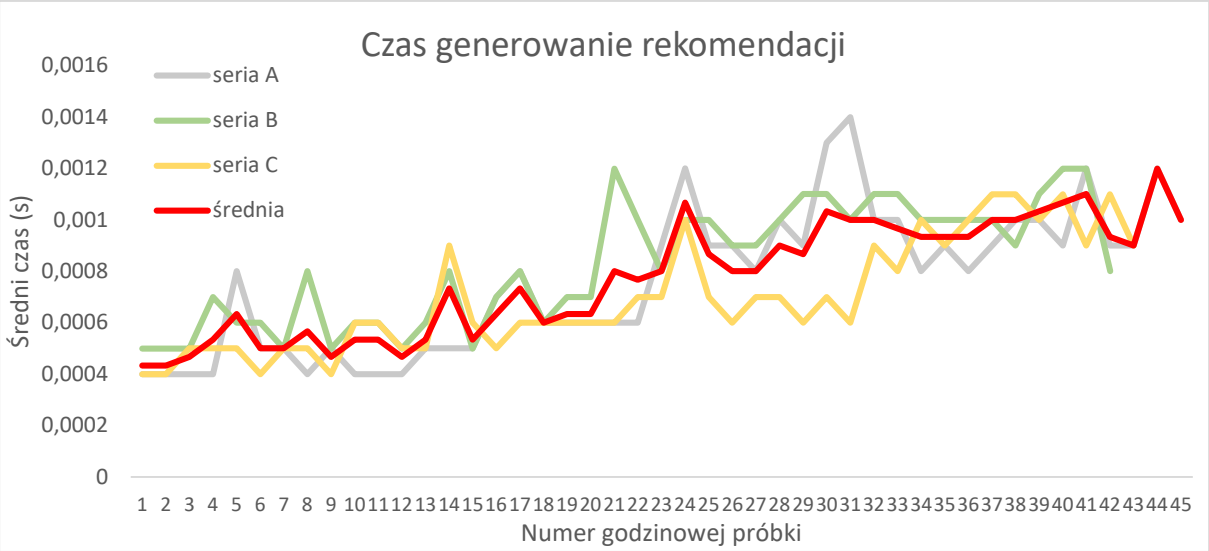
Liczba próbek była znacznie mniejsza niż podczas wcześniejszych eksperymentów. Wynikało to z faktu, że system rekomendacji bazujący na regułach asocjacji wymagał danych w postaci zbioru silnych reguł asocjacji. Szacowanie tych reguł w oparciu o opisany we wcześniejszej części pracy algorytm Apriori przekraczało bazową maksymalną wartość czasu generowania reguł asocjacji wynoszącą 60 sekund. Po przekroczeniu tego czasu system informatyczny dokonujący odkrywania reguł przerywał swoje funkcjonowanie co wynikało z ograniczonej mocy obliczeniowej środowiska badawczego. Stąd też w ramach eksperymentu zostały przeprowadzone trzy serie badań celem potwierdzenia obserwacji. Warto podkreślić, że przyjęto ten sam godzinowy cykl generowania grafu rekomendacji G oraz reguł asocjacji. Zapewniało to te samą aktualność danych będących podstawą do rekomendacji. Jeśli w przypadku oszacowania silnych reguł asocjacji przyjęto by innych cykl wyznaczania zbioru reguł (np.: co 24 h) być może wyniki byłyby inne, w szczególności w odniesieniu do algorytmu losowego.

Do oszacowania silnych reguł asocjacji został wykorzystany zbiór transakcji złożony z trzech typów zachowań użytkowników związanych z obiektami takich jak:

- zakup obiektu;
- kliknięcie w obiekt;

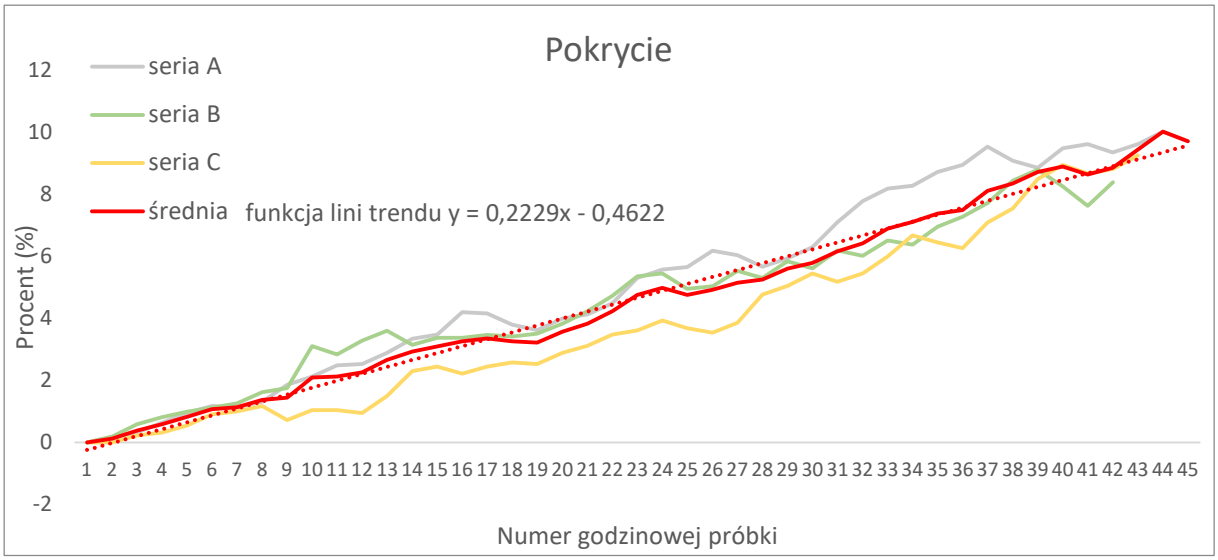
- wyróżnienie (przez użytkownika) obiektu;

Przy budowie zbioru transakcji były brane pod uwagę tylko te transakcje, w których liczność obiektów była równa lub większa od dwóch.



Wykres 7.20. Wykres dla charakterystyki czas generowania rekomendacji. Źródło: opracowanie własne na podstawie przeprowadzonych badań

Eksperyment wykazał, że wraz z upływem czasu funkcjonowania systemu rekomendacji, wartości charakterystyki czasu generowania rekomendacji dla wszystkich trzech serii wzrastały i wahały się średnio dla serii w przedziale od 0,00043s do 0,00120s i średnio wynosiły 0,00077s. Wartość ich była mniejsza niż przyjęta górna granica wartości miary czasu generowania rekomendacji wynosząca 1 sekundę. Jednak ze względu na małą liczbę uzyskanych podczas eksperymentu próbek we wszystkich seriach nie jest możliwe zaobserwowanie zmian wartości tej charakterystyki w dłuższym przedziale czasu.



Wykres 7.21. Wykres dla charakterystyki pokrycie.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

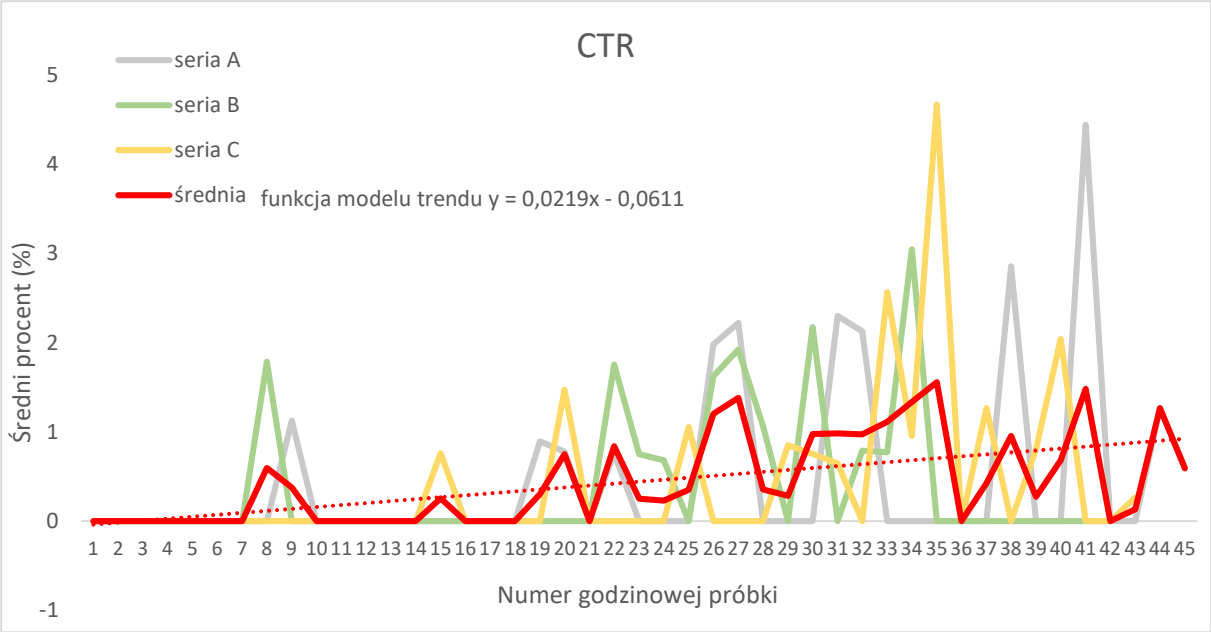
Jak widać na przedstawionym powyżej wykresie, w przypadku implementacji algorytmu rekomendacji bazującego na regułach asocjacji, wartości charakterystyki pokrycie we wszystkich seriach nie przekroczyły 10%. To bardzo niska wartość z punktu widzenia e-Commerce mocno odbiegająca od przyjętej minimalnej wartość bazowej dla miary pokrycie wynoszącej 80%. Ponadto średnia wartość pokrycia podczas eksperymentu wynosiła 4,66%.

Na podstawie funkcji modelu trendu $f(x) = 0{,}2229x - 0{,}4622$, dla średnich wartości z serii, która posiadał wysoki współczynnik dopasowania R^2 (>0,9) i funkcji odwrotnej $f^{-1}(y) = -4{,}48632(-x - 0{,}4622)$ dokonano oszacowania po jakim czasie system rekomendacji mógłby potencjalnie, gdyby nie ograniczenie mocy obliczeniowej, osiągać progi wartości charakterystyki pokrycie. Co jest istotne, graniczna wartość pokrycia zostałaby osiągnięta po 15 dniach.

Potencjalny czas	**Pokrycie**
47h	10%
92h	20%
137h	30%
182h	40%
226h	50%
271h	60%
316h	70%
361h	**80%**
406h	**90%**

451h	**100%**

Tabela 7.9. Osiąganie progów charakterystyki pokrycie podczas trwania eksperymentu.
Źródło: opracowanie własne na podstawie przeprowadzonych badań



Wykres 7.22. Wykres dla charakterystyki CTR.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

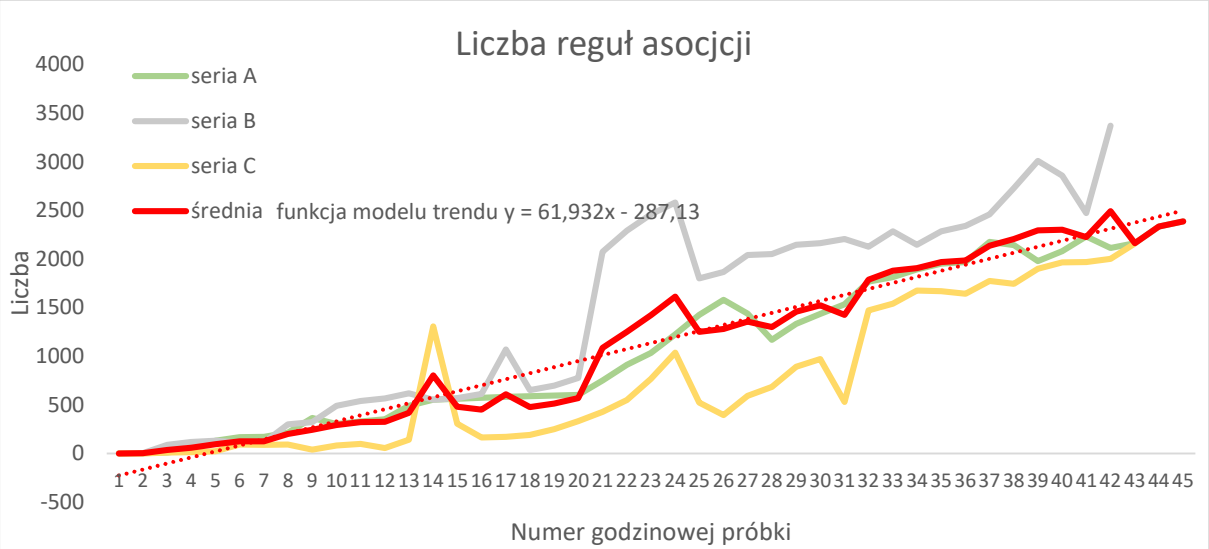
Podczas trwania eksperymentu okazało się, że charakterystyka CTR dla średnich wartości z serii analogicznie jak przy wcześniejszych eksperymentach jest bardzo zmienna w czasie i praktycznie nie jest możliwe wyznaczenie dla niej dobrej funkcji modelu trendu. Najlepsza jaką udało się wyznaczyć miała bardzo niski współczynnik dopasowania R^2 wynoszący zaledwie 0,3296. W związku z powyższym, podczas eksperymentu, oszacowanie średniej wartości tej charakterystyki dokonano na podstawie pozyskanych danych zawartych w poniższej tabeli i charakterystyka ta przyjęła średnią wartość 0,5%.

	Wartość
Liczba rekomendacji	4317
Liczba kliknięć	22
CTR	**0,50282 %**

Tabela 7.10 Wartość parametru „efektywność rekomendacji" dla eksperymentu.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

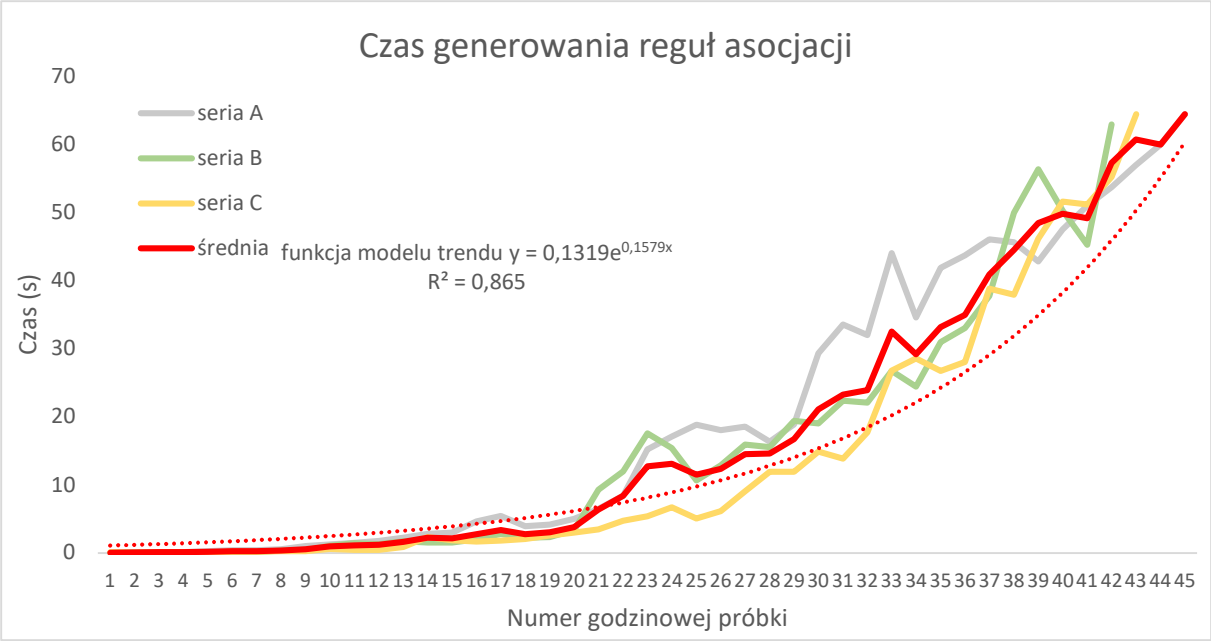
Wartość ta była niższa od przyjętej minimalnej wartości bazowej dla charakterystyki CTR wynoszącej 0,90%. Co oznacza, że zaimplementowane rozwiązanie nie będzie efektywne do zastosowania w rozwiązaniach e-Commerce.

Oprócz miar istotnych z punktu widzenia systemu e-Commerce zbadano podczas eksperymentu dodatkowe charakterystyki związane z algorytmem rekomendacji bazującego na regułach asocjacji. Wyniki zostały przedstawione poniżej.



Wykres 7.23. Wykres d*la charakterystyki liczba reguł asocjacji.*
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Można przyjąć, że liczba reguł asocjacji jest wyznacznikiem wielkości danych na jakich funkcjonuje algorytm rekomendacji bazujący na regułach asocjacji. Na podstawie wysokiej wartości współczynnika dopasowania R^2 (>0,9) modelu trendu dla charakterystyki liczba reguł asocjacji wskazano, że przyrost danych ma postać liniową w postaci $f(x) = 61{,}932x - 287{,}13$. Wynik ten potwierdził oszacowania teoretyczne złożoności obliczeniowej i pamięciowej które były klasy O(n), czyli liniowe gdzie n to liczba reguła asocjacji.



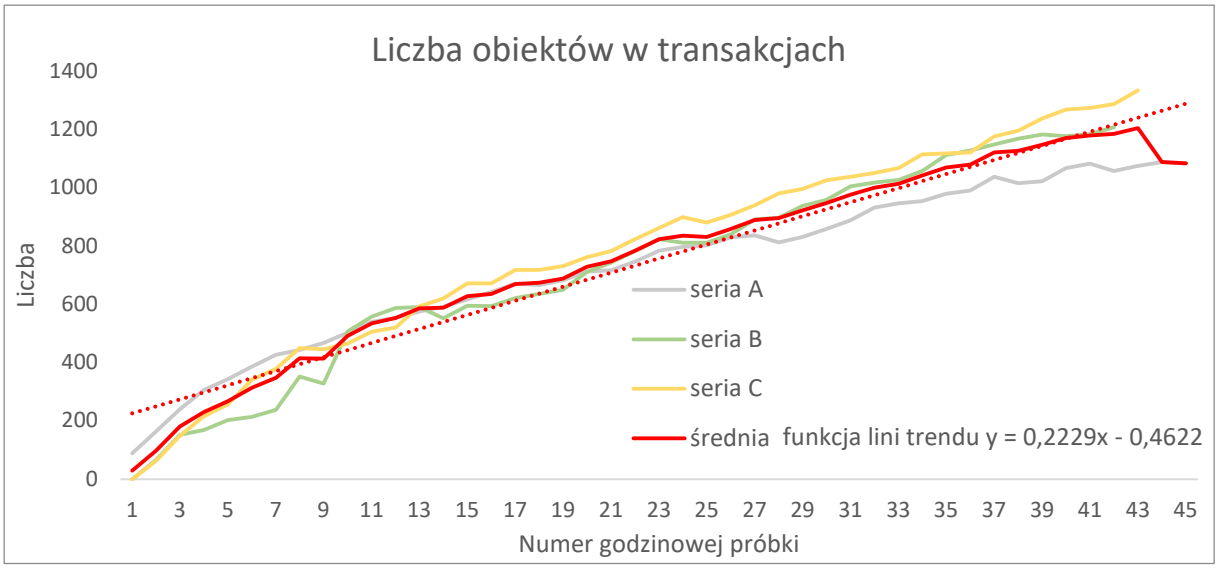
Wykres 7.24. Wykres dla charakterystyki czas generowania reguł asocjacji. Źródło: opracowanie własne na podstawie przeprowadzonych badań

Wartość charakterystyki czas generowania reguł asocjacji podczas trwania eksperymentu w każdej serii bardzo szybko rosła osiągając i przekraczając przyjętą górną granicę wartości miary czasu generowania reguł asocjacji wynoszącą 60 sekund odpowiednio:

- seria A – w 45 próbie;
- seria B – w 42 próbie;
- seria C – w 43 próbie.

Po przekroczeniu ww. górnej granicy środowisko badawcze nie miało technicznych możliwości odkrywania kolejnych reguł asocjacji.

Dla czasu generowania reguł asocjacji funkcja modelu trendu przyjmuje postać wykładniczą ze współczynnikiem dopasowania R^2 (>0,8) i ma postać wykładniczą $f(x) = 0{,}1319\mathrm{e}^{0{,}1579\mathrm{x}}$. Funkcja ta potwierdza oszacowania teoretyczne złożoności obliczeniowej i pamięciowej, które również są klasy wykładniczej $O(2^n)$ gdzie n to liczba obiektów w transakcjach. Liczba ta kształtowała się w trakcie trwania eksperymentu zgodnie z poniższym wykresem.



Wykres 7.25. Wykres dla liczby obiektów w transakcjach.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Z punktu widzenia systemu e-Commerce to bardzo zły wynik i nie daje podstaw do zastosowania w przypadku rozwiązań działających online. Algorytm ten jest bardzo podatny na problem skalowalności systemu.

Sytuacja może być poprawiona poprzez wykorzystanie do generowania reguł asocjacji środowiska serwerowego dużej mocy działającego w trybie offline i przekazującego do modułów online wyniki swoich oszacowań, czyli zbiory silnych reguł asocjacji. Jednak takie rozwiązanie odbiega od przyjętych podczas badań założeń i wykorzystywanego środowiska badawczego. Ponadto prawdopodobnie koszty utrzymania takiego środowiska mogłyby przewyższyć potencjalne korzyści co z punktu widzenia e-Commerce byłoby nieracjonalnie.

W ramach analizy zebranych podczas eksperymentu danych dokonano oszacowania korelacji pomiędzy badanymi charakterystykami a liczbą reguł asocjacji. Obliczeń korelacji dokonano względem tej charakterystyki, gdyż jest ona wyznacznikiem wielkości danych na jakich funkcjonuje algorytm rekomendacji bazujący na regułach asocjacji. Dodatkowo dokonano oszacowania korelacji przedmiotowych charakterystyk a czasem generowania reguł asocjacji. Wyniki oszacowania korelacji zostały przedstawione w postaci podmacierzy korelacji.

	Liczba reguł asocjacji	Czas generowania reguł asocjacji
Liczba reguł asocjacji	1	0,92393
Czas generowanie reguł asocjacji	0,92393	1
Czas generowanie rekomendacji	0,93955	0,82096
CTR	0,56006	0,45333

Tabela 7.11. Podmacierz korelacji
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Na bazie podmacierzy korelacji zostało pokazane, że charakterystyki czas generowania reguł asocjacji i czas generowania rekomendacji są silnie dodatnio skorelowane z liczbą reguła asocjacji. Natomiast korelacja ww. charakterystyk z CTR jest słabo skorelowana.

7.3. Podsumowanie wyników badań

W ramach podsumowania przeprowadzonych badań zostały przedstawione wyniki eksperymentów w sposób umożliwiający ich porównanie.

7.3.1. Porównanie algorytmów

Podczas każdego z eksperymentów zostały zmierzone charakterystyki miar funkcjonowania systemu zdefiniowane jako kryteria oceny algorytmów zaimplementowanych w systemach rekomendacji takie jak:

- CTR;
- pokrycie;
- czas generowanie rekomendacji;

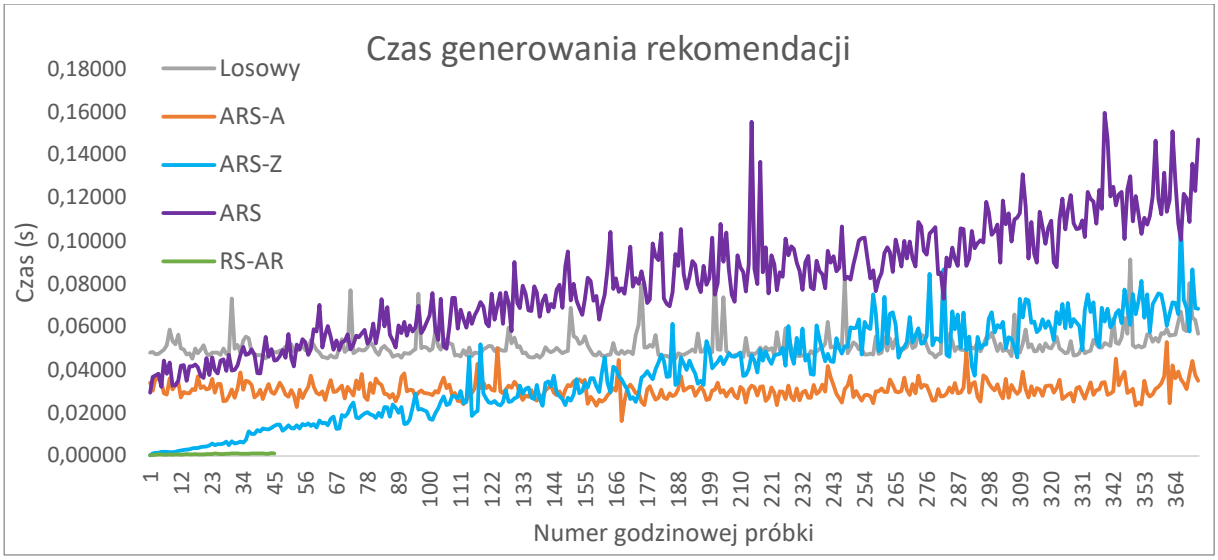
Do porównania algorytmów wykorzystano pierwszych 370 godzinowych próbek eksperymentów dla:

- systemu rekomendacji zbudowanego na bazie algorytmu losowych rekomendacji (określony skrótem „A-Los”);
- systemu rekomendacji zbudowanego na bazie algorytmu ARS w oparciu o zachowania użytkowników (określony skrótem „ARS-Z”);

- systemu rekomendacji zbudowanego na bazie algorytmu ARS w oparciu o atrybuty obiektów (określony skrótem „ARS-A”);
- systemu rekomendacji zbudowanego na bazie algorytmu ARS w oparciu łącznie o zachowania użytkowników i atrybuty obiektów (określony skrótem „ARS”);

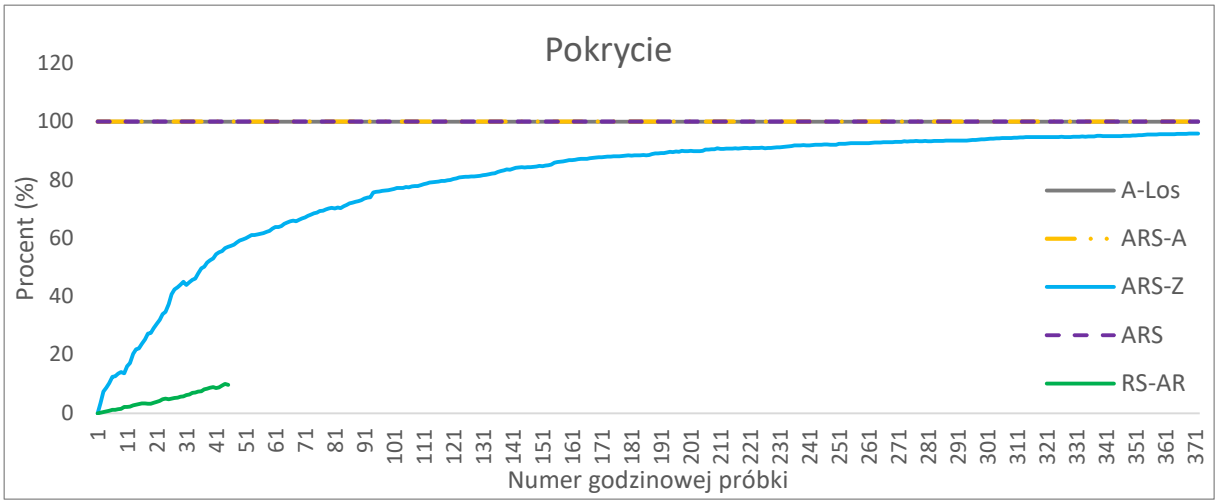
oraz 44 godzinowe próbki eksperymentu dla:

- systemu rekomendacji zbudowanego na bazie algorytmu rekomendacji bazującego na regułach asocjacji (określony skrótem „RS-AR”).



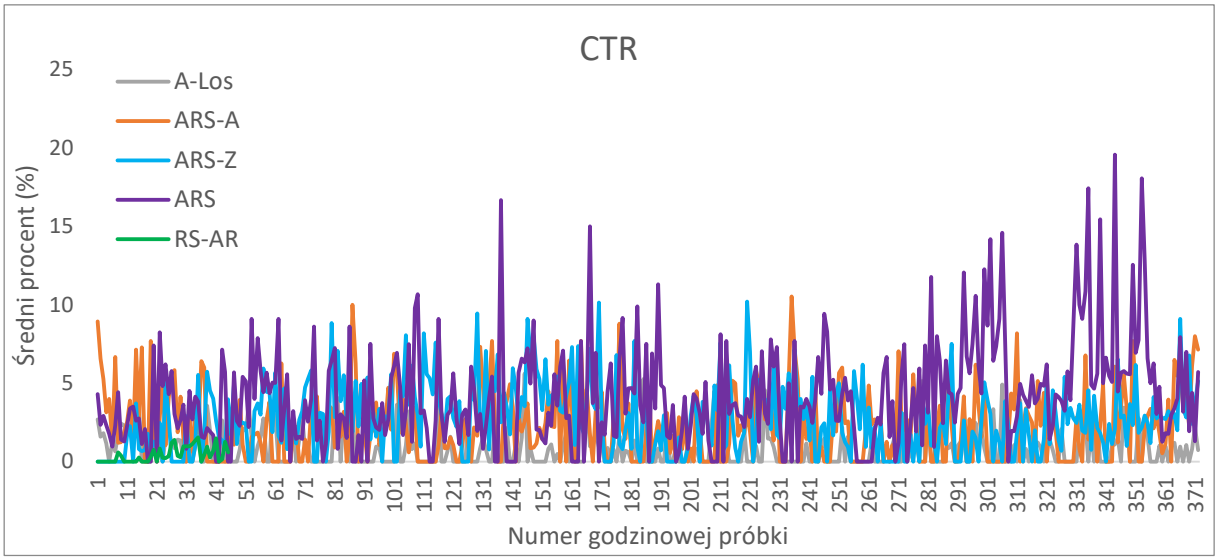
Wykres 7.26. Wykresy dla charakterystyki czas generowania rekomendacji dla różnych systemów rekomendacji.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

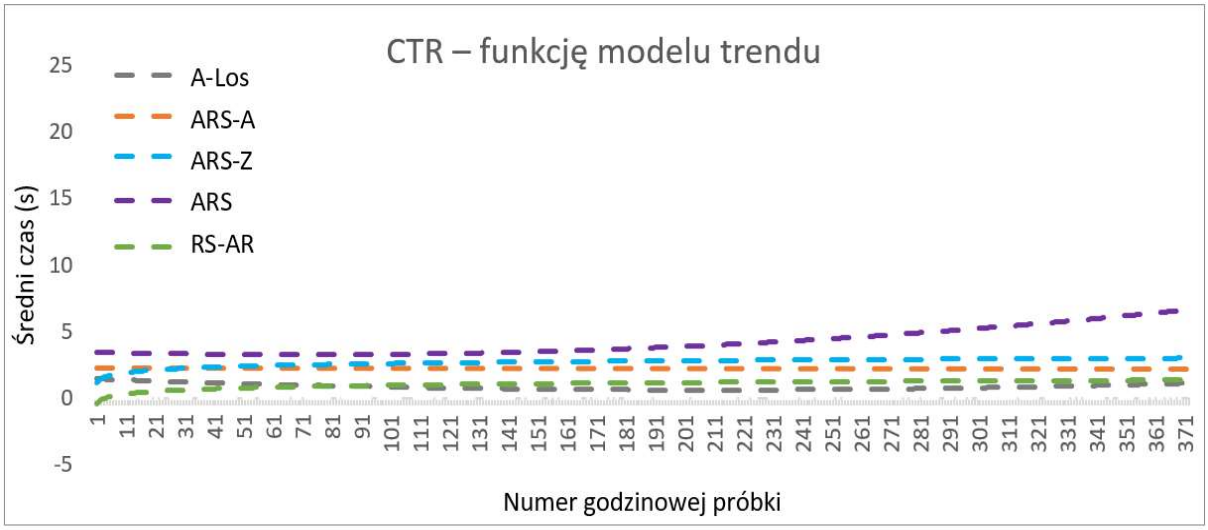
Jak zostało wykazane podczas badań najlepsze właściwości dla charakterystyki czas generowania rekomendacji miał RS-AR. Jednak wyników tych nie można traktować jako w pełni wiarygodnych z powodu krótkiego czasu trwania tego eksperymentu. W przypadku pozostałych eksperymentów wartość tej charakterystyki wahała się w stałych przedziałach od 0,0426s do 0,1033s dla A-Los i od 0,01620s do 0,05300s dla ARS-A oraz miała charakter wzrostowy nieprzekraczający 0,22450s dla ARS-Z i 0,10910s dla ARS. W żadnym przypadku nie doszło do przekroczenia przyjętej górnej granicy wartości miary czasu generowania rekomendacji wynoszącej 1 sekundę. Ciekawą obserwacją jest to, że najmniejsze wartości charakterystyki oprócz RS-AR były przyjmowane dla ARS-A i były średnio mniejsze niż dla A-Los o 0,01951s.



Wykres 7.27. Wykresy dla charakterystyki pokrycie dla różnych systemów rekomendacji.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

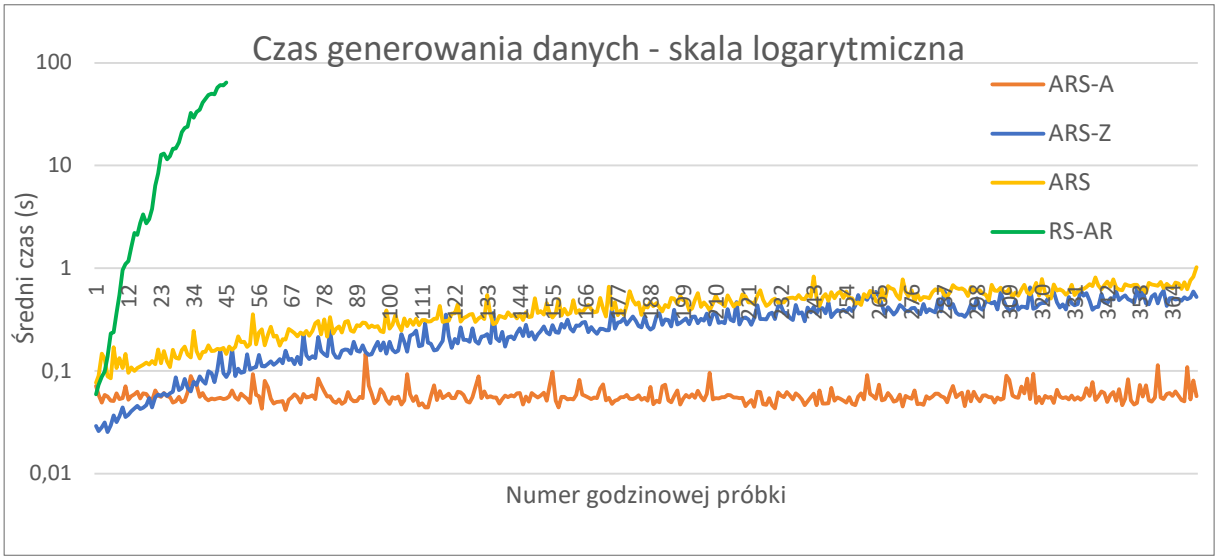
Analogicznie jak dla wcześniejszego podsumowania nie ma wielu wartości dla charakterystyki pokrycie dla RS-AR jednak widać na wykresie, że ma najgorszy trend znacznie odbiegający od przyjętej bazowej wartości wynoszącej 80%. Trend wzrostowy jest również dla ARS-Z, jednak w tym przypadku minimalna wartość bazowa zostaje osiągnięta w 121 godzinowej próbce. Dla pozostałych eksperymentów związanych z A-Los, ARS-A i RS-AR wartości charakterystyki pokrycie przez cały okres trwania eksperymentów przyjmowały najlepszą z możliwych wartości, czyli 100%.





Wykres 7.28. Wykresy dla charakterystyki CTR dla różnych systemów rekomendacji.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Podczas trwania eksperymentów okazało się, że charakterystyka CTR jest bardzo zmienna w czasie i trudna do analizy na podstawie wykresu wartości charakterystyki podczas trwania eksperymentów. Stąd też na podstawie danych wartości zostały oszacowane i pokazane na wspólnym wykresie funkcje modelu trendu dla CTR. Na ich podstawie widać, że najlepszy trend ma ARS, a najgorszy A-Los i RS-AR.



Wykres 7.29. Wykresy dla charakterystyki czas generowania danych dla różnych systemów rekomendacji.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

Istotną charakterystyką z punktu widzenia wydajności systemu teleinformatycznego systemu e-Commerce jest czas generowania danych w postaci czasu generowania grafu G oraz czasu generowania reguł asocjacji. Na wykresie widać, że najgorsze wartości dla tej charakterystyki ma RS-AR. Osiągały one bardzo szybko wartość górnej granicy czasu generowania danych wynoszącą 60s, co uniemożliwiało kontynuowanie eksperymentu. Najlepszy wynik prawie zbliżony do stałej był dla ARS-A. Dla A-Los nie były badane wartości tej charakterystyki, gdyż nie zachodził w ramach systemu rekomendacji podproces generowania danych. Czas generowania danych dla ARS-Z i ARS wrastał podczas trwania eksperymentów, ale nie osiągnął górnej wartości granicznej wynoszącej 60s.

	CTR (%)	**Pokrycie (%)**	**Czas generowania rekomendacji (s)**	**Czas generowania danych (s)**
Wartość bazowa	*0,89754*	*80*	*1*	*60*
A-Los	0,89754	100	0,05122	-
ARS-A	2,63144	100	0,03171	0,05756
ARS-Z	2,90205	79,77578	0,03868	0,29376
ARS	5,21176	100	0,09499	0,52906
RS-AR	0,50282	4,45613	0,00076	16,23778

Tabela 7.12. Średnie wartości charakterystyk dla różnych systemów rekomendacji. Źródło: opracowanie własne na podstawie przeprowadzonych badań

Porównanie średnich wartości badanych charakterystyk wskazuje, że najlepsze wartości dla charakterystyki CTR są osiągane dla ARS 5,21176%. Są one znacznie lepsze niż drugie w kolejności wartości dla ARS-Z wynoszące 2,30971%. Z punktu widzenia e-Commerce to bardzo dobry wynik. Szczególnie w połączeniu z wartością 100% dla pokrycia dla ARS, gdzie w przypadku ARS-Z średnie pokrycie wynosiło 79,77578%. Pozostałe charakterystyki związane z czasem dla systemów z wyjątkiem RS-AR są dobre lub bardzo dobre i znacznie niższe niż górne bazowe wartości dla tych charakterystyk.

Problemy zimnego startu, rzadkości danych i skalowalności były analizowane wraz z wynikami poszczególnych eksperymentów i zbiorczo celem porównania zostały przedstawione w poniższej tabeli.

	Zimny start	Rzadkość danych	Skalowalność
A-Los	brak	brak	dobra
ARS-A	brak	brak	dobra
ARS-Z	podatny	podatny	dobra
ARS	brak	brak	dobra
RS-AR	podatny	podatny	niedobra

Tabela 7.13. Porównanie problemów systemów rekomendacji dla różnych implementacji.
Źródło: opracowanie własne na podstawie przeprowadzonych badań

W przypadku badań nad ww. problemami również rozwiązanie ARS cechuje się bardzo dobrymi właściwościami.

7.3.2. Statystyczna weryfikację hipotezy dla CTR

W ramach analizy danych zgromadzonych podczas eksperymentów dla charakterystyki CTR przeprowadzono statystyczną weryfikację hipotezy, że średnie CTR dla badanych systemów rekomendacji wykorzystujące rozwiązania inne niż algorytm ARS jest mniejsze niż średnie CTR dla systemu rekomendacji bazującego na algorytmie ARS.

W tym celu wykorzystano metodę bazującą na przeprowadzeniu testu statystycznego dla wartości oczekiwanych dwóch populacji, dla których nieznane są rozkłady prawdopodobieństwa i nieznane są odchylenia standardowe, ale są duże liczności prób ($n > 30$). Celem testów tego typu jest sprawdzenie, czy średnie w dwóch populacjach są równe (H_0 – hipoteza zerowa) czy też różnią się (H_1 – hipoteza alternatywna).

Statystyka testu, służąca do testowania hipotezy zerowej $H_0 : \mu_0 = \mu_1$ i hipotezy alternatywnej $H_1 : \mu_o <> \mu_1$ ($\mu_o < \mu_1$) przyjmuje postać (Bobrowski i Maćkowiak-Łybacka 2006):

$$U = \frac{\overline{X}_{n_1} - \overline{X}_{n_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \qquad (7.1)$$

gdzie:

$\overline{X}_{n_k} = \frac{1}{n_k}\sum_{i=1}^{n_k} X_i$ – średnia dla próby k

$S_k^2 = \frac{1}{n_k}\sum_{i=1}^{n_k}(X_i - \overline{X}_k)^2$ – wariancja dla próby k

$k = 1,2$ – numery prób

n – liczność próby

Należy zaznaczyć, że testowano hipotezę H_0 dla porównania średniego CTR dla algorytmu ARS ze wszystkimi pozostałymi algorytmami. Gdzie μ_0 jest średnią CTR dla innych algorytmów, a μ_1 dla algorytmu ARS.

W przypadku hipotezy alternatywnej w postaci $\mu_1 < \mu_2$ stosuje się lewostronny obszar krytyczny: $R_\alpha = (-\infty, u_\alpha)$

Przyjmując jako próbę numer 2 ($k = 2$) wyniki z eksperymentu związanego z algorytmem ARS a jako próbę numer 1 ($k = 1$) odpowiednio wynik z pozostałych eksperymentów wyznaczono wartość statystyki U jako u_0 dla poszczególnych testów. Wyniki obliczeń przedstawiono w poniższej tabeli:

	$\boldsymbol{k}$	$\boldsymbol{n}$	$\boldsymbol{\overline{X}_{n_k}}$	$\boldsymbol{S_k}$	$\boldsymbol{S_k^2}$	$\boldsymbol{u_0}$
A-Los	1	371	0,82868	1,45823	2,12644	-5,77113
ARS-A	1	371	2,21239	4,94810	24,48367	-3,12450
ARS-Z	1	371	2,66599	4,95956	24,59726	-2,40492
RS-AR	1	45	0,47416	0,94675	0,89634	-6,25102
ARS	2	371	4,18451	11,10485	123,31777	

Tabela 7.14. Wyniki oszacowania statystyki U jako u_0 dla testów wartości oczekiwanych populacji względem systemu rekomendacji bazującego na algorytmie ARS.
Źródło: opracowanie własne.

Dla ustalonego poziomu istotności α=0,05 kwantyl rozkładu normalnego $N(0,1)$ wynosi $u_\alpha = -1,6449$. Z uwagi na postać hipotezy alternatywnej lewostronny obszar krytyczny przyjął postać $R_\alpha = (-\infty, -1,6449)$. Ponieważ dla każdego testu otrzymano $u_0 \in R_\alpha$ oznacza to, że na poziomie istotności α=0,05 należy odrzucić hipotezę zerową H_0 wskazującą na równość charakterystyki CTR na korzyść hipotezy alternatywnej H_1 wskazującej na to, że charakterystyka CTR jest największa dla algorytmu rekomendacji bazującego na algorytmie ARS, co potwierdza jakość tego algorytmu względem innych konkurencyjnych mechanizmów.

8. Kierunki rozwoju algorytmu

Algorytm rekomendacji sesji (ARS) w swej podstawowej postaci jest ahistoryczny względem działań użytkowników. Oznacza to, że dla jego funkcjonowania nie ma znaczenia historia zachowań użytkownika. Ponadto, jeśli chodzi o model grafu rekomendacji sesji G, to wszystkie klasy jąder mają takie samo znaczenie. W konsekwencji klasa jąder budowana na podstawie zachowań użytkowników typu 'kliknięcie' ma takie samo znaczenie dla algorytmu jak klasa jąder budowana na podstawie zachowań użytkowników 'zakup'. Co wydaje się, że powinno być zróżnicowane. Zakup powinien wskazywać na to, że obiekt jest bardziej atrakcyjny, a przez to powinien mieć większą wagę rekomendacji niż obiekt, który był tylko oglądany w ramach zachowania 'kliknięcie' (Malinowski 2022).

W celu wyeliminowania tych wad pierwotnego algorytmu ARS można dokonać następujących modyfikacji:

8.1. Modyfikacja 1 - dodanie wag do łuków

W ramach tej modyfikacji na etapie budowy grafu G należy związać z łukami wychodzącymi z jąder poszczególnych klas funkcję wag. Dzięki takiemu zabiegowi zostanie oddana właściwość, że wybrane klasy jąder, jak na przykład rekomendacje ekspertów lub zakupy, są bardziej wartościowe niż przypadkowe kliknięcia anonimowych użytkowników.

W przypadku tej modyfikacji należy rozpatrywany model matematyczny grafu rekomendacji sesji G uzupełnić o funkcję w przyporządkowującą każdemu łukowi liczbę naturalną interpretowaną jako waga łuku klasy w postaci:

$$w: E \to \mathrm{N}\backslash\{0\} \qquad (8.1)$$

gdzie:

E_K - zbiór łuków wychodzących z jąder klasy K

$f: K \to \mathrm{N}\backslash\{0\}$ – funkcja wagi klasy

$\forall_{e,f \in E_k \; i \; k \in K} \; w(e) = w(f) \; \wedge \; w(e) = f(k)$ – wagi łuków wychodzących z jąder tej samej klasy są sobie równe i równe wadze klasy jąder, z których wychodzą.

W związku z powyższym graf rekomendacji sesji G staje się grafem ważonym i przyjmuje postać trójki uporządkowanej (Wojciechowski i Pieńkosz 2013):

$$G =< N, E, w > \qquad (8.2)$$

gdzie:

N – zbiór wierzchołków zgodny z pierwotnym modelem grafu rekomendacji sesji G

E – zbiór łuków zgodny z pierwotnym modelem grafu rekomendacji sesji G

w – funkcja wag określona na łukach (waga łuku klasy)

W konsekwencji uzupełnienia modelu o wagi klas i wagi łuków krok *S04* algorytmu przyjmuje postać:

S04': Oszacowanie stopni wchodzących dla każdego węzła będącego obiektem podgrafu G''_m poprzez sumę wag łuków wchodzących do tego węzła.

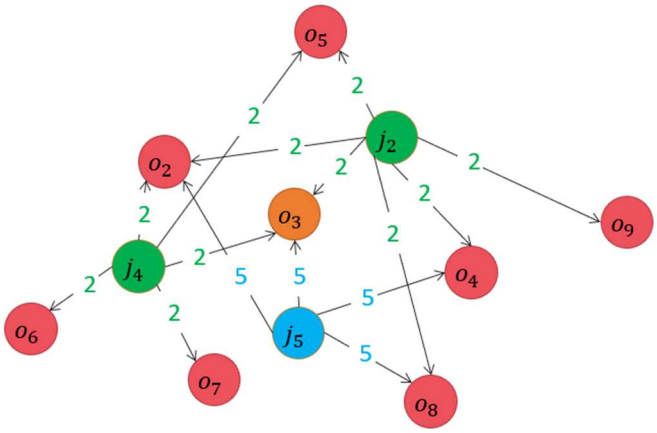
Przykład modelu z wagami łuków dla klas K_1 i K_2 i dla wartości funkcji wagi klas określonej w postaci:

$$f(k) = \begin{cases} 2\ dla\ k = K_1 \\ 5\ dla\ k = K_2 \end{cases}$$

czyli funkcja wag łuków klas przyjmuje postać:

$$w(e) = \begin{cases} 2\ dla\ e \in E_{K_1} \\ 5\ dla\ e \in E_{K_2} \end{cases}$$

Dla grafu G przedstawionego poniżej stopień wchodzący węzła o_3 obliczany w ramach kroku S04' algorytmu ma wartość 2+2+5=9 gdzie uprzednio byłoby to 1+1+1=3.



Rysunek 8.1. Ważony graf rekomendacji sesji G z dwoma klasami jąder $K_1 = \{j2, j4\}$ i $K_2 = \{j5\}$
Źródło: opracowanie własne.

8.2. Modyfikacja 2 - konstruowanie ścieżek użytkownika

W celu wyeliminowania ahistoryczności w ramach modyfikacji należy wprowadzić do algorytmu ARS zmienną pomocniczą w postaci zbioru M, w którym przechowywane są obiekty związane z zachowaniem użytkownika 'kliknięcie' (w produkt na stronie WWW). Pierwszym elementem tego zbioru jest obiekt m. Kolejnymi obiektami dodawanymi do zbioru M są wybierane przez użytkownika ‘kliknięte' w ramach sesji nowe obiekty (odwiedzone strony WWW produktów). Kroki algorytmu są wówczas zależne nie od pojedynczego obiektu m, lecz od zbioru obiektów $M = \{m_1, m_2, m_3, m_4, \dots, m_n\}$.

W związku z powyższym pierwotny algorytm zostaje zmodyfikowany w następujący sposób:

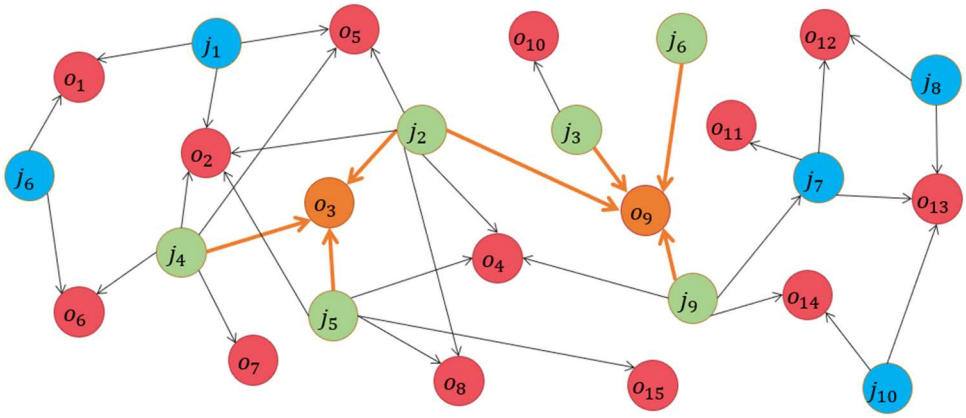
- Dodanie nowego kroku S01a:

  S01a*:* Dodanie do zbioru M obiektu m
- Krok S02 przyjmuje postać:

  S02’*:* Budowa podgrafu G'_m grafu G złożonego z węzłów zbioru M i węzłów sąsiadujących z węzłami zbioru M oraz łuków pomiędzy nimi a węzłami zbioru M
- Krok S06 przyjmuje postać:

  S06’*:* Zapisanie w wektorze R_m posortowanych obiektów bez obiektów zbioru M
- Dodanie nowego kroku S07:

S07: Odczyt wybranego przez użytkownika 'kliknięgo' obiektu $m' \in O$ i dodanie go do zbioru M



Rysunek 8.2. Fragment modelu grafu G ze zbiorem $M = \{o3, o9\}$.
Źródło: opracowanie własne.

Przykład modelu ze zbiorem M przedstawia powyższy rysunek. W tym wypadku w wyniku kroku S02' powstanie podgraf G'_m grafu G składający się ze zbioru węzłów $N' = \{o3, o9, j2, j4, j5, j3, j6, j9\}$ oraz wszystkie łuki pomiędzy nimi (oznaczone na pomarańczowo).

Podsumowanie

Powyżej przedstawione i opisane zostały tylko wybrane z możliwych modyfikacji algorytmu ARS. W toku prowadzonych badań i analiz algorytmu pojawiły się dodatkowo idee modyfikacji w takich kierunkach jak:

- dodanie kolejnej warstwy węzłów sesji i jąder - podgraf G'''_m;
- konstruowanie ścieżek rekomendacji. Wynikiem algorytmu byłaby wówczas ścieżka odwiedzin rekomendowanych obiektów kończąca się satysfakcjonującym zachowaniem (dodanie do koszyka, zakup);
- wyznaczenie optymalnych przedziałów czasu z punktu widzenia maksymalizacji CTR dla gromadzenia informacji o zachowaniach użytkowników (np.: ostatni miesiąc, ostatnie 3 miesiące itd.…)

Mogą one stanowić podstawę do kolejnych badań nad algorytmem ARS i jego wykorzystaniem w systemach rekomendacji.

9. Zakończenie

W rozprawie badano zagadnienia związane z systemami rekomendacji w obszarze e-Commerce. W szczególności dotyczyły one autorskiego algorytmu bazującego na sesjach rekomendacji tworzonych na podstawie zachowań użytkowników oraz atrybutów obiektów w systemie e-Commerce.

Problem rozważano z uwzględnieniem ograniczonych informacji opisujących użytkowników. Zaproponowano podejście rekomendacji bazującej na sesjach użytkowników, gdzie sam użytkownik nie jest identyfikowany, a o jego preferencjach świadczą zachowania podczas jego sesji. Podejście takie jest szczególnie istotne w kontekście aktualnie funkcjonujących przepisów prawa związanych z RODO i GIODO. Przepisy te wpływają na ograniczenie zdolności gromadzenia danych przez systemy informatyczne na temat użytkowników, co bardzo utrudnia szacowanie rekomendacji dla wielu algorytmów.

Tradycyjne techniki rekomendacji opisane w literaturze opierają się w dużej mierze na informacjach opisujących użytkowników takich jak ich dane demograficzne, preferencje, oceny produktów lub relacje z innymi użytkownikami. Jeśli chodzi o wykorzystanie zachowań użytkowników, to w literaturze można znaleźć przede wszystkim informacje na temat analizy asocjacji, której celem jest kojarzenie (asocjacja) wzorców zakupowych klientów na bazie danych transakcji zakupowych (koszyk zakupów).

Z zastosowaniem mechanizmów rekomendacji bazujących na zachowaniach wiąże się problem zimnego startu i rzadkości danych. Rozwiązaniem tych problemów, proponowanym w rozprawie, jest podejście w postaci autorskiego algorytmu ARS, który w swej idei dodatkowo oprócz bazowania na zachowaniach użytkowników, bazuje również na atrybutach obiektów (produktów lub usług), które to występują na starcie systemu oraz które posiadają każdy nowy obiekt dodawany do systemu. Takie podejście pozwoliło ograniczyć problemy zimnego startu i rzadkości danych.

Zaprezentowany algorytm ARS może być zaliczony do stosunkowo nowych technik rekomendacji klasyfikowanych jako bazujący na sesjach oraz bazujący na grafach heterogenicznych.

W ramach badań przeprowadzono szereg eksperymentów z różnymi formami algorytmu ARS oraz konkurencyjnym rozwiązaniem w postaci algorytmu rekomendacji bazującym na regułach asocjacji. Eksperymenty były prowadzone środowisku badawczym zaimplementowanym w funkcjonującym online systemie e-Commerce. Badania wykazały, że system rekomendacji bazujący na autorskim algorytmie ARS ma znacznie lepsze właściwości niż konkurencyjne rozwiązanie i najlepiej z porównywanych mechanizmów rozwiązuję problem badawczy rekomendacji. Wyznacznikiem jakości rozwiązania problemu była charakterystyka CTR zwana współczynnikiem kliknięć. Ponadto, najlepszy wynik CTR dla algorytmu ARS został potwierdzony metodą bazującą na przeprowadzeniu testu statystycznego dla wartości oczekiwanej populacji.

Analiza danych zebranych podczas eksperymentów wykazała, że jest silnie dodatnia korelacja pomiędzy charakterystykami czasowymi takimi jak: czas generowania grafu G, czas generowania reguł asocjacji, czas generowanie rekomendacji a wielkością danych w postaci liczby łuków grafu G lub liczby reguł asocjacji. Natomiast ww. wielkości słabo korelują się z charakterystyką CTR.

Dodatkowo eksperymenty prowadzone podczas badania wykazały, że pełen algorytm ARS ma dobre wartości charakterystyk związanych z czasem generowania rekomendacji i czasem generowania danych. Badania te wskazują na to, że system rekomendacji budowany na bazie algorytmu ARS jest dobrze skalowalny czyli działa efektywnie przy zwiększającej się z czasem wielkości danych. Taka cecha algorytmu powoduję, że może on być z powodzeniem stosowany w rozwiązaniach e-Commerce w szczególności rozwiązaniach online.

Co istotne, złożoności obliczeniowa i pamięciowa badanych algorytmów oszacowana teoretycznie oraz przedstawione na bazie literatury zostały potwierdzone w ramach przeprowadzonych eksperymentów.

Należy nadmienić, że z punktu widzenia rozwiązań e-Commerce, system rekomendacji zbudowany na bazie algorytmu ARS „sam się uczy” zachowań użytkowników wynikających z trendów takich jak na przykład moda, specjalne okazje, zmienna w czasie popularność wybranych produktów, itd.…

Na zakończenie, w dalszych badaniach dotyczących algorytmu ARS zasadne byłoby przeprowadzenie następujących działań:

- zbadanie mechanizmów usuwających problem ahistoryczności algorytmu;
- opracowanie mechanizmów usuwających problem równej wagi dla różnych klas zachowań użytkowników;
- zaimplementowanie dodawania kolejnej warstwy węzłów sesji i jąder - podgraf G'''_m;
- wyznaczenie optymalnych przedziałów czasu z punktu widzenia maksymalizacji CTR dla gromadzenia informacji o zachowaniach użytkowników (np.: ostatni miesiąc, ostatnie 3 miesiące itd.…);
- zbadanie wpływu botów na odbiór zachowań użytkowników przez systemy rekomendacji;
- określenie wpływu grup atrybutów obiektów na efektywność systemów rekomendacji.

10. Streszczenie

Tematem pracy jest algorytm rekomendacji bazujący na sesjach rekomendacji. W rozprawie zaproponowano podejście rekomendacji hybrydowej bazującej na sesjach użytkowników, w których są wyodrębniane określone zachowania oraz na atrybutach obiektów (produktów, usług) zawartych w bazie danych systemu e-Commerce. Istotnym założeniem dla badanego problemu jest fakt braku identyfikacji użytkownika umożliwiającej jego opisanie w postaci danych demograficznych lub historycznych preferencji.

W ramach pracy został opracowany model matematyczny danych grafu sesji rekomendacji G i model sesji rekomendacji, na których bazuje autorski algorytm rekomendacji ARS. Dodatkowo została przedstawiona jego implementacja w działającym online systemie e-Commerce.

Ponadto, w celu porównania autorskiego algorytmu ARS został opisany i zaimplementowany konkurencyjny algorytm wykorzystujący zachowania użytkowników bazujący na regułach asocjacji.

W ramach badań zostały przeprowadzone eksperymenty, których celem była weryfikacja przydatności zaimplementowanych algorytmów dla celów e-Commerce w obszarze rekomendacji i porównanie ich między sobą.

Eksperymenty były prowadzone w pełni funkcjonującym online środowisku e- Commerce. Do badań wykorzystano obserwację systemu rekomendacji i analizę danych rzeczywistych. Następnie na bazie zgromadzonych danych dokonano analizy i porównania. Wyniki potwierdziły użyteczność autorskiego algorytmu dla rozwiązań e- Commerce i jego przewagę nad konkurencyjnym rozwiązaniem.

11. Abstract

The subject of this paper is a recommendation algorithm based on recommendation sessions. The paper proposes a hybrid recommendation approach based on user sessions, in which specific behaviours are extracted, and on the attributes of objects (products, services) contained in the database of the e-Commerce system. An important assumption of the analysed problem studied is fact that there is no identification of the user to describe him in the form of demographic data or historical preferences.

Within the scope of the study, a mathematical model of G recommendation session graph data and a session model were developed, on which the author's recommendation algorithm is based. In addition, its implementation was presented in the operational online e-Commerce system.

Furthermore, in order to compare the original ARS algorithm, a competitive algorithm using user behaviour based on association rules has been described and implemented.

As part of the research, experiments were conducted to verify the suitability of the implemented algorithms for e-Commerce purposes in the area of recommendations and to compare them with each other.

The experiments were conducted in a fully functioning online e-Commerce environment. The research used observation of the recommendation system and analysis of real data. Analysis and comparison were then carried out on the basis of the collected data. The results confirmed the usability of the proprietary algorithm for e-Commerce solutions and its superiority over competitive solutions.

12. Bibliografia

1. Adomavicius, Gediminas, Tuzhilin, Alexander (2005), 'Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions', *IEEE Transactions on Knowledge and Data Engineering.*:
2. Agrawal, Rakesh, Imieliński, Tomasz, Swami, Arun (1993), 'Mining Association Rules Between Sets of Items in Large Databases', *ACM SIGMOD Record*: 22 (2), ss. 207-216.
3. Agrawal, Rakesh, Srikant, Ramakrishnan (1994), 'Fast Algorithms for Mining Association Rules in Large Databases', *Proc. of the 20th International Conference on Very Large Data Bases (VLDB'94)*, .
4. Ahmad, Irfan (2021), 'How Much Data Is Generated Every Minute? [Infographic] | Social Media Today', online: https://www.socialmediatoday.com/news/how-much-data-is-generated-every-minute-infographic-1/525692/ [dostęp: 22.03.2022].
5. Andrzejewski, Witold (2014), 'Odkrywanie Asocjacji', *Politechnika Poznańska, Wydział Informatyki.*:
6. Bartczak, Krzysztof (2016), 'Bariery rozwojowe handlu elektronicznego', .
7. Bartram, Finn (2020), 'List Of 15 Product Attribute Examples & Types - The Ecomm Manager', w: *The Ecomm Manager*, online: https://theecommmanager.com/product-attributes/ [dostęp: 30.03.2022].
8. Beel, Joeran, Dinesh, Siddharth, Mayr, Philipp, Carevic, Zeljko, Raghvendra, Jain (2017), 'Stereotype and Most-Popular Recommendations in the Digital Library Sowiport', *Proceedings of the 15th International Symposium of Information Science (ISI 2017)*:(January), ss. 96-108, online: http://beel.org/publications/2017 ISI -- Stereotype and Most-Popular Recommendations in Sowiport -- preprint.pdf.
9. Beel, Joeran, Langer, Stefan (2015), 'A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: 9316 (September), ss. 153-168.
10. Beel, Joeran, Langer, Stefan, Genzmehr, Marcel (2013), 'Sponsored vs. Organic (Research Paper) Recommendations and the Impact of Labeling', *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013).*:
11. Beel, Joeran, Nürnberger, Andreas, Langer, Stefan (2013), 'Persistence in Recommender Systems: Giving the Same Recommendations to the Same Users Multiple Times', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: 8092 LNCS (September),.
12. Ben Fraj, Mohtadi (2018), 'Graph based recommendation engine for Amazon products', w: *Towards Data Science*, online: https://towardsdatascience.com/graph-based-recommendation-engine-for-amazon-products-1a373e639263 [dostęp: 25.03.2022].
13. Berkovsky, Shlomo, Kuflik, Tsvi, Ricci, Francesco (2008), 'Mediation of user models for enhanced personalization in recommender systems', *User Modeling and User-Adapted Interaction*: 18 (3), ss. 245-286.
14. Berkovsky, Shlomo, Kuflik, Tsvi, Ricci, Francesco (2009), 'Cross-representation mediation of user models', *User Modeling and User-Adapted Interaction*: 19 (1-2 SPEC. ISS.), ss. 35-63.
15. Bhaskar, Karthik, Kundur, Deepa, Lawryshyn, Yuri (2020), 'Implicit Feedback Deep Collaborative Filtering Product Recommendation System', *University ofToronto*, http://arxiv.org/abs/2009.08950 [dostęp: .

16. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A. (2013), 'Recommender systems survey', *Knowledge-Based Systems*: 46 ss. 109-132, online: https://dl.acm.org/doi/abs/10.1016/j.knosys.2013.03.012 [dostęp: 21.03.2022].

17. Bobrowski, Dobiesław, Maćkowiak-Łybacka, Krystyna (2006), 'Wybrane metody wnioskowania statystycznego', .

18. Bogárdi-Mészöly, Agnes, Szitás, Zoltán, Levendovszky, Tihamér, Charaf, Hassan (2005), 'Investigating factors influencing the response time in ASP.NET web applications', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, .

19. Bouraga, Sarah, Jureta, Ivan, Faulkner, Stéphane, Herssens, Caroline (2014), 'Knowledge-based recommendation systems:A survey', *International Journal of Intelligent Information Technologies*: 10 (2), ss. 1-19.

20. Burke, Robin (2007), 'Hybrid Web Recommender Systems', *The Adaptive Web: Methods and Strategies of Web Personalization*, .

21. Cho, Yoon Ho, Kimb, Jae Kyeong, Hie Kima, Soung (2002), 'A personalized recommender system based on web usage mining and decision tree induction', *Expert Systems with Applications*: 44 (1), ss. 329–342.

22. Cormen, Thomas H., Charles, E. Leiserson, Rivest, Ronald L., Clifford, Stein (2009), 'Introduction to algorithms', *Bioinformatics: A Concept-Based Introduction*, .

23. Cremonesi, Paolo, Koren, Yehuda, Turrin, Roberto (2010), 'Performance of recommender algorithms on top-N recommendation tasks', *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*:(September), ss. 39-46.

24. Fayyaz, Zeshan, Ebrahimian, Mahsa, Nawara, Dina, Ibrahim, Ahmed, Kashef, Rasha (2020), 'Recommendation systems: Algorithms, challenges, metrics, and business opportunities', *Applied Sciences (Switzerland)*: 10 (21), ss. 1-20.

25. Fischer, Gerhard (2001), 'User Modeling in Human–Computer Interaction', *User Modeling and User-Adapted Interaction 2001 11:1*: 11 (1), ss. 65-86, online: https://link.springer.com/article/10.1023/A:1011145532042 [dostęp: 24.03.2022].

26. Fuchs, Christoph, Prandelli, Emanuela, Schreier, Martin (2010), 'The psychological effects of empowerment strategies on consumers' product demand', *Journal of Marketing*: 74 (1), ss. 65-79.

27. Gemius, Izba Gospodarki Elektronicznej (2020), 'E-commerce w Polsce 2020 Gemius dla e-Commerce Polska', ss. 11-16.

28. Glynn, Peter, Bruce, Jon-Paul, O'Connor, John, Celik, Sara (2020), 'Milliseconds make Millions', *Deloitte.Digital*, .

29. Goodrich, Michael T., Tamassia, Roberto (2002), 'Algorithm Design: Foundations, Analysis, and Internet Examples', *4.5 Bucket-Sort and Radix-Sort*, .

30. Google (2022), 'About PageSpeed Insights | Google Developers', online: https://developers.google.com/speed/docs/insights/v5/about#crux [dostęp: 20.06.2022].

31. Grachev, Gennady A. (2020), 'Pareto ratio and Pareto principle', *Preprints.ru.*:

32. Gupta, Anjali (2014), 'E-Commerce : Role of E-Commerce in Today's Business', *International Journal of Computing and Corporate Research*: 4 (1), online: http://journal.stainkudus.ac.id/index.php/equilibrium/article/view/1268/1127.

33. Han, Jiawei, Kamber, Micheline, Pei, Jian (2012), 'Data Mining: Concepts and Techniques', *Data Mining: Concepts and Techniques*, .

34. Han, Jiawei, Pei, Jian, Yin, Yiwen (1999), 'Mining frequent patterns without candidate

generation', *SIGMOD Record (ACM Special Interest Group on Management of Data)*: 29 (2),.

35. He, Qi, Jiang, Daxin, Liao, Zhen, Hoi, Steven C. H., Chang, Kuiyu, Lim, Ee Peng, Li, Hang (2009), 'Web query recommendation via sequential query prediction', *Proceedings - International Conference on Data Engineering*:(May 2014), ss. 1443-1454.
36. Hekmatfar, Taher, Haratizadeh, Saman, Goliaei, Sama (2021), 'Embedding ranking-oriented recommender system graphs', *Expert Systems with Applications*: 181 (July 2020),.
37. Hidasi, Balázs, Karatzoglou, Alexandros, Baltrunas, Linas, Tikk, Domonkos (2016), 'Session-based recommendations with recurrent neural networks', *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*:(November),.
38. Hikmawati, Erna, Maulidevi, Nur Ulfa, Surendro, Kridanto (2021), 'Minimum threshold determination method based on dataset characteristics in association rule mining', *Journal of Big Data*: 8 (1),.
39. Horzyk, Adrian (2021), 'WSTĘP DO INFORMATYKI Grafy i struktury grafowe', w: *Akademia Górniczo-Hutnicza*, online: www.agh.edu.pl [dostęp: 25.03.2022].
40. Huang, Xin, Kuijpers, Dymfke, Li, Lavonda, Sha, Sha, Xia, Chenan (2020), 'How Chinese consumers are changing shopping habits in response to COVID-19', *McKinsey & Company*: May.
41. Interactive Advertising Bureau (2017), 'Prywatność w sieci 2016/2017', .
42. Isinkaye, F. O., Folajimi, Y. O., Ojokoh, B. A. (2015), 'Recommendation systems: Principles, methods and evaluation', *Egyptian Informatics Journal*: 16 (3), ss. 261-273.
43. Jaiswal, Shefali, Singh, Anurag (2020), 'Influence of the Determinants of Online Customer Experience on Online Customer Satisfaction', *Paradigm*: 24 (1), ss. 41-55.
44. Jang, Se Won, Kim, Simon, Ha, JeongWoo (2006), 'Graph-based Recommendation Systems: Comparison Analysis between Traditional Clustering Techniques and Neural Embedding', *Journal of Dermatological Science*: 41 (2),.
45. Joeran, Beel, Stefan, Langer, Andreas, Nürnberger, Marcel, Genzmehr (2013), 'The Impact of Demographics (Age and Gender) and Other User-Characteristics on Evaluating Recommender Systems', *International Conference on Theory and Practice of Digital Libraries.*:
46. Kamehkhosh, Iman, Jannach, Dietmar, Ludewig, Malte (2017), 'A comparison of frequent paern techniques and a deep learning method for session-based recommendation', *CEUR Workshop Proceedings*: 1922 ss. 50-56.
47. Kuanr, Madhusree, Mohapatra, Puspanjali (2021), 'Assessment Methods for Evaluation of Recommender Systems: A Survey', *Foundations of Computing and Decision Sciences*: 46 (4), ss. 393-421.
48. Kumar, Mallari Vijay, Kumar, P. N. V. S. Pava. (2019), 'A study on different phases and various recommendation system techniques', *International Journal of Recent Technology and Engineering*: 7 (5), ss. 38-41.
49. Larose, Daniel T. (2006), 'Odkrywanie wiedzy z danych. Wprowadzenie do eksploracji danych', .
50. Lee, Juhnyoung, Podlaseck, Mark, Schonberg, Edith, Hoch, Robert (2000), 'Understanding Merchandising Effectiveness of Online Stores', *Electronic Markets*, , https://www.tandfonline.com/doi/abs/10.1080/10196780050033944 (dostęp: .

51. Lee, Juhnyoung, Podlaseck, Mark, Schonberg, Edith, Hoch, Robert (2001), 'Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising', *Data Mining and Knowledge Discovery*:(5), ss. 59-84.

52. Lerche, Lukas, Jannach, Dietmar, Ludewig, Malte (2016), 'On the value of reminders within e-commerce recommendations', *UMAP 2016 - Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*: ss. 27-35.

53. Li, Hui, Cai, Fei, Liao, Zhifang (2012), 'Content-based filtering recommendation algorithm using HMM', *Proceedings - 4th International Conference on Computational and Information Sciences, ICCIS 2012*:(August 2012), ss. 275-277.

54. Liling, Liu (2019), 'Summary of recommendation system development', *Journal of Physics: Conference Series*: 1187 (5),.

55. Liu, Siyi, Zheng, Yujia (2020), 'Long-tail Session-based Recommendation', *RecSys 2020 - 14th ACM Conference on Recommender Systems*: ss. 509-514.

56. Loisel, Jérôme (2001), 'Response time is critical for E-Commerce', w: *OctoPerf*, online: https://octoperf.com/blog/2015/06/17/response-time-e-commerce/#page-speed-influence [dostęp: 31.03.2022].

57. Loshin, David (2013), 'Business Intelligence: The Savvy Manager's Guide', *Morgan Kauf*.

58. Ludewig, Malte, Jannach, Dietmar (2018), 'Evaluation of session-based recommendation algorithms', *User Modeling and User-Adapted Interaction*: 28 (4-5), ss. 331-390.

59. Malinowski, Michał (2020), 'Recommendation Algorithm Based on Recommendation Sessions', *Proceedings of the 36th International Business Information Management Association (IBIMA)*:(November), ss. 11260-11272, online: https://ibima.org/accepted-paper/recommendation-algorithm-based-on-recommendation-sessions/.

60. Malinowski, Michał (2021)a, 'Zastosowanie grafów i sieci w systemach rekomendacji', w: Piotr Tomski, Katarzyna Olejniczak-Szuster (red.), *Przedsiębiorstwo w nowej rzeczywistości gospodarczej. Relacje – zmiany – strategie*, t. I, Częstochowa: Wydawnictwo Politechniki Częstochowskiej, ss.77-96.

61. Malinowski, Michał (2021)b, 'Implementation of Recommendation Algorithm Based on Recommendation Sessions in e-Commerce IT system', *7th International Conference on Second Language Studies (ICMS-2021)*: 2021 (19), ss. 14-32, online: https://eurokd.com/Resources/Uploaded/20210604162052ICMS_2021_0357_Presentation Algorithm ARS.pdf?CT=application_pdf.png.

62. Malinowski, Michał (2022), 'Algorytm rekomendacji bazujący na sesjach rekomendacji', w: Ewa Głowacka, Weronika Kortas (red.), *Architektura Informacji – Badania i Praktyka*, Toruń: Wydawnictwo Naukowe UMK, ss.83-97.

63. Malinowski, Michał, Krysiński, Stanisław (2020), 'Techniki rekomendacyjne we współczesnym marketingu', *Przedsiebiorstwo Przyszłości - Kwartalnik Uczelni Techniczno-Handlowej im. Heleny Chodkowskiej ISSN: 2080-8461*: 3 (44), ss. 74-96.

64. Malinowski, Michał, Krysiński, Stanisław (2021), 'Techniki informacyjne siłą napędową współczesnego marketingu', *Przedsiebiorstwo Przyszłości - Kwartalnik Uczelni Techniczno-Handlowej im. Heleny Chodkowskiej ISSN: 2080-8461*: 1 (46), ss. 73-94.

65. Malinowski, Michał, Sokólski, Michał (2001), 'Internet dla lekarzy', .

66. McAuley, Julian, Leskovec, Jure (2013), 'Hidden factors and hidden topics: Understanding rating dimensions with review text', *RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems*, .

67. Miyahara, Koji, Pazzani, Michael J. (2000), 'Collaborative filtering with the Simple

Bayesian Classifier', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, .

68. Montaner, Miquel, López, Beatriz, de la Rosa, Josep Lluís (2002), 'Developing trust in recommender agents', *Universitat de Girona*:(May 2014), ss. 304.
69. Morzy, Tadeusz (2022), 'Eksploracja danych : metody i algorytmy', .
70. Nakhaeizadeh, Gholamreza, Hipp, Jochen, Güntzer, Ulrich (2000), 'Algorithms for association rule mining - a general survey and comparison', *ACM sigkdd explorations newsletter*: 2 (1), ss. 58-64.
71. Nandini, Manchi, Sravya, Aaki Rupa, Swamy, Rama (2018), 'A Literature Survey on Recommender Systems', *International Journal of Research Studies in Science, Engineering and Technology*: 5 (12), ss. 64-71.
72. Nanou, Theodora, Lekakos, George, Fouskas, Konstantinos (2010), 'The effects of recommendations' presentation on persuasion and satisfaction in a movie recommender system', *Multimedia Systems*: 16 (4-5), ss. 219-230.
73. Neal, M. (2011), 'Beyond trust: Psychological considerations for recommender systems', *Proceedings of the 2011 EEE International …*:, online: http://worldcomp-proceedings.com/proc/p2011/EEE3995.pdf.
74. Niklaus, Wirth (2002), 'Algorytmy + struktury danych= programy', .
75. Norris, J. R. (1997), 'Markov Chains', .
76. Oracle (2022)a, 'MySQL :: MySQL 8.0 Reference Manual :: 8.3.9 Comparison of B-Tree and Hash Indexes', online: https://dev.mysql.com/doc/refman/8.0/en/index-btree-hash.html [dostęp: 07.02.2022].
77. Oracle (2022)b, 'MySQL :: MySQL Internals Manual :: 7.2.4 ORDER BY Clauses', online: https://dev.mysql.com/doc/internals/en/optimizer-order-by-clauses.html [dostęp: 07.02.2022].
78. Osadchiy, Timur, Poliakov, Ivan, Olivier, Patrick, Rowland, Maisie, Foster, Emma (2018), 'Recommender system based on pairwise association rules', *Expert Systems with Applications*: 115 ss. 535-542, online: https://doi.org/10.1016/j.eswa.2018.07.077.
79. Park, Young (2010), 'Recommender systems: An overview', *Encyclopedia of E-Business Development and Management in the Global Economy*: 3 ss. 1221-1230.
80. Pęczkowski M., Lasek M. (2013), 'Analiza asocjacji i reguły asocjacyjne w badaniu wyborów zajęć dydaktycznych dokonywanych przez studentów. Zastosowanie algorytmu Apriori', *Ekonomia. Rynek. Gospodarka. Społeczeństwo*:(2013),.
81. Poggi, Nicolas, Carrera, David, Gavaldà, Ricard, Ayguadé, Eduard, Torres, Jordi (2014), 'A methodology for the evaluation of high response time on E-commerce users and sales', *Information Systems Frontiers*: 16 (5), ss. 867-885.
82. Pondel, Maciej, Korczak, Jerzy (2017), 'Eksploracja danych transakcyjnych sklepu internetowego', *Zeszyty Naukowe Politechniki Częstochowskiej. Zarządzanie*: 26 (26), ss. 132-145.
83. Pu, Pearl, Chen, Li, Hu, Rong (2012), 'Evaluating recommender systems from the user's perspective: Survey of the state of the art', *User Modeling and User-Adapted Interaction*: 22 (4-5), ss. 317-355.
84. PWN, Wydawnictwo Naukowe (2022), 'heterogeniczność - definicja, synonimy, przykłady użycia', online: https://sjp.pwn.pl/szukaj/heterogeniczność.html [dostęp: 24.03.2022].
85. Quadrana, Massimo, Cremonesi, Paolo, Jannach, Dietmar (2018), 'Sequence-Aware

Recommender Systems', *ACM Computing Surveys*: 1 (1),.

86. Raeder, Troy, Chawla, Nitesh V. (2011), 'Market basket analysis with networks', *Social Network Analysis and Mining*, .

87. Ramlatchan, Andy, Yang, Mengyun, Liu, Quan, Li, Min, Wang, Jianxin, Li, Yaohang (2018), 'A survey of matrix completion methods for recommendation systems', *Big Data Mining and Analytics.*:

88. Rao, Abishek B., Kiran, Jammula Surya, G, Poornalatha (2021), 'Application of market–basket analysis on healthcare', *International Journal of Systems Assurance Engineering and Management*:, online: https://doi.org/10.1007/s13198-021-01298-2.

89. Resnick, Paul, Varian, Hal R. (1997), 'Recommender systems', *Communications of the ACM*: 40 (3), ss. 56-58, online: https://dl.acm.org/doi/abs/10.1145/245108.245121 [dostęp: 18.03.2022].

90. Ricci, Francesco, Rokach, Lior, Shapira, Bracha (2015), 'Recommender Systems Handbook', *Recommender Systems Handbook*, .

91. Rong, Honghui, Zhu, Wenjun, Zhu, Cui (2022), 'Graph hierarchical dwell-time attention network for session-based recommendation', *ITM Web of Conferences*: 02032 (47), ss. 1-10.

92. Rowell, Eric (2013), 'Big-O Algorithm Complexity Cheat Sheet (Know Thy Complexities!) @ericdrowell', online: https://www.bigocheatsheet.com/ [dostęp: 07.02.2022].

93. Sarwar, Badrul, Karypis, George, Konstan, Joseph, Riedl, John (2000), 'Application of Dimensionality Reduction in Recommender System A Case Study Technical Report CS-TR 00-043', *Computer Science and Engineering Dept., University of Minnesota*:(August),.

94. Sarwar, Badrul, Karypis, George, Konstan, Joseph, Riedl, John (2001), 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the 10th International Conference on World Wide Web, WWW 2001*:(August), ss. 285-295.

95. Shams, Bita, Haratizadeh, Saman (2017), 'Graph-based collaborative ranking', *Expert Systems with Applications*: 67 (April 2016), ss. 59-70.

96. Shi, Chuan, Zhang, Zhiqiang, Luo, Ping, Yu, Philip S., Yue, Yading, Wu, Bin (2015), 'Semantic path based personalized recommendation on weighted heterogeneous information networks', *International Conference on Information and Knowledge Management, Proceedings*: 19-23-Oct- ss. 453-462, online: http://dx.doi.org/10.1145/2806416.2806528. [dostęp: 24.03.2022].

97. Singh, Kapil Kumar, Aggrawal, Mayank, Pathak, Mohit, Dubey, Pranav (2022), 'Recommender System for an E- commerce Web application', *International Journal for Research in Applied Science & Engineering Technology (IJRASET) Cite*: 10 (V),.

98. Siyan, Karanjit S., Parker, Tim (2002), 'TCP/IP. Księga eksperta.', .

99. Sriram, Stuti Mehra (2018), '3 Tips To Have Great Ecommerce Product Attributes', w: *Vue.ai*, online: https://vue.ai/blog/ai-in-retail/ecommerce-product-attributes/ [dostęp: 30.03.2022].

100. Steck, Harald (2013), 'Evaluation of recommendations: Rating-Prediction and Ranking', *RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems*: ss. 213-220, online: http://dx.doi.org/10.1145/2507157.2507160. [dostęp: 24.03.2022].

101. Stolecka-Makowska, Agata (2016), 'Zakupy Konsumentów Przez Internet w Polsce i Unii Europejskiej – analiza porównawcza', *Zeszyty Naukowe Uniwersytetu Ekonomicznego w Katowicach*:(302), ss. 162-174.

102. Su, Xiaoyuan, Khoshgoftaar, Taghi M. (2009), 'A Survey of Collaborative Filtering Techniques', *Advances in Artificial Intelligence*: 2009 (Section 3), ss. 1-19.

103. Tahyudin, Imam, Haviluddin, Haviluddin, Nanbo, Hidetaka (2019), 'Time complexity of a priori and evolutionary algorithm for numerical association rule mining optimization', *International Journal of Scientific and Technology Research*: 8 (11), ss. 483-485.

104. Vaishnavi, S., Jayanthi, A., Karthik, S. (2013), 'Ranking Technique to Improve Diversity in Recommender Systems', *International Journal of Computer Applications*: 68 (2), ss. 20-24.

105. Wang, Xiao, Ji, Houye, Cui, Peng, Yu, P., Shi, Chuan, Wang, Bai, Ye, Yanfang (2019), 'Heterogeneous graph attention network', *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*:(May), ss. 2022-2032.

106. Wang, Yinggui, Wang, Ben, Huang, Yuxin (2020), 'Comprehensive analysis and mining big data on smart E-commerce user behavior', *Journal of Physics: Conference Series*, .

107. Wojciechowski, Jacek, Pieńkosz, Krzysztof (2013), 'Grafy i sieci', .

108. Wójcik, Jacek (2018), 'Prywatność jako przedmiot wymiany', *Roczniki Kolegium Analiz Ekonomicznych / Szkoła Główna Handlowa*: 49 Społecz ss. 125--135.

109. Wroblewski, Piotr (2015), 'Algorytmy struktury danych i techniki programowania', .

110. Yu, Yonghong, Wang, Can, Gao, Yang (2014), 'Attributes Coupling based Item Enhanced Matrix Factorization Technique for Recommender Systems', *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*: ss. 1-15.

111. Zhang, Xi Zheng (2007), 'Building personalized recommendation system in E-Commerce using association rule-based mining and classification', *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, ICMLC 2007*: 7 ss. 4113-4118.

112. Ziegler, Cai-Nicolas, McNee, Sean M., Konstan, Joseph A., Lausen, Georg (2005), 'Improving recommendation lists through topic diversification', *Conference: the 14th international conference*:(June), ss. 22