МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
«____»_____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

ДОСЛІДЖЕННЯ І РОЗРОБКА СЕГМЕНТІВ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ НА БАЗІ ВЕБ-АНАЛІТИКИ

Освітньо-кваліфікаційний рівень “Магістр”
Спеціальність 122 – “Комп'ютерні науки”

Науковий керівник роботи:	_____ (підпис)	І. С. Скарга-Бандурова (ініціали, прізвище)
Консультант з охорони праці:	_____ (підпис)	Я. О. Критська (ініціали, прізвище)
Студент:	_____ (підпис)	А.Е. Канакін (ініціали, прізвище)
Група:		КН-17Дм

Сєвєродонецьк 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень магістр
Напрям підготовки
(шифр і назва)
Спеціальність 122 – «Комп'ютерні науки»
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри
І.С. Скарга-Бандурова
«_____» _____20____р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Канакіну Артуру Едуардовичу
(прізвище, ім'я, по батькові)

1. Тема роботи **ДОСЛІДЖЕННЯ І РОЗРОБКА СЕГМЕНТІВ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ НА БАЗІ ВЕБ-АНАЛІТИКИ**

керівник проекту (роботи) Скарга-Бандурова І.С., д.т.н., професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від " 18 " 10 2018 р. № ____
2. Термін подання студентом роботи
3. Вихідні дані до роботи Матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз методів рекомендаційної системи з використанням веб-аналітики
Розробка рекомендаційної системи для інтрнет магазину
Розробка методу системи з рекомендацій суспільно с веб-аналітикою для залучення користувачів до інтернет проекту з продажу іжі
Проведення експериментів
Питання охорони праці, екології.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Критська Яна Олександрівна		

7. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналітичний огляд літератури за темою роботи	1.09.18 – 1.10.18	
2	Аналіз методів і моделей профілювання поведінки водія	1.09.18 - 2.10.18	
3	Аналіз методів діагностики стану транспортного засобу	3.10.18 – 9.10.18	
4	Вирішення проблем з ефективною передачею та збереженням отриманих даних	10.10.18 – 24.10.18	
5	Проведення необхідних експериментів	25.10.18 – 25.11.18	
6	Розгляд питань охорони праці та основних напрямків їх дотримання	13.11.18 – 15.12.18	
7	Оформлення пояснювальної записки	22.12.18 – 28.12.18	
8	Оформлення презентації роботи	29.12.18 – 7.01.19	

Студент _____ Канакін А. Е.
(підпис) (прізвище та ініціали)

Керівник _____ Скарга-Бандурова І. С.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Канакін А. Е. Дослідження і розробка сегментів рекомендаційної системи на базі веб-аналітики.

У рекомендаційних системах використовується явний або неявний збір даних. При явному зборі від користувача потрібно заповнювати опитувальні анкети для виявлення переваг, а при неявному зборі для виявлення переваг користувача і складання рейтингів відбувається автоматичне протоколювання його дій. У цій роботі були представлені методи неявного збору даних для рекомендаційної системи. Були розглянуті усі недоліки методів, та їх переваги. Покращення ефективності результатів роботи алгоритмів, для кінцевого користувача, завдяки даним з веб-аналітики.

Ключові слова: веб-аналітика, рекомендаційна система, інтернет магазин, аналіз.

АННОТАЦИЯ

Канакин А. Э. Исследование и разработка сегментов рекомендательной системы на базе веб-аналитики.

В рекомендательных системах используется явный или неявный сбор данных. При явном сборе от пользователя требуется заполнять опросные анкеты для выявления преимуществ, а при неявном сборе для выявления предпочтений пользователя и составление рейтингов происходит автоматическое протоколирование его действий. В этой работе были представлены методы неявного сбора данных для рекомендательной системы. Были рассмотрены все недостатки методов, и их преимущества. Повышение эффективности результатов работы алгоритмов, для конечного пользователя, благодаря данным с веб-аналитики.

Ключевые слова: веб-аналитика, рекомендательная система, интернет магазин, анализ.

ABSTRACT

Kanakin A. E. Research and development of web-based advisory system segments.

Recommendation systems use explicit or implicit data collection. When explicitly collecting from the user, it is necessary to fill out questionnaires for the identification of benefits, and when implicitly collecting for the identification of user benefits and rating is an automatic logging of its actions. Methods of implicit data collection for the advisory system were presented in this paper. Were considered all the disadvantages of the methods, but their advantages. Improving the performance of algorithms, for the end user, thanks to data from web analytics.

Key words: web analytics, advisory system, online store, analysis.

ЗМІСТ

ВСТУП

Актуальність досліджень. Протягом значного відрізку часу в цілому всесвіті стрімкими темпами збільшується кількість інформації. Люди кожної доби сприймають та фільтрують вхідний потік інформації, що підтягується з різних джерел: праця, життєві проблеми, популярні джерела інформації тощо. Після створення мережі Інтернет кількість такої інформації стала стрімко наростати, з'явилася велика кількість послуг для надання користувачам всього необхідного для зручного життя.

За останню десятиденку нажили значної популярності інтернет-сервіси, що пропонують товари всіх припустимих видів (інтернет-магазини), інформацію на будь-який смак (інтернет-журнали, новини, книги, статті) тощо. Користувачу стало дуже важко розбиратися в каталогах товарів та списках статей, навіть із вбудованим пошуком та фільтрацією, бо дуже важко зробити вибір при настільки широкому об'ємі інформації.

Рекомендаційні системи з'явилися на сьогоднішньому ринку IT як механізм для заміни статичному переліку рекомендацій при пошуку або покупках на веб-сайтах. Ці системи утворюють рейтинговий перелік об'єктів на основі різних критеріїв: популярність, релевантність, історія оцінок тощо.

Актуальність теми полягає в тому що, за підтримкою прогнозування рекомендацій вони мають на меті збільшити користувачів до конкретного сервісу. Також, при розробленні рекомендаційної системи з релевантними рекомендаціями, що одержали довіру користувачів, можна розташовувати серед цих рекомендацій інші товари, що популяризуються.

Метою даної кваліфікаційної роботи є вивчення методів та алгоритмів рекомендаційної системи та впровадження їх в існуючий інтернет проект, який призначений для продажу та доставки їжі.

Щоб досягти поставленої мети необхідно вирішити перелік наступних *завдань:*

- Зробити аналіз усіх існуючих алгоритмів рекомендаційної системи.
- Провести серію експериментів, в ході яких виявити найкращий варіант системи для інтрнет магазину, а також проаналізувати вхідні дані які маємо для правильної роботи алгоритмів.
- Обробити отримані результати, підвести підсумки, розкрити найбільш ефективні алгоритми та виявити найменш ефективні.
- Виконати необхідні введення в існуючу систему на основі зроблених досліджень в цій роботі.

Об'єкт досліджень: процес підбору рекомендаційних характеристик для інтернет магазину.

Предметом дослідження є алгоритми та рекомендації щодо покращення результатів роботи, а також модифікації запитів.

Методи дослідження: в рамках дослідження використовувалися методи колаборативної фільтрація, content based тощо.

Практична значимість, або результатом виконання дослідження є:

– рекомендаційна система у парі за даними з веб аналітики;
– методика оцінки ефективності роботи використаних алгоритмів;

Публікації. За темою роботи з викладенням її основних результатів опублікована 1 стаття в науковому фаховому виданні України.

Структура та обсяг магістерської роботи. Магістерська робота містить анотацію, вступ, 4 розділи, перелік використаної літератури, додаток. Пояснювальна записка містить 70 сторінок, 11 таблиць та 18 рисунків.

РОЗДІЛ 1
АНАЛІЗ СУЧАСНИХ АЛГОРИТМІВ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

1.1 Аналіз вимог

Об'єктом дослідження є рекомендаційні системи на базі веб-аналітики.

Мета і завдання дослідження: Практична реалізація методів веб-аналітики, областю застосування веб-аналітики є розширення функціональності сайтів, вимір для користувача активності і оцінка ефективності рекламних інтернет-ресурсів.

Включає наступні методи: аналіз відвідуваності сайту, аналіз поведінки відвідувачів на сторінці і бенчмаркінг.

Обробляти отриманні данні, та формування рекомендаційних систем, їх розробку і розвиток впливали кілька різних областей знань.

Вивчення існуючих методів для розробки рекомендаційних систем та побудова єдиної рекомендаційної моделі

Проаналізувати отримані результати, підбити підсумки, скласти упорядкований перелік рекомендацій по оптимізації, виявити найбільш ефективні в своїй галузі застосування, виявити найменш ефективні.

Провести необхідні впровадження в існуючу систему на основі проведених досліджень.

1.2 Аналіз програмних та інструментальних засобів

1.2.1 Історія зародження рекомендаційних систем

Рекомендаційні системи з’явилися в інтернеті досить давно, близько 20 років тому. Однак справжній підйом в цій області трапився приблизно 5-10 років тому, коли відбулося змагання Netflix Prize. Компанія Netflix - постачальник фільмів і серіалів. Вона тоді давала в прокат НЕ цифрові копії фільмів, а розсилала VHS-касети і DVD. Для них було дуже важливо підвищити якість рекомендацій. Чим краще Netflix рекомендує своїм користувачам фільми, тим більше вони беруть потім в прокат. Відповідно, зростає і прибуток компанії. У 2006 році вони запустили змагання Netflix Prize. Вони виклали у відкритий доступ зібрані дані: близько 100 мільйонів оцінок за п’ятибальною шкалою з зазначенням ID проставити їх користувачів. Учасники змагання повинні були якомога краще передбачати, яку оцінку поставить певного фільму той чи інший користувач. якість передбачення вимірювалося за допомогою метрики RMSE (середньоквадратичне

відхилення). У Netflix вже був алгоритм, який передбачав оцінок з якістю 0.9514 за метрикою RMSE. Завдання було поліпшити прогноз хоча б на 10% - до 0.8563. Переможцю був обіцяний приз в $ 1 000 000.

Змагання тривало приблизно три роки. За перший рік якість поліпшили на 7%, далі все трохи сповільнилося. Але в кінці дві команди з різницею в 20 хвилин надіслали свої рішення, кожне з яких проходило поріг в 10%, якість у них була однакова з точністю до четвертого знака Італійка, яка запізнилася команда (як і багато інших, які брали участь в конкурсі) залишилися ні з чим, однак сам конкурс дуже сильно прискорив розвиток в цій галузі. [3]

Першим рекомендаційним сервісом став Імхонета, заснований в 2007 році. Сервіс одночасно працював з різними видами контенту. Творці назвали його мультикультурним, тому що він допомагав користувачу зробити вибір в різних культурних і навколо культурних областях, рекомендуючи книги, фільми, ігри та інше. У нульових рекомендаційні сервіси почали проникати в область електронної комерції. На сьогоднішній день кращі рекомендаційні системи реалізовані у декількох компаній:

а) Amazon.com - інтернет-магазин, покупці якого можуть

б) залишати відгуки про товари і оцінювати їх, що, в свою чергу, в Надалі допомагає вибрати потрібну річ гарної якості;

в) Rich Relevance - рекомендаційна платформа для мультиканальною персоналізації інтернет-магазинів, створена розробниками Amazon.com;

г) Retail Rocket - російська рекомендаційна платформа, аналог Rich Relevance

На малюнку 1.1 наведено приклад того, як виглядають рекомендації на відомому онлайн-гіпермаркеті Amazon

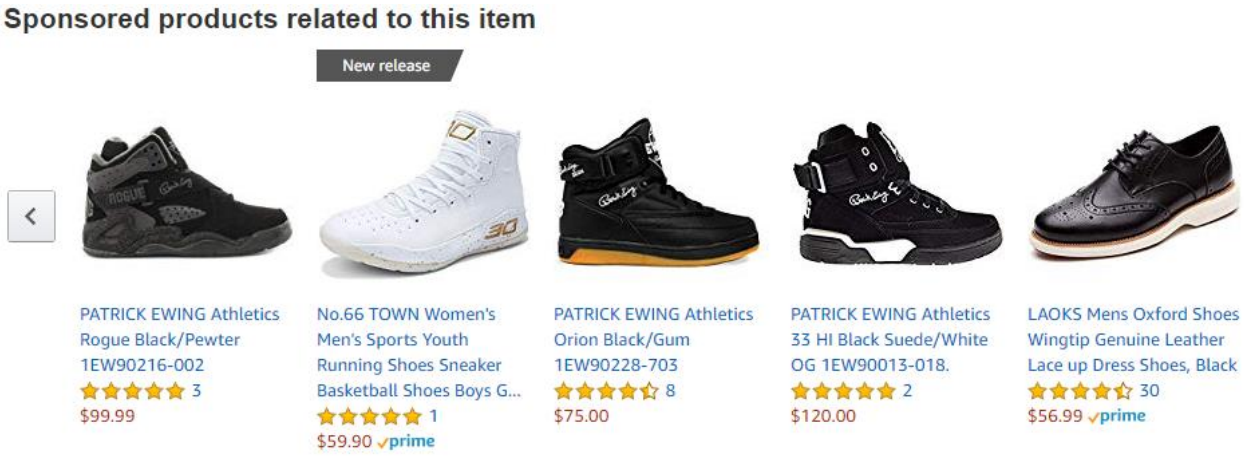


Рис. 1.1 Приклад рекомендацій на Amazon

В даному випадку такий спосіб рекомендації забезпечує зручність навігації користувача по веб-ресурсу. Якщо електронний магазин містить більше, ніж 5 000 найменувань різної продукції, то зорієнтуватися стає досить важко, а що робити, якщо товарів більше декількох десятків тисяч? На допомогу приходить міні-програму пошуку товарів, які з якихось основними параметрами відповідають шуканого (т. е. належать до однієї групі продуктів) або які комплементарні по відношенню до шуканої речі (наприклад, миючий засіб для посудомийки на додаток до самого агрегату). Однозначно, це підвищує конверсію сайту. (Конверсія в інтернет-маркетингу - це відношення числа відвідувачів сайту, виконали на ньому будь-які цільові дії (приховані або прямі вказівки рекламодавців, продавців, творців контенту - покупку, реєстрацію, підписку, відвідування певної сторінки сайту, перехід по рекламному посиланню), до загальної кількості відвідувачів сайту, виражене в відсотках).

1.2.2 Розвиток рекомендаційних систем

У більшості випадків розвиток рекомендаційних систем полягає в поліпшенні алгоритмів рекомендації. Мета цього прогресу - давати відвідувачам сайтів найбільш точні рекомендації, що задовольняють їх запити.

Для досягнення цього математичні алгоритми, що лежать в основі рекомендаційних сервісів, повинні постійно навчатися. Тут вступає в силу машинне навчання та інтелектуальний аналіз даних Data mining. Схематично це можна описати так: сервіс веб-сайту надає користувачеві набір рекомендацій, далі розробник отримує від нього зворотний зв'язок, аналізує її на предмет відповідності даних раніше рекомендацій інтересам відвідувача, перенавчати математичну модель, потім знову пропонує рекомендації і так далі по колу.

Поліпшення математичних алгоритмів рекомендаційних систем розвивається по наступних аспектах:

а) кількість рекомендованого контенту - його повинно бути достатня кількість, щоб було що і з чого радити;
б) кількість інформації про рекомендований контенті (наприклад, хто автор книги, хто перекладач, хто ілюстратор, скільки символів в ній) - чим її більше, тим точніше ми зможемо вибрати потрібний;
в) обсяг інформації про користувачів (стать, ім'я, вік, країна проживання) - знову ж таки, чим її більше, тим конкретніше будуть підібрані рекомендації;

г) зручність для користувача частини програми, тобто інтерфейсу - чим комфортніше для відвідувача, тим більше інформації ми зможемо отримати про його реакції на наші поради.

1.2.3 Огляд методів веб-аналітики

Аналіз відвідуваності є досить поширеним методом веб-аналізу, тому що дозволяє проаналізувати статистичні дані, абсолютні та відносні показники відвідуваності сайту за допомогою спеціальних програм - лічильників.

Аналіз usability (юзабіліті) спрямований на дослідження конверсійних шляхів відвідувача по сайту. Сам термін «юзабіліті» слід розглядати як синонім слова «ергономічність» з тією лише різницею, що юзабіліті визначає не матеріальні якості продуктів і послуг, а нематеріальні (ступінь зручності користування і зрозумілості клієнту). При грамотному використанні юзабіліті дозволяє не тільки покращенню просування сайту, але і підвищити його конверсію в 10 і більше разів.

Оцінка і аналіз поведінки відвідувачів на сторінці є одним з головних пріоритетів веб-аналізу, оскільки за допомогою сайту з клієнтом відбувається контакт, що істотно ускладнює взаємодію зі споживачем і виявлення його потреб. Проблема в даному випадку вирішується за рахунок використання інструментарію веб-аналітики. Сучасний інструментарій веб-аналізу має широкі можливості для вивчення поведінки споживачів їй, даючи максимально можливу деталізацію активності споживача з можливістю візуалізації всіх дій відвідувачів сайту (рухів миші, кліків, натискань клавіш, скролінгу і т.д.). Аналіз поведінкової інформації представляється у вигляді карт активності відвідувачів на кожній сторінці сайту.

Важливим в умовах жорстокої конкуренції є порівняльний аналіз сайтів із загальносвітовими галузевими тенденціями і конкурентами на світових і регіональних ринках послуг (бенчмаркінг). Бенчмаркінг представлений на ринку веб-аналітики такими компаніями, як Alexa, GemiusAudience і Google Trends.

Практична реалізація методів веб-аналітики проводиться за допомогою інструментарію веб-аналізу. Збір і аналіз статистичної інформації про сайт здійснюється за допомогою установки лічильників і лог-аналізаторів. Інструмент лічильник - це зовнішні програми, завдання яких - збір і обробка статистики про відвідування користувачами кожної сторінки сайту. До переваг лічильників слід віднести простоту і зручність у використанні, і отримання оперативної та наочної інформації. Однак існують недоліки, які знижують ефективність лічильників: необхідність установки на сайт

стороннього програмного коду, високий ступінь ризику втрати даних при неповно й завантаженні сторінки або технічних збоях, неможливість збору статистичної інформації про завантажений контент, трафік сайту і закладок для користувача інтерфейсів. Найбільш поширеними лічильниками-рейтингами є Liveinternet, OpenStat, HotLog і ін.

Лог-аналізатор являє собою більш досконалий інструмент веб-аналізу, який має розширений функціонал. Лог-аналізатор дозволяє відстежувати і аналізувати помилки роботи сервера, хакерські атаки, створювати специфічні звіти, а також формувати більш точні дані про відвідування, завантажування контенту і закладок. Однак такі функціональні можливості вимагають високої кваліфікації адміністратора сайту. До основних лог-аналізаторів слід віднести WebTrends, Webalizer, AWStats.

Найбільш розповсюдженою системою пошуку в мережі інтернет є Google. Google Analytics (GA) являє собою безкоштовний інтернет-сервіс для створення детальної статистики відвідувачів веб-сайтів. Статистика відвідувачів представляється можливим за рахунок JS-код на сторінках свого сайту. За умови дозволеного виконання Javascript, код відстеження (JS-код) спрацьовується при кожному відкритті сторінки користувачем.

Однак Google Analytics - це не просто лічильник відвідувань, це повноцінний інструмент аналізу ефективності роботи сайту фірми і проведених маркетингових заходів. Google Analytics дозволяє не тільки відслідковувати джерела відвідувачів сайту, але і аналізувати їх ефективність. У GA створена система вибору готових статистичних звітів, які базуються на наступних функціональних можливостях:

- відстеження цілей;
- інтеграція Google Analytics з Google AdWords і Google AdSense;
- відстеження продажів інтернет-магазинів;
- відстеження мобільних пристроїв;
- відстеження внутрішнього пошуку по сайту;
- порівняння показників;
- відстеження використання Flash, Ajax і відео;
- розширена сегментація в Google Analytics;
- призначені для користувача звіти;
- експорт даних у формати Excel, CSV, PDF;
- відправка звітів по електронній пошті;
- API для розробників.

Комбінування різних функціональних можливостей дозволяє формувати звіти, найбільш повно відповідають запитам. Вивчення споживчого поведінки на сайті, юзабіліті і аналіз відвідуваності сайту будується на звітах, сформованих за певними критеріями.

Сервіс інтегрований з Google AdWords. Особливістю сервісу є те, що веб-майстер може оптимізувати рекламні та маркетингові кампанії Google AdWords за допомогою аналізу даних, отриманих за допомогою сервісу Google Analytics, про шляхи приходу відвідувачів, про час перебування на сайті і про географічне місце розташування відвідувачів. Користувачеві представляються групи оголошень і віддача від ключових слів в звітах. Також доступні додаткові можливості, включаючи поділ відвідувачів на групи.

Користувачі сервісу можуть визначити цілі і послідовності переходів. Метою може виступати сторінка завершення продажів, показ певних сторінок або загрузка файлів. Використовуючи цей інструмент, маркетологи можуть визначати, яка з рекламних кампаній є успішною, і знаходити нові джерела цільової аудиторії.

Системо утворюючим ядром сервісу Google Analytics є аналітичні інструменти.

За допомогою аналітичних інструментів адміністратор сайту може виконувати детальний аналіз різних даних, що відображають взаємодію відвідувачів зі сторінками сайту.

Безумовною перевагою сервісу GA є звітність в режимі реального часу. Такий вид звітності призначений для миттєвої оцінки результатів (оперативна оцінка зацікавленості відвідувачів до нових матеріалів сайту, оцінка ефекту записів в соціальних мережах і ін.). Однак стандартизовані звіти не завжди відображають необхідні дані для детального аналізу поведінки споживача. Тоді користувачам сервісу представляється можливість сформувати настроюється звіт з індивідуальним набором показників і параметрів, який більш точно буде підходити вимогам користувачів.

Широкі можливості Google Analytics дозволяють не тільки використовувати стандартні параметри і показники, а й створювати власні для збору тих даних, які не відслідковуються автоматично. Введення користувальницьких змінних дозволяє зробити аналіз більш гнучким у вивченні взаємодії відвідувачів з вмістом сайту.

Значно полегшують веб-аналіз зведення, що представляють собою набір віджетів. За допомогою зведень можна відстежувати одночасно кілька показників, швидко сприймати ефективність акаунтів і зіставляти дані з різних звітів. Відстежити відповідають за візуалізацію показників у зведенні і оперативне відображення інформації. Безумовним плюсом зведень є моніторинг та загальний доступ до ключовими показниками ефективності, а також SEO для пошукової оптимізації.

Одним з ключових аналітичних інструментів Google Analytics є візуалізація даних. Звіт по візуалізації являє собою графічне представлення показників обраних користувачем і дозволяє порівнювати обсяги трафіку з різних джерел, вивчати структуру

трафіку і вимірювати ефективність сайту. Для ще більшої персоналізації відображення даних буде використано стандартні конфігурації API. [2]

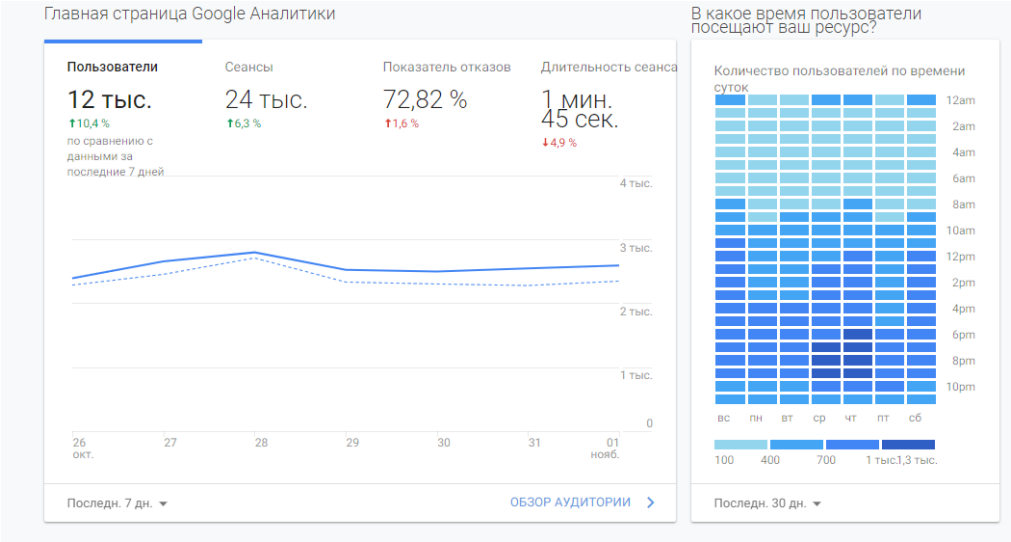


Рис. 1.2 Візуалізація даних

Однак функціональні можливості Google Analytics представлені не тільки аналітичними інструментами. Сучасний сервіс дозволяє проводити мобільну аналітику, аналіз контенту сайту, аналіз конверсії і соціальних функцій. Аналіз контенту дає працівникам уявлення про те, які сторінки їхнього сайту найбільш популярні у відвідувачів (відвідування сторінок), скільки вони проводять часу на них (тривалість відвідувань) і скільки конверсій приносить сторінка (перегляди).

Розвиток електронної комерції зумовив необхідність вивчення ефективності віддачі сайту. Аналіз конверсії дозволяє з'ясувати, які канали цифрового маркетингу приваблюють відвідувачів на сайт, а також виміряти кількість продажів, завантажень і т.д. Аналіз конверсію покликаний адаптувати і оптимізувати сайт і маркетингові програми до бізнес-цілям туристичного підприємства. Основним інструментом представлення даних є воронка продажів, що відображає послідовність переходів до мети (від холодного контакту до завершення покупці).

Для ефективного використання інструменту аналіз конверсії необхідно враховувати, що сайт містить динамічні матеріали, тому цілі конверсій повинні також бути динамічними. Тобто зміна контенту на сайті, наприклад сторінки спец пропозицій SPO, повинна мати на увазі зміни цілей конверсії.

Розрахувати ефективність і проаналізувати результати продажів через мережу інтернет представляється можливим за допомогою звітів про електронну торгівлю. Цей звіт дає чітке уявлення про найбільш проданих і популярних пропозиціях продуктів та

послуг. Також крамниця може відстежувати транзакції по компаніям і оптимізувати кошик з метою підвищення продажів і лояльності клієнтів.

Для більш чіткого розуміння поведінки відвідувачів на сайті та їх дій з пошуку сайту використовують можливості багатоканальних послідовностей і стеження за переміщеннями відвідувачів. Багатоканальні послідовності відображають вплив усіх факторів цифрового маркетингу (пошукові системи, медійна реклама, соціальні мережі, директ-мейл, партнерські канали та ін).

Візуалізація переходів і багатоканальних цільових послідовностей дає можливість відстежувати шляхи конверсії і визначати, що з контенту подобається відвідувачам, а що ні. Крім того, цей сервіс дозволяє визначити сильні та слабкі сторони системи навігації по сайту і оптимізувати її. [1]

1.2.4 Приклади додатків по рекомендаційним системам

Напівавтоматична розмітка контенту. Веб-сервіс Delicious надає користувачеві можливість створювати власну колекцію закладок (посилань на інтернет ресурси), а також розмічати закладки семантичними тегами.

Сервіс накопичує інформацію в взаємодіях виду (користувач, закладка) і (користувач, закладка, тег). Крім зазначеної вище завдання рекомендації закладок, має місце завдання передбачення релевантності тега в контексті закладки, створеної певним користувачем.

Оцінка знань по невеликій кількості питань. У конкурсі з аналізу даних «What Do You Know?» На платформі Kaggle, учасникам пропонувалося передбачити коректність відповіді студента на запитання під час проходження тестування.

У цьому завданню є три чинники, кожен описується набором ознак (в тому числі категоріальних):

– користувачів (відомий тільки ідентифікатор)
– питання (номер питання, тип питання, ідентифікатор тесту, такі ключові слова)
– контекст (дата і час відповіді, швидкість відправки відповіді).

Навчальна вибірка містить четвірки (користувач, питання, контекст, коректність відповіді). [7]

Використовуючи модель, пророкує коректність відповідей на питання по відома набору трійок (користувач, питання, контекст), користувачеві (студенту) можна задати невелике число тестових питань, пості отримання відповідей на які, передбачити

коректність відповідей на всі питання з тесту (або з декількох різних тестів), таким чином оцінивши рівень знань студента і виявивши прогалини в навчанні.

Слід зазначити, що краще рішення конкурсу використовував факторизаціїні машини.

1.3 Аналіз математичних моделей і методів для вирішення задачі

Можна виділити три основні підходи до побудови рекомендаційних систем:

а) на підставі описів (content-based);

б) колаборативна фільтрація (collaborative filtering);

в) гібридний підхід.

Content-based. Підхід на підставі описів передбачає, що про користувачів і про рекомендовані об'єкти відомо досить багато інформації. Наприклад, всі користувачі заповнюють анкету, в якій вказують свою соціально-демографічні інформацію, інтереси, і т.д. Про товари з інтернет-магазину може бути відомо їх опис, призначення, цінова категорія, бренд, і інші характеристики. За історії взаємодії користувачів і об'єктів на сервісі можна побудувати навчальну вибірку і звести пророкування хороше ставлення до добре вивченою задачі навчання по прецедентах.

На практиці, використання такого походу сильно обмежена, тому що збір описової інформації про користувачів і об'єктах дуже дорога процедура, яку часто неможливо організувати не на шкоду якості використання сервісу, що робить рекомендаційну систему невиправдано дорогий.

Колаборативна фільтрація. Колаборативною фільтрацією називається пророкування ступеня переваги в умовах, коли рекомендаційна система не володіє якою-небудь описової інформацією про користувачів і об'єктах (або не використовує), будує прогноз виключно на підставі взаємодії користувачів з об’єктів. [23]

Гібридний підхід. Незважаючи на те, що алгоритми колаборативної фільтрації на практиці показують високі показники ефективності, облік додаткової інформації може стежити показники ще вищі. Одним з недоліків колаборативної фільтрації в порівнянні з методами, заснованими на просторі ознак описі, є проблема холодного старту

Гібридний підхід використовує композиції алгоритмів заснованих на ознакових описах і результатів колаборативної фільтрації.

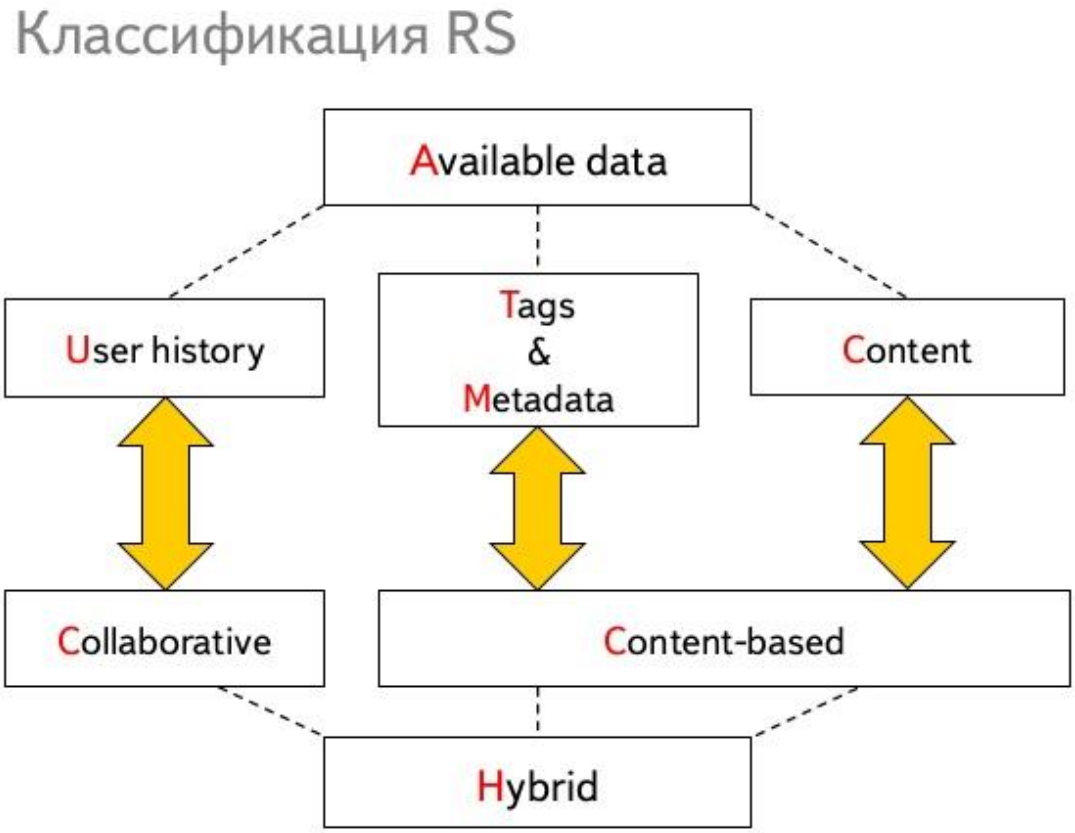


Рис. 1.3 Класифікація рекомендаційних систем

1.3.1 Завдання ознакової фільтрації

Колаборативною фільтрацією називається пророкування ступеня переваги в умовах, коли рекомендаційна система не володіє якою-небудь описової інформацією про користувачів і об'єктах (або не використовує), будує прогноз виключно на підставі взаємодії користувачів з об'єктів.

Нехай U - безліч користувачів (users);

I - безліч об'єктів (items), інформація про відомих перевагах представлена у вигляді набору трійок:

$$D = \{(u,i,rui)\}(u,i)\in R, \quad (1.1)$$

де $r_{ui} \in R$ - речова ступінь переваги об'єкта $i \in I$ користувачем $u \in U$; $R \subseteq U \times I$ - безліч пар (користувач, об'єкт), про які відома ступінь переваги.

Для подальшої зручності, введемо також позначення: R *(u)* = *{i: (u, i)* ∈ R} - безліч об'єктів, суміжних з користувачем *u*, аналогічно: R *(i)* = *{u: (u, i)* ∈ R }.

За відомою інформацією D потрібно вміти будувати пророкування переваги $r_{ui} \approx r_{ui}$ для нових пар *(u, i)* 6∈ R.

Будемо називати матрицею оцінок матрицю

$$R \in (\mathrm{R} \cup \emptyset)^{|U| \times |I|} \tag{1.2}$$

Рядки якої відповідають користувачам, стовпці - об'єктам, а елементи приймають значеня r_{ui}, якщо *(u, i)* ∈ R, інакше - пропуск ∅. На завдання колаборативної фільтрації можна дивитися як на задачу заповнення пропущених значень в матриці.

Крім передбачення значення переваги, на практиці можуть бути цікаві такі завдання:

– побудова списку рекомендацій з об'єктів, на які не відома ступінь переваги (нові для користувача):

$$\text{Recommended}_K(u) = \text{Top}_K \max_i r_{ui} \; 7 \rightarrow \{(i_1, r_{ui\,1}), (i_2, r_{ui\,2}), \ldots, (i_K, r_{ui\,K})\} \tag{1.3}$$

– визначення ступінь схожості об'єктів і побудова списків найбільш схожих:

$$\text{Similar}_K(i) \; 7 \rightarrow \{(i_1, s_{ii\,1}), (i_2, s_{ii\,2}), \ldots, (i_K, s_{ii\,K})\}, \tag{1.4}$$

де s_{ij} - ступінь схожості між двома об'єктами;

– обґрунтування списку рекомендацій: деякий людино-зрозуміле пояснення, чому користувачеві *u* був порекомендувати об'єкт *i.*

Підходи до вирішення завдання колаборативної фільтрації умовно можна розділити на дві великі групи :

а) засновані на евристиках (memory / heuristic-based),

б) засновані на побудові моделі переваги (model-based).

Memory-based. До першої групи методів (memory-based) відносяться алгоритми, що виражають припущення значення безпосередньо через елементи матриці оцінок.

Яскравим прикладом memory-based алгоритму колаборативної фільтрації є зважування переваги по користувачах (user-based) і по об'єктах (item-based).

$$\bar{r}_u = \frac{1}{|\mathcal{R}(u)|}\sum_{i\in\mathcal{R}(u)} r_{ui},\ \bar{r}_i = \frac{1}{|\mathcal{R}(i)|}\sum_{u\in\mathcal{R}(i)} r_{ui} \tag{1.5}$$

середні значення переваг по користувачам і об'єктам, sim *(u, u* $^{0)}$, sim *(i, i* $^{0)}$ - наперед задані метрики схожості користувачів і об'єктів.

Міра схожості sim *(u, u* $^{0)}$ (і аналогічна для об'єктів) обчислюється по матриці оцінок *R*, або з використанням додаткової інформації про користувачів (об'єктах). Міра схожості є важливим параметром алгоритму. Найбільш загальновживані прості метрики схожості - кореляція Пірсона і косинусна відстань відповідних рядків (стовпців) матриці оцінок:

$$\mathrm{sim}(u,u') = \frac{\sum_{i\in\mathcal{R}(u)\cap\mathcal{R}(u')}(r_{u,i}-\bar{r}_u)(r_{u',i}-\bar{r}_{u'})}{\sqrt{\sum_{i\in\mathcal{R}(u)\cap\mathcal{R}(u')}(r_{u,i}-\bar{r}_u)^2\sum_{i\in\mathcal{R}(u)\cap\mathcal{R}(u')}(r_{u',i}-\bar{r}_{u'})^2}} \tag{1.6}$$

Найбільш використовувані і реалізовані у вигляді бібліотек з відкритим вихідним кодом memory-based алгоритми.

Подібні алгоритми гарні для одноразового обчислення рекомендацій на розподіленому кластері, добре лягають на обчислювальну архітектуру MapReduce, проте погано підходять для оперативного оновлення рекомендацій. Налаштування заходи схожості для завдання з конкретної предметної області є скоріше мистецтвом, ніж налагодженою технологією. [31]

Одна з найбільш серйозних проблем memory-based методів - неадекватність передбачень в умовах сильної розрідженості матриці оцінок *R* (в сенсі пропущених значень), що призводить до неможливості підрахунку метрик схожості (в разі, якщо безліч R *(u)* ∩ R *(u* $^{0)}$- порожньо). Сильна розрідженість матриці оцінок може бути наслідком проблеми холодного старту, проте в деяких областях вона сильно розріджена завжди і не може стати досить щільної (наприклад, в разі інтернет-магазинів, користувач залишає інформацію про переваги в середньому 2-5 об'єктів).

Model-based. Алгоритми з другої групи (model-based) прагнуть вибрати функцію r (u, i; θ) з деякого сімейства моделей, параметризрвані θ ∈ Θ. Вибір відбувається, наприклад, шляхом мінімізації емпіричного ризику:

$$\mathcal{L}(\theta) = \sum_{(u,i)\in\mathcal{R}} l(\hat{r}(u,i;\theta), r_{ui}) + \lambda\Omega(\theta) \to \min_{\theta\in\Theta}, \tag{1.7}$$

де *l (r,* r) - функція втрат регресії;

λ - сила регуляризації;

Ω *(θ)* - регуляризатора на безлічі параметрів Θ;

Всі перераховані методи мають наступні недоліки:

- Проблема холодного старту.
- Погані прогнози для нових / нетипових користувачів / об'єктів.
- Тривіальність рекомендацій.
- Ресурсомісткість обчислень. Для того, щоб будувати припущення нам потрібно тримати в пам'яті всі оцінки всіх користувачів.

1.3.2 Модель переваги SVD

Одна з найбільш відомих моделей переваги - SVD, визначається наступним чином:

$$\hat{r}(u,i;\theta) = \boldsymbol{p}_u^T \boldsymbol{q}_i, \quad \theta = (\{\boldsymbol{p}_u\}_{u\in U}, \{\boldsymbol{q}_i\}_{i\in I}) \tag{1.8}$$

Де pu ∈ RR - вектор прихованих (латентних) переваг користувача;

$q_i \in R^R$ - аналогічний вектор для об'єкта;

R - розмірність латентних векторів, будемо називати її рангом моделі SVD. Модель SVD навчається шляхом оптимізації квадратичних втрат $l(r,r\hat{}) = (r\hat{}-r)^2$ з квадратичною регуляризації:

$$\Omega(\theta) = \Omega(\theta) = {}^{P}{}_{u\in U} \|p_u\|^2 + {}^{P}{}_{i\in I} \|q_i\|^2. \tag{1.9}$$

Назва моделі SVD не слід плутати з сингулярним розкладанням (Singular Value Decomposition). Таку назву моделі увійшло в обіг в співтоваристві дослідників рекомендаційних систем з моменту публікації Саймоном Фанком статті в блозі. Ідея подібної моделі була швидко підхоплена і поширена дослідниками в різних модифікаціях тому що мала досить хорошу узагальнюючу здатність в порівнянні з memory-based методами, популярними на той момент. Як буде показано далі, дана модель має мало спільного з сингулярним розкладанням. [26]

У своїй статті, Саймон Фанк оптимізував модель SVD методом стохастичного градієнтного спуску. Трохи пізніше був запропонований більш ефективний і допускає розпаралелювання по користувачам і об’єктам алгоритм ALS (Alternating Least Squares). Його ідея полягає в тому, що при фіксованих латентних векторах об’єктів $\{q_i\}i_{\in I}$,

оптимізація (4) тільки по латентним векторах користувачів $\{p_u\}u_{\in U}$ розбивається на незалежні регуляризоване завдання найменших квадратів по користувачам:

$$\mathcal{L}_u(\boldsymbol{p}_u) = \sum_{i \in \mathcal{R}(u)} (\boldsymbol{p}_u^T \boldsymbol{q}_i - r_{ui})^2 + \lambda \|\boldsymbol{p}_u\|^2 \to \min_{\boldsymbol{p}_u}, \quad \forall u \in U. \tag{1.10}$$

Для кожного користувача u ∈ U, оптимальний латентний вектор $p^{?}{}_{u}$ U обчислюється як рішення СЛАР з матрицею розміру R × R:

$$\text{p}^{\star}_{u} = \left(\sum_{i \in \mathcal{R}(u)} \boldsymbol{q}_i \boldsymbol{q}_i^T + \lambda \boldsymbol{I} \right)^{-1} \left(\sum_{i \in \mathcal{R}(u)} r_{ui} \boldsymbol{q}_i \right). \tag{1.11}$$

При фіксованих латентних векторах користувачів, оптимальні латентні вектора об'єктів обчислюються аналогічним чином. Алгоритм ALS полягає в циклічному перерахунку латентних векторів користувачів і об'єктів.

1.3.3 Зворотній зв'язок в рекомендаційних системах

Зворотним зв'язком (feedback) користувача на деякий об'єкт в рекомендаційних системах прийнято називати подія, за яким можна судити про перевагу користувача до об'єкта. Ось кілька прикладів зворотного зв'язку від користувача:

- проставлення оцінки об'єкту за бальною шкалою (кількість зірок);
- натискання на кнопку «подобається» (лайк) / «не подобається» (дізлайк);
- відвідування сторінки з описом об'єкта, перехід за посиланням на об'єкт (клік);
- відвідування сторінки з описом об'єкта більш ніж один раз (зацікавленість); • додавання в корзину / покупка об'єкта в разі, якщо це товар.

Саме по зворотного зв'язку користувача на різні об'єкти, рекомендаційна система формує матрицю оцінок переваг R, до якої потім застосовуються алгоритми колаборативної фільтрації. Перетворення зворотного зв'язку в числове значення переваги - непроста і дуже важливе завдання в налаштуванні рекомендаційних систем. Як правило, при виборі схеми оцінки переваги оптимізується метрика, безпосередньо пов'язана з ключовими показниками ефективності (KPI) бізнесу. Техніки підбору схеми оцінки переваги виходять за рамки даної роботи.

За видами зворотного зв'язку, завдання моделювання переваги в рекомендаційних системах прийнято розділяти на два види:

а) з явною зворотним зв'язком (explicit feedback);

б) з неявній зворотним зв'язком (implicit feedback).

Так, наприклад, рекомендації за оцінками з п'ятибальною шкали - приклад завдання з явною зворотним зв'язком. Рекомендаційні системи, що керуються актами покупок, відвідуванням сторінок - приклади завдань з неявній зворотним зв'язком.

У разі неявній зворотного зв'язку є невизначеність в тому, позитивно або негативно впливають конкретний акт зворотного зв'язку на ступінь переваги. Купівля товару в інтернет-магазині може означати досягнення користувачів своєї споживчої мети (позитивне перевагу), але в той же час покупець міг після отримання товару в ньому розчаруватися і правильно було б зарахувати негативну ступінь переваги. Очевидно, що відвідування сторінок користувачами веб-сервісу можуть відбуватися при абсолютно різного ступеня зацікавленості користувача в контенті.

Варто відзначити досить типову ситуацію, коли рекомендаційної системі подаються на вхід виключно позитивні приклади взаємодії користувачів і об'єктів. Наприклад, веб-сервіс Twitter не має функціональності, що дозволяє користувачеві висловити своє низьке перевагу контенту, а присутній тільки лише спосіб «заохотити» той чи інший контент, поширивши його своїм передплатникам за допомогою функції «репост». Подібна зворотний зв'язок користувача дуже надійно (в порівнянні з іншими) вказує на позитивну ступінь переваги. Надійність «Фортеця» в сервісі Twitter підкріплена відповідальністю користувачів перед своїми передплатниками.

В роботі висувається три типи припущень про матрицю оцінок R, використовуваних в алгоритмах колаборативної фільтрації:

а) всі пропуски в матриці R відбулися випадково (MAR, missing-at-random);

б) всі пропуски є наслідком негативного переваги (AMAN, all-missingare-irrelevant);

в) пропуски в матриці R відбулися не випадково (MNAR, missing-not-at-random).

Під «випадковістю» вище мається на увазі розподіл усіх припущення про усунення розподілу переваги пропущених оцінок в негативну сторону.

Алгоритми колаборативної фільтрації, що використовують гіпотези AMAN і MNAR історично прийнято називати «застосовними до даних з неявній зворотним зв'язком» (For implicit feedback datasets).

Ідея обліку гіпотез AMAN / MNAR породила групу алгоритмів колаборативної фільтрації, які налаштовують латентні фактори користувачів і об'єктів не тільки на відомі

елементи матриці R, а й на припущення [14, 21, 24, 11]. Пропущені елементи, за припущенням, мають негативну оцінку переваги, але при цьому впливають на параметри моделі з меншою вагою, ніж відомі.

1.3.4 Облік контексту в рекомендаційних системах

Часто, для побудови релевантного списку рекомендацій, рекомендаційної системі потрібна інформація про контекст, в якому веб-сервіс запитує рекомендацію для користувача. Ось деякі приклади такої контекстної інформації:

– географічне положення мандрівника (наприклад, в разі рекомендації пам'яток);
– стан кошика відвідувача інтернет-магазину, наявність замовлення в оформленні;
– сезон, час, погода та інші фактори.

У простих випадках (як, наприклад з географічним положенням), досить фільтрувати об'єкти рекомендацій деяким правилом. У більш складних випадках (як, наприклад, погода) важко побудувати однозначно розумні правила фільтрації об'єктів.

В алгоритмах колаборативної фільтрації з використанням контекстної інформації потрібно навчитися передбачати оцінку переваги $r_{uic} \approx r_{uic}$ для користувача $u \in U$, об'єкта $i \in I$ і контексту $c \in C$, де C - заздалегідь заданий набір контекстів.

Одним з основних підходів до моделювання переваги з урахуванням контекстної інформації є факторізаціоної моделі. В роботі пропонується використовувати факторізаціоної машини (будуть описані далі), контекстну інформацію представляти у вигляді додаткових ознак для факторізаціоної машини. В роботі пропонується кодувати контекст в додаткові координати тензора оцінок, здійснювати нізкорангових канонічний розклад тензора оцінок (запропонований в роботі алгоритм iTALS далі буде розглянуто докладно), а в роботі - попарне тензорне розкладання (Pairwise-Interaction Tensor Factorization). Більш детальний огляд алгоритмів колаборативної фільтрації з урахуванням контексту можна знайти в роботі.

1.3.5 Алгоритми кластерізації

Алгоритми кластерізації - це різновид "Спонтанного навчання" (unsupervised learning), що дозволяє виявити структуру в рядах на перший погляд випадкових (або немаркованих) даних. У загальному випадку такий алгоритм базується на виявленні подібності між елементами (наприклад, між читачами блогу) за допомогою обчислення їх

відстані від інших елементів в просторі ознак (feature space) (ознакою в просторі ознак може, наприклад, бути кількість прочитаних статей в наборі блогів). Кількість незалежних ознак визначає розмірність простору ознак. Якщо елементи "близькі" один до одного, то їх можна об'єднати в один кластер. Існує безліч алгоритмів кластеризації. Найпростішим з них є алгоритм k-середніх (k-means), який розділяє елементи на k кластерів. Спочатку елементи розподіляються по цим кластерам в довільному порядку. Потім для кожного кластера обчислюється центр мас (або просто центр) як функція його членів. Після цього перевіряється відстань кожного члена кластера від центру цього кластера. якщо по результатами цієї перевірки член виявляється ближче до іншого кластеру, то він переміщається в цей кластер. Після перевірки всіх відстаней для всіх членів центри кластерів обчислюються заново. При досягненні стабільного стану (в процесі чергової ітерації члени не транспортувалися) набір вважається кластерізованний належним чином, і алгоритм зупиняється. Обчислення відстані між двома об'єктами може бути важким для візуалізації.

Один з поширених методів вирішення цього завдання полягає в тому, щоб розглядати кожен член кластера як багатовимірний вектор і обчислювати для нього т.ч. евклідова відстань.

Існує безліч інших різновидів кластеризації, в тому числі теорія адаптивного резонансу (Adaptive Resonance Theory), нечітка кластеризація методом С-середніх (Fuzzy C-means), імовірнісна кластеризація за допомогою ЕМ-алгоритму (Expectation-Maximization) і т.д.

1.3.6 Байєсова Мережі довіри

Байєсовські мережі довіри - Bayesian Believe Network - використовуються в тих областях, які характеризуються успадкованою невизначеністю. Ця невизначеність може виникати внаслідок:

- неповного розуміння предметної області;
- неповних знань;
- коли завдання характеризується випадковістю.

Таким чином, байєсовські мережі довіри (БСД) застосовують для моделювання ситуацій, що містять невизначеність в деякому сенсі. Для байесовских мереж довіри іноді використовується ще одна назва причинно-наслідковий мережу, в яких випадкові події з'єднані причинно-наслідковими зв'язками.

З'єднання методом причин і наслідків дозволяють більш просто оцінювати ймовірності подій. У реальному світі оцінювання найбільш часто робиться в напрямку від "спостерігача" до "спостереження", або від "ефекту" до "слідству", яке в загальному випадку більш складно оцінити, ніж напрямок "наслідок -> ефект", тобто в напрямку від слідства.

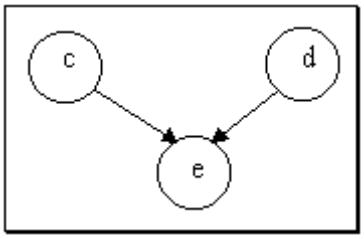


Рис. 1.4 Приклад найпростішої байєсівської мережі довіри.

Розглянемо приклад мережі (рис.1.4), в якій ймовірність перебування вершини "e" в різних станах (ek) залежить від станів (ci, dj) вершин "c" і "d" і, в загальному випадку, визначається виразом:

$$p(e_k) = \sum_i \sum_j p(e_k | c_i, d_j) \times p(c_i, d_j) \tag{1.12}$$

де p (ek | ci, dj) - умовна ймовірність перебування вершини "e" в стані (ek) в залежності від станів (ci, dj), в яких знаходяться вершини "з" і "d". Але так як події, представлені вершинами "c" і "d" є незалежними, тобто між ними відсутній причинно-наслідковий связб, то їх спільна ймовірність може бути представлена у вигляді *p (ci, dj) = p (ci) * p (dj)*

Розглянемо приклад більш складної мережі (рис.1.5). Даний малюнок ілюструє умовну незалежність подій "з" і "d" від впливають на них груп подій.

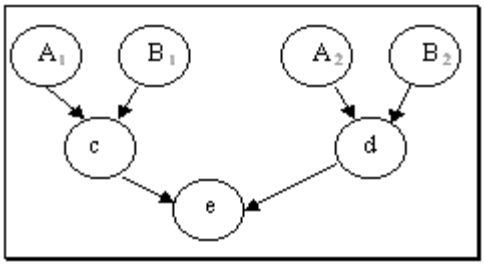


Рис.1.5 Дворівнева БСД.

Для оцінки ймовірності перебування вершини "e" в різних станах слід використовувати наведене вище вираз. Що ж стосується вершин "c" і "d", то для їх оцінки будемо використовувати вирази, аналогічні тому, що і для обчислення p (ek), тоді:

$$p(c_i) = \sum_m \sum_n p(c_i | A_{1m}, B_{1n}) \times p(A_{1m}) \times p(B_{1n}),$$
$$p(d_j) = \sum_m \sum_n p(d_j | A_{2m}, B_{2n}) \times p(A_{2m}) \times p(B_{2n}). \quad (1.13)$$

З цих виразів видно, що вершина "e" умовно не залежить від вершин A_1, A_2, B_1, B_2, так як немає стрілок безпосередньо з'єднують ці вершини.

Розглянувши ці приклади спробуємо тепер більш точно визначити основні поняття, що використовуються в БСД. Байєсовські мережі довіри - це спрямований ациклічний граф, що володіє наступними властивостями:

– кожна вершина являє собою подія, що описується випадковою величиною, яка може мати кілька станів;
– всі вершини, пов'язані з "батьківськими" визначаються таблицею умовних ймовірностей (ТУВ) або функцією умовних ймовірностей (ФУВ);
– для вершин без "батьків" ймовірності її станів є безумовними (маргінальними).

Іншими словами, в байесовских мережах довіри вершини представляють собою випадкові змінні, а дуги - імовірнісні залежності, які визначаються через таблиці умовних ймовірностей. Таблиця умовних ймовірностей кожної вершини містить ймовірності станів цієї вершини за умови станів її “батьків”.

1.3.6.1 Приклад побудови найпростішої байєсівської мережі довіри

У цьому прикладі розглядаємо невелику яблучну плантацію "яблучного Джека". Одного разу Джек виявив, що його прекрасне яблучне дерево втратило листя. Тепер він хоче з'ясувати, чому це сталося. Він знає, що листя часто обпадає, якщо:

– дерево засихає в результаті нестачі вологи;
– або дерево хворіє.

Дана ситуація може бути змодельована байєсівської мережею довіри, що містить 3 вершини: "Хворіє", "засохли" і "Облетіло".

Рис.1.6. Приклад байєсівської мережі довіри з трьома подіями.

В даному найпростішому випадку розглянемо ситуацію, при якій кожна вершина може приймати всього лише два можливих станів і, як наслідок знаходиться в одному з них, а саме:

Таблиця 1.1 – Байєсівської мережі довіри

Вершина БСД	**Стан 1**	**Стан 2**
"Хворе"	"хворе"	"так"
"Засохло"	"засохло"	"так"
"Облетело"	"так"	"так"

Вершина "Хворіє" говорить про те, що дерево захворіло, якщо воно знаходиться в стані "хворіє", в іншому випадку вона знаходиться в стані "ні". Аналогічно для інших двох вершин. Вже згадана баєсова мережа довіри, моделює той факт, що є причинно-наслідковий залежність від події "Хворіє" до події "Облетіло" і від події "засохли" до події "Облетіло". Це відображено стрілками на байєсівської мережі довіри.

Коли є причинно-наслідковий залежність від вершини A до іншої вершини B, то ми очікуємо, що коли A знаходиться в деякому певному стані, це впливає на стан B. Слід бути уважним, коли моделюється залежність в байесовских мережах довіри. Іноді зовсім не очевидно, який напрямок повинна мати стрільця.

Наприклад, в розглянутому прикладі, ми говоримо, що є залежність від "Хворіє" до "Облетіло", так як коли дерево хворіє, це може викликати опадання його листя. Обпадання листя є наслідком хвороби, а не хвороба - наслідком опадання листя.

На наведеному вище малюнку дано графічне представлення байєсівської мережі довіри. Однак, це тільки якісне уявлення байєсівської мережі довіри. Перед тим, як назвати це повністю байєсівської мережею довіри необхідно визначити кількісне уявлення, тобто безліч таблиць умовних ймовірностей:

Таблиця 1.2 - ТУВ для трьох вершин байєсівської мережі довіри

Априорная вероятность *p*(“Болеет”)			Априорная вероятность *p*(“Засохло”)	
Болеет = «болеет»	Болеет = «нет»		Засохло = «засохло»	Засохло = «нет»
0,1	0,9		0,1	0,9

Таблица условных вероятностей *p*(“Облетело” | ”Болеет”, ”Засохло”)				
	Засохло = «засохло»		Засохло = «нет»	
	Болеет = «болеет»	Болеет = «нет»	Болеет = «болеет»	Болеет = «нет»
Облетело = «да»	0,95	0,85	0,90	0,02
Облетело = «нет»	0,05	0,15	0,10	0,98

Наведені таблиці ілюструють ТУВ для трьох вершин байєсівської мережі довіри. Зауважимо, що всі три таблиці показують імовірність перебування деякої вершини в певному стані, обумовленим станом її батьківських вершин. Але так як вершини Хворіє і засохли не мають батьківських вершин, то їх ймовірності є маргінальними, тобто не залежить (не залежать від) ні від чого.

На даному прикладі ми розглянули, що і як описується дуже простий байєсівської мережею довіри. Сучасні програмні засоби (такі як MSBN, Hugin і ін.) Забезпечують інструментарій для побудови таких мереж, а також можливість використання байесовских мереж довіри для введення нових свідоцтв і отримання рішення (висновку) за рахунок перерахунку нових можливостей у всіх вершинах, відповідних зновувведенням свідченнями .

У нашому прикладі нехай відомо, що дерево скинуло листя. Це свідчення вводиться вибором стану "та" в вершині "Облетіло".

Після цього можна дізнатися ймовірності того, що дерево засохло. Для наведених вище вихідних даних, результати виведення шляхом поширення ймовірностей по БСД будуть.

1.4 Постановка наукової задачі та обґрунтування досліджень

Будь-яка рекомендаційна система допомагає вирішувати певну бізнес-завдання. А результат вимірюється зрозумілими для бізнес-завдання способами - кількістю відвідувачів, продажами, CTR, і т.д. Однак якість рекомендаційного алгоритму таким способом виміряти занадто складно - воно буде залежати від величезної кількості умов, серед яких рекомендаційний алгоритм сам по собі може виявитися справою іншою.

Так виявляється, що формальні чисельні критерії до рекомендаційних алгоритмам потрібно розробляти у відрив від бізнес-завдання. В результаті, роботи, присвячені рекомендаційним системам, сповнені різними корисними чисельним метриками, але іноді досить складним по відношенні до важливість справ.

Оптимальний підхід, в якому йтиметься - це спробувати сконструювати метрику безпосередньо під завдання.

1.5 Висновки до першого розділу

У рекомендаційних системах використовується явний або неявний збір даних. При явному зборі від користувача потрібно заповнювати опитувальні анкети для виявлення переваг, а при неявному зборі для виявлення переваг користувача і складання рейтингів відбувається автоматичне протоколювання його дій. Найбільш очевидний спосіб неявного збору інформації характерний для систем електронної комерції, де рейтинг товара у користувача оцінюється в залежності від кількості замовлених одиниць, включених користувачем в свій замовлення.

Моделі нізкорангових наближень мають величезний потенціал для вирішення завдань колаборативної фільтрації, а також з інших предметних областей, в яких мають місце взаємодії між об'єктами різних типів.

Проведено огляд технік побудови рекомендаційних систем, виявлено потребу в алгоритмі колаборативної фільтрації, що враховує довільні опису факторів, здатному працювати для даних з неявній зворотним зв'язком (implicit feedback). Також, проведено огляд алгоритмів колаборативної фільтрації.

РОЗДІЛ 2
АНАЛІЗ АЛГОРИТМІВ І МОДЕЛЕЙ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

2.1 Завдання рекомендаційної системи

Завдання рекомендаційної системи - проінформувати користувача про товар, який йому може бути найбільш цікавий в даний момент часу. Клієнт отримує інформацію, а сервіс заробляє на наданні якісних послуг. Послуги - це не обов'язково прямі продажі товару. Сервіс також може заробляти на комісійних або просто збільшувати лояльність користувачів, яка потім виливається в рекламні та інші доходи.

Залежно від моделі бізнесу рекомендації можуть бути його основою, як, наприклад, у TripAdvisor, а можуть бути просто зручним додатковим сервісом (як, наприклад, в якомусь інтернет-магазині одягу), покликаним поліпшити Customer Experience і зробити навігацію по каталогу більш зручною.

Персоналізація онлайн-маркетингу - очевидний тренд останнього десятиліття. За оцінками McKinsey, 35% виручки Amazon або 75% Netflix припадає саме на рекомендовані товари і відсоток цей, ймовірно, буде рости. Рекомендаційні системи - це про те, що запропонувати клієнту, щоб зробити його щасливим.

Щоб проілюструвати все різноманіття рекомендаційних сервісів, наведу список основних характеристик, за допомогою яких можна описати будь-яку рекомендаційну систему. [2]

Предмет рекомендації - що рекомендується. Тут велика різноманітність - це можуть бути товари (Amazon, Ozon), статті (Arxiv.org), новини (Surfingbird, Яндекс.Дзен), зображення (500px), відео (YouTube, Netflix), люди (Linkedin, LonelyPlanet), музика (Last.fm, Pandora), плейлисти та інше. В цілому, рекомендувати можна що завгодно.

Мета рекомендації - навіщо рекомендується. Наприклад: покупка, інформування, навчання, заклад контактів.

Контекст рекомендації - що користувач в цей момент робить. Наприклад: дивиться товари, слухає музику, спілкується з людьми.

Джерело рекомендації - хто рекомендує:

- аудиторія (середній рейтинг ресторану в TripAdvisor),
- схожі за інтересами користувачі,
- експертне співтовариство (буває, коли мова про складне товар, такому, як, наприклад, вино).
- Ступінь персоналізації.

Не персональні рекомендації - коли вам рекомендують те ж саме, що всім іншим. Вони допускають таргетинг по регіону або часу, але не враховують ваші особисті переваги.

Більш просунутий варіант - коли рекомендації використовують дані з вашої поточної сесії. Ви подивилися кілька товарів, і внизу сторінки вам пропонуються схожі.

Персональні ж рекомендації використовують всю доступну інформацію про клієнта, в тому числі історію його покупок.

Прозорість – це коли люди більше довіряють рекомендації, якщо розуміють, як саме вона була отримана. Так менше ризик нарватися на «недобросовісні» системи, які просувають проплачений товар або ставлять більш дорогі товари вище в рейтингу. Крім того, хороша рекомендаційна система сама повинна вміти боротися з купленими відгуками і накрутками продавців.

Маніпуляції до речі бувають і ненавмисними. Наприклад, коли виходить новий блокбастер, насамперед на нього йдуть фанати, відповідно, першу пару місяців рейтинг може бути сильно завищений.

Формат рекомендації - це може бути спливаюче віконце, що з'являється в певному розділі сайту відсортований список, стрічка внизу екрана або щось ще.

Алгоритми, незважаючи на безліч існуючих алгоритмів, всі вони зводяться до декількох базових підходів, які будуть описані далі. До найбільш класичним відносяться алгоритми Summary-based, Content-based (моделі засновані на описі товару), Collaborative Filtering (колаборативна фільтрація), Matrix Factorization (методи засновані на матричному розкладанні) і деякі інші. [9]

У центрі будь-рекомендаційної системи знаходиться так звана матриця переваг. Це матриця, по одній з осей якої відкладені всі клієнти сервісу (Users), а по інший - об'єкти рекомендації (Items). На перетині деяких пар (user, item) дана матриця заповнена оцінками (Ratings) - це відомий нам показник зацікавленості користувача в даному товарі, виражений за заданою шкалою (наприклад від 1 до 5).

Користувачі зазвичай оцінюють лише невелику частину товарів, що є в каталозі, і завдання рекомендаційної системи - узагальнити цю інформацію і передбачити ставлення клієнта до інших товарів, про які нічого не відомо. Іншими словами потрібно заповнити всі незаповнені клітинки на зображенні вище.

Шаблони споживання у людей різні, і не обов'язково повинні рекомендуватися нові товари. Можна показувати повторні позиції, наприклад, для поповнення запасу. За цим принципом виділяють дві групи товарів.

Повторювані. Наприклад, шампуні або бритвені верстати, які потрібні завжди.

Неповторні, наприклад, книги або фільми, які рідко набувають повторно.

Якщо продукт не можна явно віднести до одного з класів, має сенс визначати допустимість повторних покупок індивідуально (хтось ходить в магазин тільки заради арахісового масла певної марки, а кому-то важливо спробувати все, що є в каталозі).

Поняття «цікавинки» теж суб'єктивне. Деяким користувачам потрібні речі тільки з їхньої улюбленої категорії (conservative recommendations), а хтось, навпаки, більше відгукується на нестандартні товари або групи товарів (risky recommendations). Наприклад, відеохостинг може рекомендувати користувачеві тільки нові серії улюбленого серіалу, а може періодично закидати йому нові шоу або взагалі нові жанри. В ідеалі варто вибирати стратегію показу рекомендацій під кожного клієнта окремо, за допомогою моделювання категорії клієнта. [23]

Призначені для користувача оцінки можна отримати двома способами:

– явно (explicit ratings) - користувач сам ставить рейтинг товару, залишає відгук, лайкать сторінку,
– неявно (implicit ratings) - користувач явно своє ставлення не виражає, але можна зробити непрямий висновок з його дій: купив товар - значить він йому подобається, довго читав опис - значить є інтерес і т.п.

Звичайно, явні переваги краще - користувач сам говорить про те, що йому сподобалося. Однак на практиці далеко не всі сайти надають можливість явно виражати свій інтерес, та й не всі користувачі мають бажання це робити. Найчастіше використовуються відразу обидва типи оцінок і добре доповнюють один одного.

Також важливо відрізняти терміни Prediction (прогноз ступеня інтересу) і власне Recommendation (показ рекомендації). Що і як показувати - це окреме завдання, яка використовує отримані на кроці Prediction оцінки, але може бути реалізована по-різному.

Іноді термін "рекомендація" вживають в ширшому сенсі і мають на увазі будь-яку оптимізацію, будь то вибірка клієнтів для рекламної розсилки, визначення оптимальної ціни пропозиції або просто вибір найкращої стратегії комунікацій з клієнтом. У статті я обмежуся класичним визначенням даного терміна, що означає вибір найбільш цікавого товару для клієнта.

2.1.1 Неперсоналізовані рекомендації

Почнемо з неперсоналізовані рекомендацій, оскільки вони найпростіші в реалізації. У них потенційний інтерес користувача визначається просто середнім рейтингом товару:

«Усім подобається - значить сподобається і вам». За цим принципом працює більшість сервісів, коли користувач не авторизується в системі, наприклад, той же TripAdvisor.

Показуватися рекомендації можуть по-різному - як банер збоку від опису товару (Amazon), як результат запиту, відсортоване за певним параметру (TripAdvisor), або якось ще. [19]

Середній рейтинг від покупців також може зображуватися різними способами. Це можуть бути зірочки поруч з товаром, кількість лайків, різниця позитивних і негативних голосів (як зазвичай роблять на форумах), частка високих оцінок або взагалі гістограма оцінок. Гістограми - найбільш інформативний спосіб, але у них є один мінус - їх складно порівнювати між собою або сортувати, коли потрібно вивести товари списком.

2.1.2 Проблема холодного старту

Холодний старт - це типова ситуація, коли ще не накопичено достатню кількість даних для коректної роботи рекомендаційної системи (наприклад, коли товар новий або просто його дуже рідко купують). Якщо середній рейтинг пораховано за оцінками всього трьох користувачів (ueser1, ueser2 і ueser3), така оцінка явно не буде достовірною, і користувачі це розуміють. Часто в таких ситуаціях рейтинги штучно коректують.

Перший спосіб - показувати не середнє значення, а згладжені середнє (Damped Mean). Сенс такий: при малій кількості оцінок відображається рейтинг більше тяжіє до нікому безпечного «середнього» показнику, а як тільки набирається достатня кількість нових оцінок, «усереднюються» коригування перестає діяти.

Інший підхід - розраховувати по кожному рейтингу інтервали достовірності (confidence Intervals). Математично, чим більше оцінок, тим менше варіація середнього і, отже, більше впевненість в його правильності. А в якості рейтингу можна виводити, наприклад, нижню межу інтервалу (Low CI Bound). При цьому зрозуміло, що така система буде досить консервативною, з тенденцією до заниження оцінок з нових товарів (якщо, звичайно, це не хіт). [4]

Оскільки оцінки обмежені певною шкалою (наприклад від 0 до 1), звичайний спосіб розрахунку інтервалу достовірності тут погано застосуємо: через хвостів розподілу, що йдуть на нескінченність і симетричності самого інтервалу. Є альтернативний і більш точний спосіб його порахувати - Wilson Confidence Interval. При цьому виходять несиметричні інтервали приблизно такого вигляду. [21]

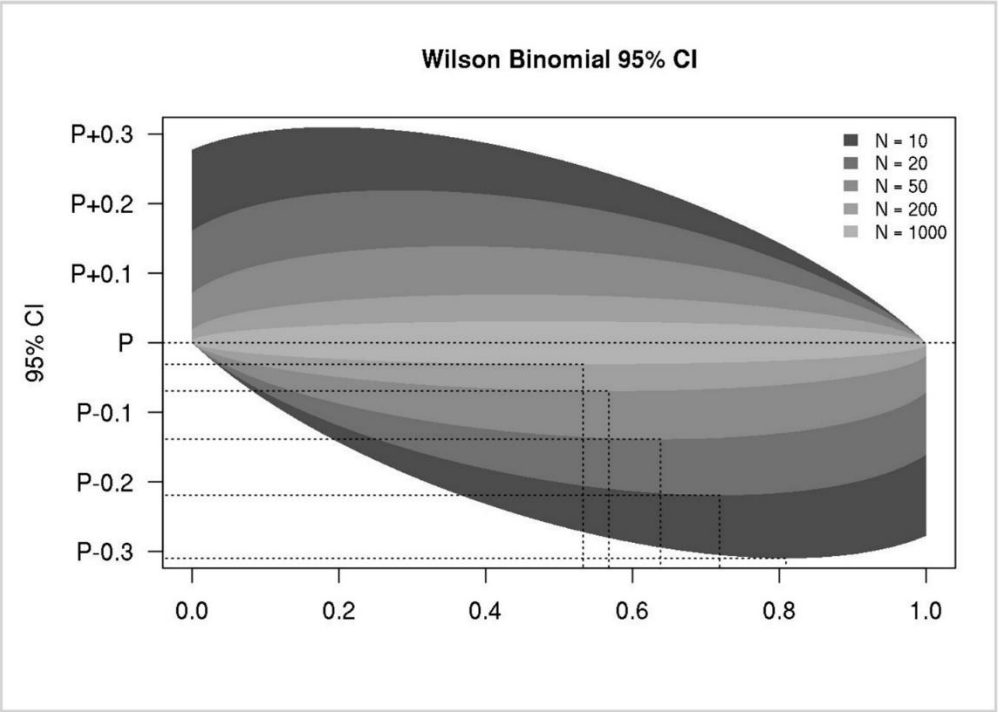


Рис. 2.1 Wilson Confidence Interval

На зображенні вище по горизонталі відкладена оцінка середнього значення рейтингу, по вертикалі - розкид навколо середнього значення. Кольором виділені різні розміри вибірки (очевидно, ніж вибірка більше, тим менше інтервал достовірності).

Проблема холодного старту так само актуальна і для неперсоналізовані рекомендацій. Загальний підхід тут - замінювати те, що в даний момент не може бути пораховано, різними евристиками (наприклад, замінювати середнім рейтингом, використовувати алгоритм простіше, або взагалі не використовувати товар, поки не будуть зігнані дані).

2.1.3 Актуальність рекомендацій

У деяких випадках також важливо враховувати «свіжість» рекомендації. Це особливо актуально для статей або постів на форумах. Свіжі записи повинні частіше потрапляти в топ. Для цього використовуються коригувальні коефіцієнти (damping factors). Нижче пара формул для розрахунку рейтингу статей на медіа сайтах.

Приклад розрахунку рейтингу в журналі Hacker news:

$$Rank = \frac{(U-D-1)^{0.8} * P}{T^{1.8}} \tag{2.1}$$

де U = upvotes;

D = downvotes;

P (Penalty) - додаткова коригування для імплементації інших бізнес-правил.

Розрахунок рейтингу в Reddit:

$$Rank = log_{10}(max(1, U - D)) - \frac{|U-D|T}{const} \quad (2.2)$$

де U = число голосів «за»;

D = число голосів «проти»;

T = час запису.

Перший доданок оцінює «якість запису», а друге робить поправку на час.

Очевидно, що універсальної формули не існує, і кожен сервіс винаходить ту формулу, яка найкраще вирішує його завдання - перевіряється це емпірично.

2.1.4 Оцінка якості системи

Тестування рекомендаційної системи - процес непростий і завжди викликає безліч питань, головним чином через неоднозначність самого поняття «якість».

Взагалі, в задачах машинного навчання є два основні підходи до тестування:

– offline тестування моделі на історичних даних за допомогою ретро-тестів,
– тестування готової моделі за допомогою A / B тестування (запускаємо кілька варіантів, дивимося який дає кращий результат).

Обидва ці підходи активно застосовуються при розробці рекомендаційних систем. Почнемо з offline тестування.

Основне обмеження, з яким доводиться стикатися - оцінити точність прогнозу ми можемо тільки на ті товари, які користувач уже оцінив.

Стандартний підхід - це крос-валідація методами leave-one-out і leave-p-out. Багаторазове повторення тесту з усередненням результатів дозволяє отримати більш стійку оцінку якості.

– leave-one-out - модель навчається на всіх оцінених користувачем об'єктах, крім одного, а тестується на цьому одному об'єкті. Так робиться для всіх n об'єктів, і серед отриманих n оцінок якості обчислюється середнє.
– leave-p-out - те ж саме, але на кожному кроці виключається p точок.
– Всі метрики якості можна умовно розділити на три категорії:

– Prediction Accuracy - оцінюють точність пророкує рейтингу,
– Decision support - оцінюють релевантність рекомендацій,
– Rank Accuracy метрики - оцінюють якість ранжирування видаються рекомендацій.

На жаль, не існує єдиної рекомендованої метрики на всі випадки життя і кожен, хто займається тестуванням рекомендаційної системи, підбирає її під свої цілі.

Коли рейтинги оцінюються по безперервної шкалою (0-10), як правило, досить метрик класу Prediction Accuracy. [14]

Таблиця 2.1 – Формули помилок

Назва	**Формула**	**Опис**
MAE (Mean Absolute Error)	$E(\|P-R\|)$	Середнє абсолютне відхилення
MSE (Mean Squared Error)	$E(\|P-R\|^2)$	Середньоквадратичної помилка
RMSE (Root Mean Squared Error)	$\sqrt{E(\|P-R\|^2)}$	Корінь з середньоквадратичної помилки

Метрики класу Decision Support працюють з бінарними даними (0 і 1, та й немає). Якщо в нашій задачі рейтинги спочатку відкладаються по безперервної шкалою, їх можна перевести в бінарний формат, застосувавши вирішальне правило - скажімо, якщо оцінка менше 3.5, вважаємо оцінку «поганий», а якщо більше, то «хорошою».

Таблиця 2.2 - Метрики класу Decision Support

Назва	Формула	Опис
Precision	$\frac{TP}{TP+FP}$	Частка рекомендацій, що сподобалися користувачеві
Recall	$\frac{TP}{TP+TN}$	Частка цікавих користувачеві товарів, яка показана
F1-Measure	$\frac{2PR}{P+R}$	Середнє гармонійне метрик Precision і Recall. Корисно, коли заздалегідь неможливо сказати, яка з метрик важливіше
ROC AUC		Наскільки висока концентрація цікавих товарів на початку списку рекомендацій
Precision@N		Метрика Precision, обчислена, виходячи на Top-N записах
Recall@N		Метрика Recall, обчислена, виходячи на Top-N записах
AverageP		Середнє значення Precision на всьому списку рекомендацій

Як правило, рекомендації виводяться списком з декількох позицій (спочатку топова, потім в порядку убування пріоритету). Метрики класу Rank Accuracy вимірюють наскільки правильний порядок показу рекомендацій в відсортованому списку.

Таблиця 2.3 - Метрики класу Rank Accuracy

Назва	Формула	Опис
Mean Reciprocal Rank	$E\left(\frac{1}{pos}\right)$	На якій позиції списку рекомендацій користувач знаходить першу корисну
Spearman Correlation	$E\left(\lvert P-R\rvert^2\right)$	Кореляція (Спірмена) реального і прогнозованого рангів рекомендацій
nDCG	$\sum \frac{R(i)}{max(1,log(i))}$	Інформативність видачі з урахуванням ранжирування рекомендацій
Fraction of Concordance Pairs	$P\left(X_R > X_P\right)$	Наскільки висока концентрація цікавих товарів на початку списку рекомендацій

Якщо ми візьмемо рекомендаційні системи в онлайн бізнесі, то вони, як правило, переслідує дві (іноді суперечливі) цілі:

а) проінформувати користувача про цікавий товар,
б) спонукати його зробити покупку (шляхом розсилки, складання персонального пропозиції і т.д.).

як і в будь-якої моделі, спрямованої на мотивацію користувача до дії, оцінювати слід тільки інкрементальний приріст цільового дії. Тобто, наприклад, при підрахунку покупок за рекомендацією нам потрібно виключити ті, які користувач і так сам би зробив без нашої моделі. Якщо цього не зробити, ефект від впровадження моделі буде сильно завищений.

Lift - показник того, у скільки разів точність моделі перевершує якийсь baseline алгоритм. У нашому випадку baseline алгоритмом може бути просто відсутність рекомендацій. Дана метрика добре відловлює частку інкрементальних покупок і це дозволяє ефективно порівнювати різні моделі.

2.1.5 Неявні рейтинги і унарні данні

На початку свого розвитку рекомендаційні системи застосовувалися в сервісах, де користувач явно оцінює товар шляхом виставлення йому рейтингу - це і Amazon, і Netflix і інші сайти інтернет торгівлі. Однак зі зростанням популярності рекомендаційних систем виникла потреба застосовувати їх ще й там, де ніяких рейтингів немає - це можуть бути банки, автомайстерні, кіоски з шаурмою і будь-які інші сервіси, де з якоїсь причини неможливо налагодити систему оцінювання. У цих випадках інтереси користувача можна обчислити лише за непрямими ознаками - про користувальницьких перевагах говорять певні дії з товаром, наприклад, перегляд опису на сайті, додавання товару в кошик і т.д. Тут використовується принцип «купив - значить любить!». Така система неявного оцінювання називається Implicit Ratings.

Неявні рейтинги очевидно працюють гірше явних, оскільки вносять на порядок більше шуму. Адже користувач міг купити товар в подарунок дружині або зайти на сторінку з описом товару, тільки щоб залишити там коментар в стилі «яка ж все-таки це гидота» або задовольнити свою природну цікавість.

Якщо у випадку з явними рейтингами ми вправі очікувати, що хоч одну негативну оцінку ні-ні та й поставить, то негативну оцінку ми нізвідки не візьмемо. Якщо користувач не купив книгу «П'ятдесят відтінків сірого», він міг це зробити з двох причин:

– вона йому дійсно не цікава (це негативний кейс),
– вона йому цікава, але він просто не знає про неї (це пропущений позитивний кейс).

В обох прикладах завдання перетворюється в задачу Unary Class Classification.

Найбільш очевидний варіант вирішення - йти по простому шляху і вважати відсутність оцінки негативною оцінкою. У деяких випадках це більш виправдано, в деяких - менше. Наприклад, якщо ми знаємо, що користувач товар швидше за все бачив (наприклад, ми йому його показували в списку товарів, а він перейшов на товар, що йде за ним), то відсутність переходу дійсно може говорити про відсутність інтересу.

2.2 Висновки до другого розділу

У цьому розділу ми виявили що завдання рекомендаційної системи щоб проінформувати користувача про товар, який йому може бути найбільш цікавий в даний момент часу. У центрі будь-рекомендаційної системи знаходиться так звана матриця переваг. Це матриця, по одній з осей якої відкладені всі клієнти сервісу (Users), а по інший - об'єкти рекомендації (Items). На перетині деяких пар (user, item) дана матриця заповнена оцінками (Ratings) - це відомий нам показник зацікавленості користувача в даному товарі, виражений за заданою шкалою (наприклад від 1 до 5).

Також виявили проблему холодного старту, бо вона є типовою ситуація, коли ще не накопичено достатню кількість даних для коректної роботи рекомендаційної системи (наприклад, коли товар новий або просто його дуже рідко купують). У додаток, зробили розгляд якості рекомендаційної системи та неявні рейтинги.

РОЗДІЛ 3
ПРАКТИЧНЕ РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНИЙ СИСТЕМИ НА БАЗІ ВЕБ АНАЛІТИКИ ДЛЯ ІНТЕРНЕТ МАГАЗИНУ

3.1 Огляд основної платформі інтернет магазину

3.1.1 Опис OpenCart

OpenCart - система керування вмістом з відкритим кодом, призначена для створення інтернет-магазинів. Розповсюджується за ліцензією GNU General Public License.

OpenCart може бути встановлено на будь який веб-сервер Apache з підтримкою PHP5 і MySQL. Навколо OpenCart сформувалася велика спільнота (понад 46000 учасників), завдяки якій створено понад 8500 безкоштовних розширень у вигляді додаткових модулів.

Найвагомішими перевагами OpenCart над системами Magento, Virtuemart і osCommerce є сучасна MVC-архітектура, підвищена швидкість роботи, vQmod, багатофункціональна адміністративна панель управління контентом та менше споживання серверних ресурсів.

OpenCart добре зарекомендував себе в комерційному секторі як надійна і недорога в обслуговуванні система електронної торгівлі, з підтримкою розрахунку всіма найвідомішими системами електронної оплати.

3.1.2 Основні можливості

Технічні переваги

- Підтримка PHP 5.x і MySQL 4.x, 5.x
- Код відповідає основним принципам шаблону Model-View-Controller, який дозволяє проводити роботи різної складності незалежно одна від одної
- Порівняно з конкурентами (Magento, Virtuemart, osCommerce) має кращі показники швидкості і несе менше навантаження на сервер
- Підтримка багатьох сучасних браузерів.
- Вбудована багатомовність. Доступна українська мова
- Необмежена кількість категорій і товарів
- Підтримка шаблонів, модулів і доповнень

Адміністрування / База

- Підтримка одного і більше магазинів
- Необмежена кількість продуктів і категорій

– Підтримка фізичних і віртуальних (з можливістю завантаження) товарів
– Легкість резервного копіювання і відновлення бази даних
– Статистика товарів, замовлень і продажів
– Багатомовність
– Підтримка різних валют та їхніх курсів

Клієнтська частина

– Реєстрація покупців
– Усі замовлення зберігаються в базі даних для ефективного пошуку історії покупок
– Клієнти можуть переглядати історію і статус своїх замовлень
– Тимчасовий кошик для гостей і постійний для зареєстрованих клієнтів
– Швидкий і зручний інструмент пошуку
– Підтримка SSL (Secure Sockets Layer)
– Зручна навігація по сайту
– Клієнт може мати декілька адрес доставки в персональній адресній книзі

Система оплати і доставки

– Підтримка багатьох типів платежів (чеки, платіжні доручення)
– Підтримка багатьох платіжних систем за допомогою модулів (2CheckOut, PayPal, Authorize.Net, iPayment, RuPay, Webmoney, Приват24)
– Налаштування методів оплати для різних регіонів
– Розрахунок вартості доставки на базі ваги і ціни товару та зони доставки
– Безліч модулів розрахунку вартості доставки
– Розрахунок податків з урахуванням регіону
– Перелік функціоналу постійно розвивається.

Варто звернути увагу:

– Шаблонізація на базі PHP
– В даний час не оптимізована робота SEO
– Швидкі темпи розвитку версій, що потребує частого оновлення ядра

3.2 Реалізація рекомендаційної системи

3.2.1 Огляд даних та створення рекомендаційної системи

Ми розглянули алгоритм колаборативной фільтрації і далі розробимо рекомендаційну систему по схожості користувачів, яка визначається з використанням косинусной схожості по оцінкам продукції. Рекомендоційна система буду використана на сайті з продажу та доставки їжі. Для разробки алгоритму нам потрібні данні о замовленях, ці дані беруться с бази даних SQL *«oc_order_product» (рис. 3.1)*, де поле «order_id» - це ідентифікатор користувача, де зберігаються такі дані як: ім'я, телефон, адрес и т.д. *«product_id»* - продукт з сайту, де поле *«name»* його назва і поле *«reward»* - оцінка котру поставив користувач для цього продукту. Усі описані поля потрібні для розробки рекомендаційною системи для інтернет магазину.

order_	order_id	product_id	name	model	quantity	price	total	tax	reward
68	47	113	Сет канна	Сет	1	620.0000	620.0000	0.0000	4
67	46	151	Сет кумамото	Сет	1	530.0000	530.0000	0.0000	5
54	40	128	Филадельфия	Маки-суши	1	140.0000	140.0000	0.0000	3
55	40	151	Сет кумамото	Сет	1	530.0000	530.0000	0.0000	4
56	41	77	Кадзура	маки- суши	1	120.0000	120.0000	0.0000	3
57	42	108	Салат хияши с кальмаром и тигровой креветкой	Салат	1	90.0000	90.0000	0.0000	5
58	42	116	Сет орхидея	Сет	1	665.0000	665.0000	0.0000	4
59	43	62	Банзай	горячие роллы	1	150.0000	150.0000	0.0000	5
60	43	130	Фудзияма	Маки-суши	1	115.0000	115.0000	0.0000	4
61	44	118	Сет ханами	Сет	1	300.0000	300.0000	0.0000	3
62	45	147	Ролл с красной икрой	маки- суши	1	130.0000	130.0000	0.0000	5
63	45	81	Канада	маки- суши	1	170.0000	170.0000	0.0000	4
64	45	82	Кани-унаги	маки- суши	1	135.0000	135.0000	0.0000	5
65	45	108	Салат хияши с кальмаром и тигровой креветкой	Салат	1	90.0000	90.0000	0.0000	2
66	45	73	Ика темпура	маки- суши	1	130.0000	130.0000	0.0000	4
69	47	151	Сет кумамото	Сет	1	530.0000	530.0000	0.0000	5
70	48	115	Сет купидон	Сет	1	480.0000	480.0000	0.0000	4
71	49	70	дзен	маки- суши	1	165.0000	165.0000	0.0000	5
72	49	130	Фудзияма	Маки-суши	1	115.0000	115.0000	0.0000	5
73	49	81	Канада	маки- суши	1	170.0000	170.0000	0.0000	5
74	49	62	Банзай	горячие роллы	1	150.0000	150.0000	0.0000	5
75	50	116	Сет орхидея	Сет	1	565.0000	565.0000	0.0000	4
76	51	108	Салат хияши с кальмаром и тигровой креветкой	Салат	1	90.0000	90.0000	0.0000	5

Рис. 3.1 Дані о замовленнях

Для рекомендації користувачеві №1 будь-якого продукту, вибирати потрібно з продуктів, які подобаються якимось користувачам 2, 3,4 і так далі, які найбільш схожі за своїми оцінками на користувача №1. Потрібно отримати чисельне вираження цієї «схожості» користувачів. Оцінки, виставлені окремо взятим користувачем, є вектор в M-вимірному просторі продуктів. Косинусна міра для двох векторів - це косинус кута між ними. Косинус кута між двома векторами - це їх скалярний твір, поділений на довжину кожного з двох векторів:

$$cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}||_2 \times ||\vec{y}||_2} \tag{3.1}$$

Для цього нам потрібні *«poruct_id»* та *«reward».* Отже, у нас є матриця переваг користувачів і ми вміємо визначати, наскільки два користувача схожі один на одного. Тепер залишилося реалізувати алгоритм колаборативної фільтрації. Вибрати користувачів, смаки яких найбільше схожі на смаки розглянутого. Для цього для кожного з користувачів потрібно обчислити обрану міру (в нашому випадку косинусному) щодо розглянутого користувача, і вибрати найбільших. Для активного користувача отримуемо данні (рис.3.2).

user_id:	12	24	35	45
текущий	0,7897	0	0,0789	0,879
all				1,7476

Рис. 3.2 Дані о схожості користувачів

Далі для кожного з користувачів помножити його оцінки на обчислену величину заходи, таким чином оцінки більш «схожих» користувачів будуть сильніше впливати на підсумкову позицію продукту. Також, для кожного з продуктів порахувати суму каліброваних оцінок найбільш близьких користувачів, отриману суму розділити на суму заходів обраних користувачів. Сума представлена на рис. 3.3 в рядку «all», підсумкове значення в рядку «getresult»

	product_1	product_2	product_3	product_4	product_5
user_12	2,127	1,892	0	0,789	1,545
user_24	0	0	0	0,48	0,245
user_35	0	0,0158	0,01	0	0
user_45	0,2578	0,0078	0	0	0,0009
all	2,3848	1,9156	0,01	1,269	1,7909
getresult	3,4789	2,5478	0,5788	1,9878	2,4547

Рис. 3.3 Коефіцієнт популярності товару

Отже, у вигляді формули цей алгоритм може бути представлений як

$$r_{u,i} = k \sum_{u' \in U} sim(u, u') r_{u',i} \tag{3.2}$$

де функція sim - обрана нами міра схожості двох користувачів;

U - безліч користувачів;

r - виставлена оцінка;

k - нормувальний коефіцієнт (3.3):

$$k = 1/\sum_{u' \in U} |sim(u, u')| \tag{3.3}$$

Для застосування даного алгоритму рекомендаційною системи та інтеграції його в інтрнет-магазин на OpenCart, був написаний код на PHP (рис 3.4)

```
<?php
function recomend($userID, $userRates, $nBestUsers, $nBestProducts) {
    $matches =  distCosine($userRates[$userID],$userRates[$u]);
foreach ($u  as $userRates) {
if ($u <> $userID) {
    $bestMatches = sorted($matches, lambda(x,y), True);
    foreach ($line as $bestMatches)  {
       echo $related_user; // шаблон профиля пользывателя
}
}
    $sim = dict();
    $sim_all = sum($x[1]);
    foreach ($x as $bestMatches){
if ($x[1] > 0.0) {
$bestMatches  = $x[1];
}
}
    foreach ($relatedUser as $bestMatches) {
        foreach ($product as $userRates[relatedUser]){
            if ($product != $userRates[userID]) {
                if ($product == $sim){
                    $sim[product] = 0.0;
                $sim[$product] += $userRates[$relatedUser][$product] * $bestM
    foreach ($product as $sim) {
        $sim[$product] /= $sim_all;
    $bestProducts = sorted($sim, lambda(x,y),True);
echo $product_template; //Шаблон отображения продукта для OpenCart
 }
}
}
}
?>
```

Рис. 3.4 Код колаборативної фільтрації на PHP

У класичній реалізації алгоритму є один явний мінус - він погано застосуємо на практиці через квадратичної складності. Дійсно, як будь-який метод найближчого сусіда, він вимагає розрахунку всіх попарних відстаней між користувачами (а користувачів можуть бути мільйони). Неважко порахувати, що складність розрахунку матриці

відстаней буде, де n - число користувачів, а m - число товарів. При мільйон користувачів для зберігання матриці відстаней в сирому вигляді, потрібно мінімум 4TB.

Дана проблема частково може бути вирішена покупкою високопродуктивного заліза. Але якщо підходити з розумом, то краще ввести коригування в алгоритм:

– оновлювати відстані ні до кожної покупки, а Батче (наприклад, раз в день),
– не перераховувати матрицю відстаней повністю, а оновлювати її Інкрементальний,
– зробити вибір на користь ітеративних і наближених алгоритмів (наприклад ALS).

Для того щоб алгоритм був ефективний, важливо щоб виконувалося кілька припущень.

– Смаки людей не змінюються часом (або змінюються, але для всіх однаково).
– Якщо смаки людей збігаються, то вони збігаються у всьому.

Так часто буває, коли рекомендовані товари однорідні. Якщо ж це не так, то у пари клієнтів цілком можуть збігатися переваги в їжі. Околиця користувача в просторі переваг (його сусіди), яку ми будемо аналізувати для генерації нових рекомендацій, можна вибирати по-різному. Ми можемо працювати взагалі з усіма користувачами системи, можемо задати певний поріг близькості, можемо вибрати кілька сусідів випадковим чином або брати n найбільш схожих сусідів (це найбільш популярний підхід).

Отже, зробив усі удосконалення нашого коду для інтернет магазину ми отримали наглядний результат на рисунку 3.5

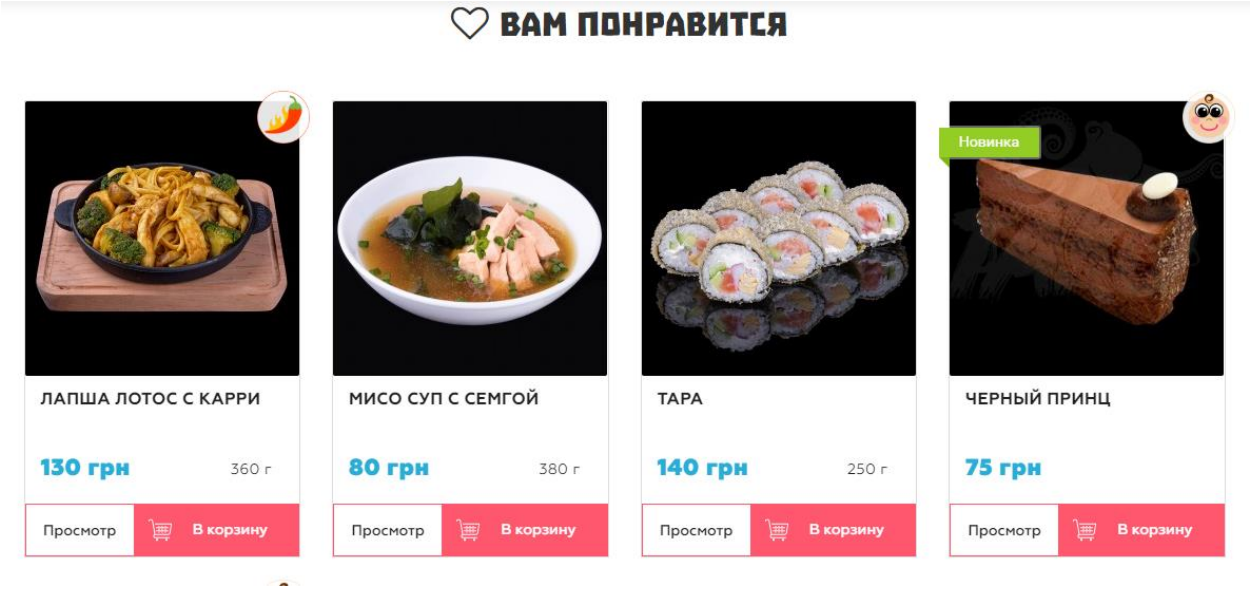


Рис. 3.5 Рекомендації продукту на веб-сайті

3.3 Персоналізована рекомендаційна система спільно з веб-аналітикою

3.3.1 Обробка даних з веб-аналітки для інтернет магазину

Безумовною перевагою сервісу GA є звітність в режимі реального часу. Такий вид звітності призначений для миттєвої оцінки результатів (оперативна оцінка зацікавленості відвідувачів до нових матеріалів сайту, оцінка ефекту.

Широкі можливості Google Analytics дозволяють не тільки використовувати стандартні параметри і показники, а й створювати власні для збору тих даних, які не відслідковуються автоматично. Введення користувальницьких змінних дозволяє зробити аналіз більш гнучким у вивченні взаємодії відвідувачів з вмістом сайту.

Google Analytics Reporting API версії 4 - самий передовий програмний спосіб отримання даних з Google Analytics. Цей інтерфейс дає наступні можливості:

– створювати спеціальні зведення для перегляду даних Google Analytics;
– автоматизувати створення складних звітів;
– використовувати дані Google Analytics в інших бізнес-додатках.

Початок підключеня API до нашего веб-сайту зображено на рис. 3.6

Code.gs × * appsscript.json ×

```
{
  "timeZone": "Asia/Calcutta",
  "dependencies": {
    "enabledAdvancedServices": [{
      "userSymbol": "Analytics",
      "serviceId": "analytics",
      "version": "v3"
    }]
  },
  "webapp": {
    "access": "ANYONE_ANONYMOUS",
    "executeAs": "USER_DEPLOYING"
  },
  "exceptionLogging": "STACKDRIVER",

"oauthScopes": ["https://www.googleapis.com/auth/analytics",
"https://www.googleapis.com/auth/analytics.readonly",
"https://www.googleapis.com/auth/script.container.ui",
"https://www.googleapis.com/auth/script.external_request",
"https://www.googleapis.com/auth/script.send_mail",
"https://www.googleapis.com/auth/spreadsheets"]
}
```

Рис. 3.6 Підключення API до сайту

Таблиця 3.1 - Основні показники

Google Analytics	**API**
Сеанс	ga:sessions
Відмови	ga:bounces
тривалість сеансу	ga:sessionDuration
Мета: № (досягнуті переходи до цілі№)	ga:goalXXCompletions
Досягнуті цілі	ga:goalCompletionsAll
Користувачі	ga:users

Отриманні данні ми зберігаємо у власній базі даних, для покращення швидкості рекомендацій для користувачів.

	A	B	C	D
6	Contains Sampled Data	FALSE		
7				
8	Totals For All Results			
9		ga:visitors	ga:visits	ga:pageviews
10		25427922	36652479	196078188
11				
12	ga:date	ga:visitors	ga:visits	ga:pageviews
13	7/24/2012	1005076	1496419	8527591
14	7/25/2012	990405	1478556	8288818
15	7/26/2012	974560	1448241	8133244
16	7/27/2012	835253	1217863	6588438
17	7/28/2012	502197	734101	3544563
18	7/29/2012	648697	923266	4558041
19	7/30/2012	1235844	1730840	8776544
20	7/31/2012	1080240	1582784	8958766
21	8/1/2012	1335327	1820586	9053423
22	8/2/2012	1027152	1500316	8387924
23	8/3/2012	862318	1246976	6770756
24	8/4/2012	511405	744169	3566201
25	8/5/2012	657646	934436	4606254
26	8/6/2012	1262327	1754775	8840669
27	8/7/2012	1015676	1501851	8506009
28	8/8/2012	978386	1452978	8179291
29	8/9/2012	939440	1397524	7858495
30	8/10/2012	821888	1193644	6446368
31	8/11/2012	506482	736625	3521449
32	8/12/2012	614748	876594	4267081
33	8/13/2012	1209940	1675513	8428013
34	8/14/2012	922853	1364158	7929869
35	8/15/2012	882365	1304730	7462165
36	8/16/2012	932327	1381302	7873382
37	8/17/2012	815299	1183101	6514820
38	8/18/2012	506676	733154	3530669
39	8/19/2012	623174	880140	4374023
40	8/20/2012	1211578	1671344	8495697
41	8/21/2012	518643	686493	4089625

3.7 Вихідні дані з GA

3.4 Реалізація рекомендацій по данним з GA для веб сайту

Нам важливо розуміти бажання людини для створення правильної рекомендаційною системи, спираючись на час, день тижня і відвідуваність сайту в реальному часі. Отримуючи дані з аналітики, ми можемо прогнозувати бажання відвідувача, наприклад:

- Пропонувати відвідувачеві супи або салати якщо це обідній час
- Якщо це вихідний день або кінець робочого тижня, відвідувач не поспішати робити і готовий спробувати щось нове.
- У святкові дні буде великий попит на нові пропозиції

Для цього ми будемо використовувати рекомендаційний систему з веб аналітикою що б враховувати всі чинники які вплинуть на замовлення людини.

За допомогою моделі аналізу поведінки споживачів на ринку ми можемо зрозуміти, що саме йому належить дізнатися про поведінку споживачів, які кроки для цього зробити. Але має бути ще правильно оцінити реакцію споживачів як на самі вироби, так і на заходи в області маркетингу, внести відповідні корективи в намічений курс дій.

Важливо завжди бачити загальні тенденції, щоб стратегія формувалася в руслі цих тенденцій або по меншій мірі не суперечила їм. Перш за все слід якомога точніше дізнатися смаки і нахили окремих груп споживачів, що вони найбільше цінують в речах, якими користуються, які властивості виробів або якісні параметри послуг їм найбільше до душі.

Раз у споживача з'явилися відчуття, за ними безпосередньо випливають певні запити і переваги, причому ще до того, як людина усвідомлює, що ж насправді йому потрібно (який виріб, яка модель, якої фірми і т.п.). Власне, запит споживача можна визначити як інертний стан людини, що виявляється в його активній поведінці в разі адекватної мотивації. Іншими словами, коли людина починає усвідомлювати свої запити, йому потрібно ще досить сильна мотивація, щоб він почав діяти на ринку в пошуку способу задоволення своїх запитів.

У міру взаємодії користувача з системою (скажімо, він купує суші), векторні опису придбаних ним товарів об'єднуються (підсумовуються і нормалізуються) в єдиний вектор і, таким чином, формується вектор його інтересів. Далі досить знайти товар, опис якого найближче до вектору інтересів, тобто вирішити задачу пошуку n найближчих сусідів.

Не всі елементи однаково значущі: наприклад, союзні слова, очевидно, не несуть ніякої корисної навантаження. Тому при визначенні числа співпадаючих елементів в двох векторах все вимірювання потрібно попередньо зважувати по їх значимості. Дану задачу вирішує добре відоме в Text Mining перетворення TF-IDF, яке призначає більшу вагу більш рідкісним інтересам. Збіг таких інтересів має більше значення при визначенні близькості двох векторів, ніж збіг популярних. Принцип TF-IDF тут в тій же мірі можна застосувати і до звичайних номінальним атрибутам, таким, як наприклад, категорія, вага, склад. TF - міра значущості атрибута для користувача, IDF - міра «рідкості» атрибута.

В якості запобіжного близькості двох векторів найчастіше використовується косинусное відстань.

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{3.4}$$

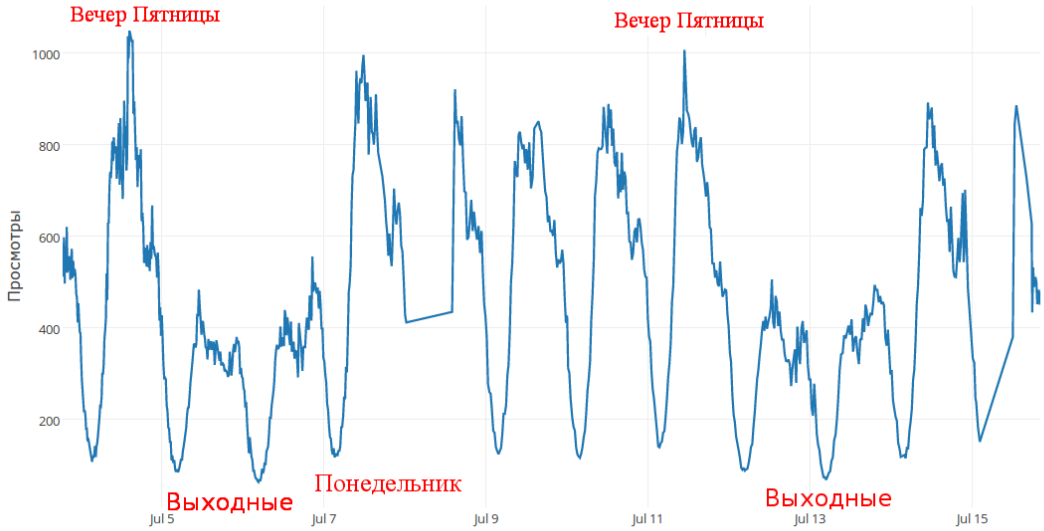


Рис. 3.7 Графік зміни відвідувачів по хвилинах

Першим етап роботи алгоритму спільно з веб аналітикою, рекомендації на початку дня. Як показують загальні дані по замовленнях, в першій половині дня покупці віддають перевагу більш легку їжу і не готові викладати гроші на сніданок. Так що, результат роботи алгоритму для початку робочого дня, це легкий салат з десертом і напоєм.

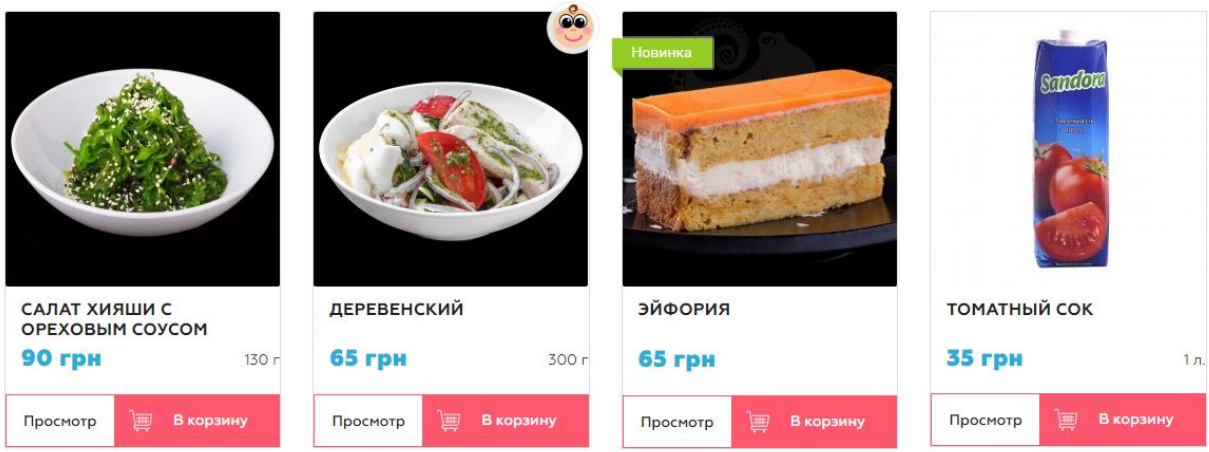


Рис. 3.8 Рекомендації в першій половині робочого дня

На основі даних які ми отримуємо від аналітики та результатом роботи алгоритму отримуємо рекомендації для відвідувача в обідню перерву. Дана рекомендується оптимально підходить для відвідувача в перерві дня. В дану рекомендацію під даний час входить гаряче блюдо (суп), друге (лапша або рис) і легкий десерт (рис. 3.9)

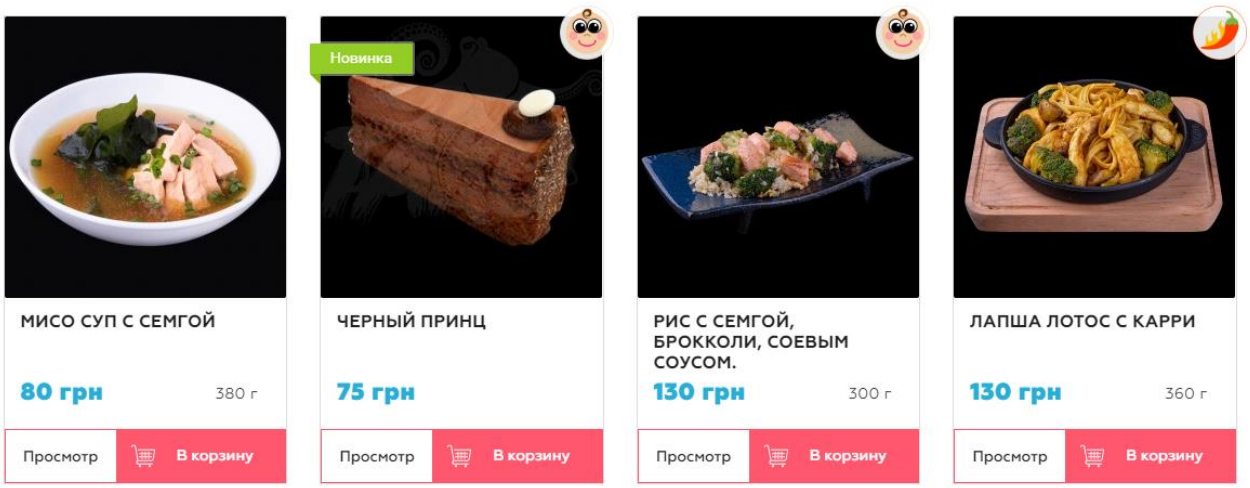


Рис. 3.9 Рекомендація в обідню перерву робочого дня

Наступний етап в роботі системи, це рекомендації для кінця робочого дня. Важливо завжди бачити загальні тенденції, щоб стратегія формувалася в руслі цих тенденцій або по меншій мірі не суперечила їм. Перш за все слід якомога точніше дізнатися смаки і нахили окремих груп споживачів, що вони найбільше цінують в речах, якими користуються, які властивості виробів або якісні параметри послуг їм найбільше до душі. За даними отриманими з попередніх замовлень, можна зробити висновок що в кінці робочого дня люди більш відкриті до покупок в більших обсягах, проте не так як в святкові дні або вихідні дні. За допомогою Google Analytics ми отримуємо пікову кількість відвідувань на сайт, в режимі реального часу, так ми визначаємо, що кінець дня у відвідувачів закінчено, якщо використовувати статичний графік по дню (наприклад, з 17: 00 до 20:00) ми можемо не врахувати короткі дні, або пробки в місті, коли люди напружені і не готові до нав'язливих пропозицій. Як тільки ми отримуємо наплив відвідувачів алгоритм налаштовується на більш «агресивну» рекомендацію, пропонуючи більш широкий асортимент (рис. 3.10)

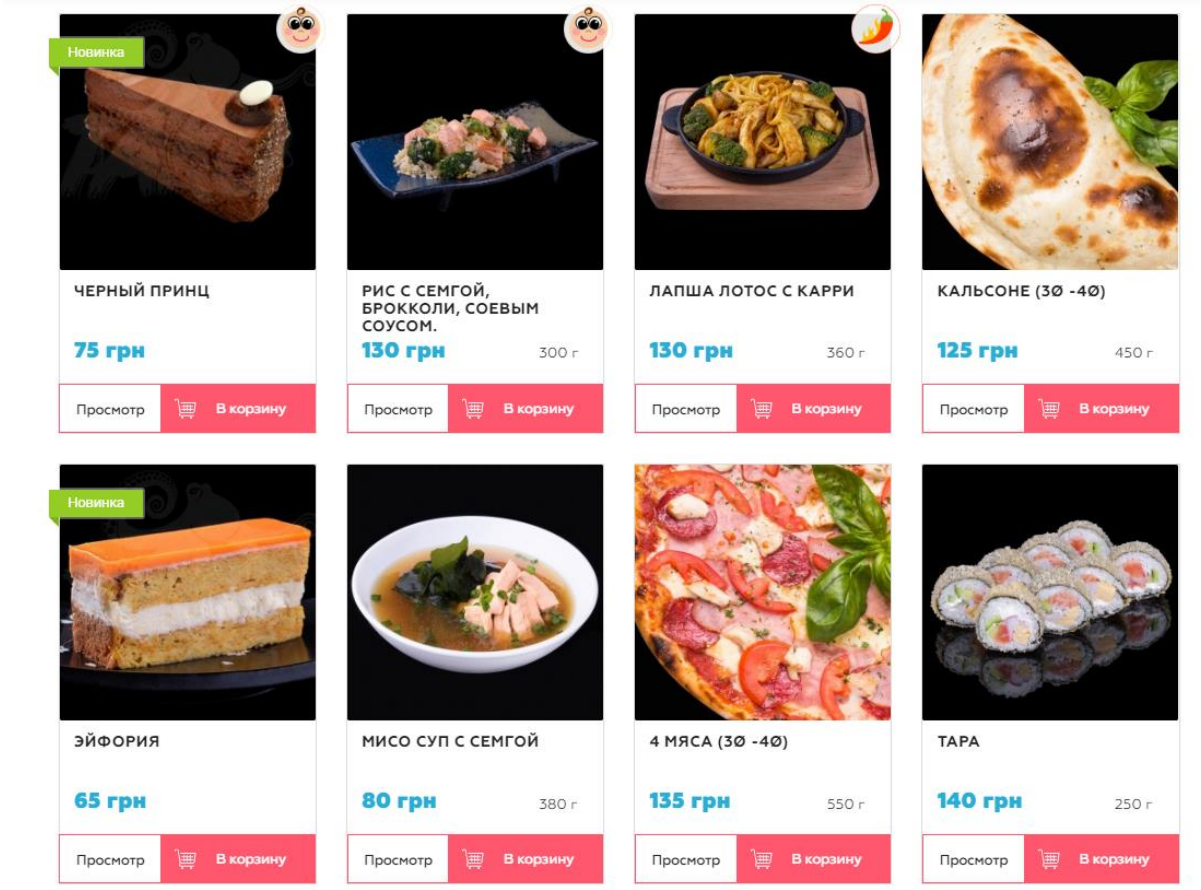


Рис. 3.10 «Агресивна» рекомендація в завершені робочого дня

Для інших днів, таких як вихідні, свята або кінець дня п'ятниці (ми його відносимо як вихідний). Формується також більш розширена рекомендаційна система як на рис. 3.10.

Пікові відвідування за тиждень по годинам, можна побачити, з графіка Google Analytics на малюнку 3.11

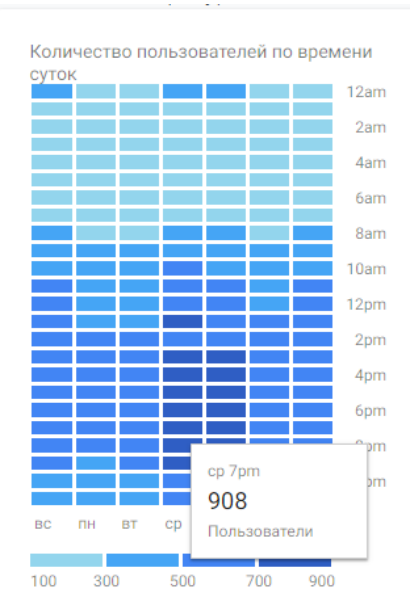


Рис. 3.11 Відвідуваність по годинах

3.5 Висновки до третього розділу

За допомогою моделі аналізу поведінки споживачів на ринку ми можемо зрозуміти, що саме йому належить дізнатися про поведінку споживачів, які кроки для цього зробити. Але має бути ще правильно оцінити реакцію споживачів як на самі вироби, так і на заходи в області маркетингу, внести відповідні корективи в намічений курс дій.

Важливо завжди бачити загальні тенденції, щоб стратегія формувалася в руслі цих тенденцій або по меншій мірі не суперечила їм.

Ми розглянули алгоритм колаборативної фільтрації і content based, та далі розробили рекомендаційну систему по схожості користувачів та товарів, яка визначається з використанням косинусної схожості по оцінкам продукції.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної магістерської роботи було використання інформаційних технологій для аналізу методів веб-аналітики, областю застосування веб-аналітики є розширення функціональності сайтів, вимір для користувача активності і оцінка ефективності рекламних інтернет-ресурсів. Обробляти отриманні данні, та формування рекомендаційних систем, їх розробку і розвиток впливали кілька різних областей знань.

Так як в процесі проектування використовувалося ПК, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде використовуватися розроблена об'єктна модель для оцінки ризиків та огляд технік побудови рекомендаційних систем, виявлено потребу в алгоритмі фільтрації, що враховує довільні описи факторів.

4.1 Загальні питання з охорони праці

В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів шиї, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці.

4.1.1 Правові та організаційні основи охорони праці

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

Користувачі персональних комп'ютерів, для яких ця робота є головною, підлягають медичним оглядам: попереднім — під час влаштування на роботу і періодичним — протягом професійної діяльності раз на два роки. Жінок з часу встановлення вагітності та в період годування дитини грудьми до роботи з ПК не допускають.

Обов'язки працівників щодо додержання вимог нормативно-правових актів з охорони праці (ст. 14), відповідальність робітників всіх категорій за порушення вимог щодо охорони праці (ст. 44) та структура організації/виробництв системи управління охорони праці визначені у [1].

4.1.2 Організаційно-технічні заходи з безпеки праці

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [2].

Також впроваджені організаційні заходи з пожежної безпеки - навчання і перевірку знань відповідно до вимог Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 29.09.2003 N 368, зареєстрованого в Міністерстві юстиції України 11.12.2003 за N 1148/8469 [3].

4.2 Аналіз стану умов праці

Робота над створенням об’єктної моделі забезпечення оцінки кібербезпеки, розрахунок уразливості системи і визначення зв'язків атак і захистів проходитиме в побутовому приміщенні. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 - Розміри приміщення

Найменування	**Значення**
Довжина, м	3
Ширина, м	3
Висота, м	2,5
Площа, $м^2$	9
Об'єм, $м^3$	22,5

Згідно з [4] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Для зручності спільної роботи з іншими працівниками (обговорення ідей, з'ясування проблем і т.д.) в кімнаті є дивани і журнальний стіл, обставлені живими квітами. Також робочий процес пов'язаний з багатьма документами, теками, журналами для чого приміщення облаштоване принтером і шафою для зручності. Задля дотримання визначеного рівня мікроклімату в будівлі встановлено систему опалення та кондиціонування.

Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [5] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 ÷500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷800

4.2.3 Навантаження та напруженість процесу праці

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекоси у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м’язово скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи на ведені в [5].

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви:

- для операторів персональних комп’ютерів тривалістю 15 хв через дві години роботи;

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Роботу, пов'язану з ЕОМ з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання [6], які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої.

Основними робочими характеристиками персонального комп'ютера є:

– робоча напруга U=+220В +-5%;
– робочий струм I=2А;
– споживана потужність Р=350 Вт.

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [5].

За умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, забруднення повітря шкідливими речовинами, шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3).

Таблиця 4.3 - Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	К ількісна оцінка	Нормативні документи
1	2	3	4
Фізичні			
- підвищена температура поверхонь обладнання	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	ДСН 3.3.6.042-99 [4]
- підвищений рівень шуму на робочому місці	-//-	2	ДСН 3.3.6.042-99 [4]
- підвищена або знижена вологість повітря	-//-	2	ДСН 3.3.6.042-99 [4]
- підвищена або знижена рухливість повітря	-//-	1	ДСН 3.3.6.042-99 [4]
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	ГОСТ 12.1.030-81 [7] ГОСТ 13109-97 [8]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	ДБН В.2.5-28:2015 [9]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	ДБН В.2.5-28:2015 [9]
- підвищена яскравість світла	порушення умов праці (організації місця праці - налагодження моніторів)	1	ДСанПіН 3.3.2.007-98[5]
- понижена контрастність	-//-	1	ДСанПіН 3.3.2.007-98[5]
психофізіологічні:			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування;	4	НПАОП 0.00-1.28-10[6] ДСанПіН 3.3.2.007-98[5]

Продовження таблиці 4.3

- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці - сидіння користувача,) та організації робочого часу - безперервна робота)	2	НПАОП 0.00-1.28-10[6] ДСанПіН 3.3.2.007-98[5]

4.3.2 Пожежна безпека

Для гасіння пожеж в офісному приміщенні пропонується використовувати порошкові або вуглекислотні вогнегасники, так як вони є універсальними.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу.

В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Пожежна безпека при застосуванні ЕОМ забезпечується:

1) системою запобігання пожежі,
2) системою протипожежного захисту,
3) організаційно-технічними заходами.

Згідно [10] таке приміщення, площею 9 м^2, відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння. Відповідно до норм первинних засобів пожежогасінні пропонується використовувати:

- ручний вуглекислий вогнегасник ОУ-5 в кількості 1 шт.;
- повсть 1 1 м^2, кошму 2×1,5 м^2 або азбестове полотно 2×2 м^2 в кількості 1 шт.

Виникнення пожежі можливе, якщо на об'єкті є горючі речовини, окислювач і джерела запалювання. Вірогідність пожежної небезпеки приймається значною, якщо ймовірна взаємодія цих трьох чинників. Горючими компонентами є: будівельні матеріали для акустичної і естетичної обробки приміщень, перегородки, підлоги, двері, ізоляція силових, сигнальних кабелів і т.д.

4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [11] і наведені в табл. 4.4:

Таблиця 4.4 - Норми мікроклімату робочої зони об'єкту

Період року	**Категорія робіт**	**Температура C^0**	**Відносна вологість %**	**Швидкість руху повітря, м/с**
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

4.4.2 Освітлення

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

У проекті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше -1/8, в побутових – 1/10:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10}\right) \cdot S_n, \tag{4.1}$$

де S_b – площа віконних прорізів, м^2;

S_n – площа підлоги, м^2.

$S_n = a \cdot b = 3 \cdot 3 = 9$ м^2,

$S_{вік} = 1/10 \cdot 9 = 0{,}9$ м^2.

Приймаємо 2 вікна площею $S = 0{,}9$ м^2 кожне.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 5400 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \tag{4.2}$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, м^2; S = 9 м^2;

Z – поправочний коефіцієнт світильника (Z = 1,15 для ламп розжарювання та ДРЛ;

Z = 1,1 для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \times 9 \times 1.1 \times 1.5}{5400 \times 0.575 \times 2} = 0{,}7 \approx 1$$

Приймаємо освітлювальну установку, яка складається з 1 світильника, який складається з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.5 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення > 40 м^3 на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

4.6 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Загальний опір захисного заземлення визначається за формулою:

$$R_{ççï} = \frac{R_ç \cdot R_n}{R_n \cdot n \cdot \eta_ç + R_ç \cdot \eta_n} \quad (4.3)$$

де $R_з$ - опір заземлення, якими можуть бути труби, опори, кути і т.п., Ом;

R_n - опір опори, яка з'єднує заземлювачі, Ом;

n - кількість заземлювачів;

$\eta_з$ - коефіцієнт екранування заземлювача; приймається в межах 0,2 ÷ 0,9; $\eta_з = 0,7$

η_n - коефіцієнт екранування сполучної стійки; приймається в межах 0,1 ÷ 0,7; $\eta_n = 0,5$;

Опір заземлення визначається за формулою:

$$R_ç = \frac{\rho}{2\pi \cdot l}\left(\ln\frac{2 \cdot l}{d} + \frac{1}{2}\ln\frac{4 \cdot t + l}{4 \cdot t - l}\right) \quad (4.4)$$

де ρ - питомий опір ґрунту, залежить від типу грунту, Ом·м;

для піску - 400 ÷ 700 Ом·м; приймаємо ρ = 400 Ом·м;

l - довжина заземлювача, м; для труб - 2-3 м; l = 3 м;

d - діаметр заземлювача, м; для труб - 0,03-0,05 м; d = 0,05 м;

t - відстань від середини забитого в ґрунт заземлювача до рівня землі, м; t = 2 м.

$$R_з = \frac{400}{2 \cdot 3{,}14 \cdot 3}\left(\ln\frac{2 \cdot 3}{0{,}05} + \frac{1}{2}\ln\frac{4 \cdot 2 + 3}{4 \cdot 2 - 3}\right) = 110, \text{ Ом}$$

Опір смуги, що з'єднує заземлювачі, визначається за формулою:

$$R_n = \frac{\rho}{2\pi \cdot L} \cdot \ln\frac{2 \cdot L^2}{b \cdot t_l} \quad (4.5)$$

де L - довжина смуги, що з'єднує заземлювачі (м) і приблизно дорівнює периметру будівлі: $П_{буд.} = 42 \cdot 2 + 38 \cdot 2 = 160$ м; L = 160 м;

b - ширина смуги, м; b = 0,03 м;

t_1 - глибина заземлення від рівня землі, м; $t_1 = 0,5$ м.

$$R_n = \frac{400}{2 \cdot 3{,}14 \cdot 160} \cdot \ln\frac{2 \cdot 160^2}{0{,}03 \cdot 0{,}5} = 5{,}99, \text{ Ом}$$

Кількість заземлювачів захисного заземлення визначається за формулою:

$$n = \frac{2 \cdot R_ç}{4 \cdot \eta_ç} = \frac{2 \cdot 110}{4 \cdot 0{,}7} = 79 \text{ øò} \quad (4.6)$$

де 4 - допустимий загальний опір, Ом;

2 - коефіцієнт сезонності.

Визначаємо загальний опір захисного заземлення:

$$R_{ззп} = \frac{110 \cdot 5{,}99}{5{,}99 \cdot 79 \cdot 0{,}7 + 110 \cdot 0{,}5} = 1{,}7 \text{ , Ом}$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{ззп} < 4$ Ом.

4.7 Охорона навколишнього природного середовища

4.7.1 Загальні дані з охорони навколишнього природного середовища

Діяльність за темою магістерської роботи, а саме: Методи забезпечення кібербезпеки систем релейного захисту та автоматики в процесі її виконання впливає на навколишнє природнє середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища», Законом України «Про забезпечення санітарного та епідемічного благополуччя населення», Законом України «Про відходи», Законом України «Про охорону атмосферного повітря», Законом України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру», Водний кодекс України.

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

В процесі діяльності розробника об’єктної моделі за допомогою ПК виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

Відпрацьовані люмінесцентні лампи - І клас небезпеки

Акумулятор для джерел безперебійного харчування -III клас небезпеки

Змінні носії інформації - IV клас небезпеки

Макулатура - IV клас небезпеки

Побутові відходи - IV клас небезпеки

4.7.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі

Наводяться вимоги зберігання виявлених за своєю роботою відходів відповідно до вимог [12].

Відходи в міру їх накопичення збирають у тару, відповідну класу небезпеки, з дотриманням правил безпеки, після чого доставляють до місця тимчасового зберігання відходів відповідно до затвердженої схеми їх розміщення.

Не допускається зберігання відходів у невстановлених схемою місцях, а також перевищення норм тимчасового зберігання відходів.

Способи тимчасового зберігання відходів визначаються видом, агрегатним станом і класом небезпеки відходів:

- Відходи I класу небезпеки зберігаються в герметичній тарі (сталеві бочки, контейнери). У міру наповнення тару з відходами закривають герметично сталевий кришкою;

- Відходи II класу небезпеки в залежності від агрегатного стану зберігаються в поліети-ленових мішках, бочках, сховищах та інших видах тари, яка запобігає поширенню шкідливих речовин;

- Відходи III класу небезпеки зберігаються в тарі, яка забезпечує локалізацію зберігання, дозволяє виконувати вантажно-розвантажувальні і транспортні роботи і виключає поширення в ОС шкідливих речовин;

- Відходи IV класу небезпеки можуть зберігатися відкрито на промисловому майданчику у вигляді конусоподібної купи, звідки їх автонавантажувачем перевантажують у самоскид і до-ставляють на місце утилізації або захоронення;

4.7.3 Визначення впливу та заходів щодо поводження з відходами ІТ галузі

З метою визначення та прогнозування впливу відходів на навколишне середовище, своєчасного виявлення негативних наслідків, їх запобігання відповідно до Закону України «Про відходи» повинен здійснюватися моніторинг місць утворення, зберігання, і

видалення відходів. Відомості про місце утворення та місце розташування відходів зазначаються та наводяться у таблиці 4.5.

Таблиця 4.5 - Відомості про місце утворення та місце розташування відходів

з/п	Код та найменування відходів за ДК -005-96	Технологічний процес або виробництво, де утворюються відходи / клас небезпеки	Місце розташування відходу, тара та її кількість, місткість, розміри у разі наявності майданчиків розташування відходів необхідно зазначити тип покриття та наявність даху)
	2	3	4
	7710.3.1.26 Лампи люмінесцентні, та відходи, які містять ртуть, інші зіпсовані або відпрацьовані (Відпрацьовані ртутьвмісні люмінесцентні лампи)	1	буд.78, кв. 63
	7710.3.1.01 Макулатура паперова та картонна (Макулатура)		буд.78, кв. 63
	Акумулятор для джерел безперебійного живлення	3	буд.78, кв. 63

Висновки до розділу 4

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в дипломній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

А також визначені основні екологічні аспекти впливу на навколишнє природнє середовище та зазначені заходи щодо поводження з ними.

Перелік посилань до розділу 4

1. НПАОП 0.00-6.03-93 «Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві»

2. НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці»

3. НАПБ Б.02.005-2003 «Типове положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України»

4. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»

5. ДСанПіН 3.3.2.007-98 «Правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»

6. НПАОП 0.00.-1.28-10 «Правил охорони праці під час експлуатації електронно-обчислювальних машин»

7. ГОСТ 12.1.030-81 «ССБТ. Електробезпечність .Захисне заземлення. Занулення»

8. ГОСТ 13109-97 «Електрична енергія. Сумісність технічних засобів вієлектромагнітних. Норми якості електроенергопостачання загального призначення »

9. ДБН В.2.5-28:2015 «Природнє і штучне освітлення»

10. НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою»

11. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»

12. ДСанПіН 2.2.7.029 «Гігієнічні вимоги щодо поводження з промисловими відходами та визначення їх класу небезпеки для здоров'я населення».

ВИСНОВКИ

В результаті роботи зробили аналіз основних алгоритмів для побудування рекомендаційної системи. Було розглянуто усі можливості, переваги та недоліки кожного з алгоритму та реалізовано аналіз підходів. Також зроблено огляд помічних засобів для кореляції результатів, отриманих після результатів роботи основних алгоритмів.

Мета рекомендаційної системи щоб проінформувати користувача про товар, який йому може бути найбільш цікавий в даний момент часу. Клієнт отримує інформацію, а сервіс заробляє на наданні якісних послуг. У роботі розглянули алгоритм колаборативной фільтрації і content based, та далі розробили рекомендаційну систему по схожості користувачів та товарів, яка визначається з використанням косинусної схожості по оцінкам продукції.

Розглянули проблему холодного старту, бо це типова ситуація, коли ще не накопичено достатню кількість даних для коректної роботи рекомендаційної системи (наприклад, коли товар новий або просто його дуже рідко купують).

За допомогою моделі аналізу поведінки споживачів на ринку ми змогли зрозуміти, що саме йому належить дізнатися про поведінку споживачів, які кроки для цього зробити.

Отже, у даній роботі ми змогли розробити рекомендаційну систему суспільно з веб-аналітикою, де у міру взаємодії користувача з системою, векторні опису придбаних ним товарів об'єднуються (підсумовуються і нормалізуються) в єдиний вектор і, таким чином, формується вектор його інтересів для відображення найкращих пропозицій у інтрнет магазині.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Analytical tools Google Analytics [Електронний ресурс]. Режим доступа: http://www.google.by/intl/ru_ALL/analytics/features/analysis–tools.html.
2. Analytics online store for 5 min: Как держать продажи под. контролем [Електронний ресурс]. Режим доступа: https://www.ecwid.ru/blog/customize-dashboard-google-analytics-and-yandex-metrika.html
3. Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pages 253–260. ACM, 2002.
4. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, pages 285–295. ACM, 2001.
5. Bal´azs Hidasi and Domonkos Tikk. Context-aware item-to-item recommendation within the factorization framework. In Proceedings of the 3rd Workshop on Context-awareness in Retrieval and Recommendation, pages 19–25. ACM, 2013.
6. Bal´azs Hidasi and Domonkos Tikk. Fast als-based tensor factorization for context-aware recommendation from implicit feedback. In Machine Learning and Knowledge Discovery in Databases, pages 67–82. Springer, 2012.
7. Chengjie Sun, Lei Lin, Yuan Chen, and Bingquan Liu. Expanding user features with social relationships in social recommender systems. In Natural Language Processing and Chinese Computing, pages 247–254. Springer, 2013. 33
8. Christian Desrosiers and George Karypis. A comprehensive survey of neighborhoodbased recommendation methods. In Recommender systems handbook, pages 107–144. Springer, 2011.
9. Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. In SDM, volume 5, pages 1–5. SIAM, 2005.
10. Emmanuel J Cand`es and Benjamin Recht. Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6):717–772, 2009.
11. Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In Recommender systems handbook, pages 217–253. Springer, 2011.
12. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6):734–749, 2005.

13. Gregory Piatetsky. Interview with simon funk. ACM SIGKDD Explorations Newsletter, 9(1):38–40, 2007. 32
14. Harald Steck. Training and testing of recommender systems on data missing not at random. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 713–722. ACM, 2010.
15. Istv´an Pil´aszy and Domonkos Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In Proceedings of the third ACM conference on Recommender systems, pages 93–100. ACM, 2009.
16. Istv´an Pil´aszy, D´avid Zibriczky, and Domonkos Tikk. Fast als-based matrix factorization for explicit and implicit feedback datasets. In Proceedings of the fourth ACM conference on Recommender systems, pages 71–78. ACM, 2010.
17. Iv´an Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In Proceedings of the 5th ACM conference on Recommender systems, RecSys 2011, New York, NY, USA, 2011. ACM.
18. James Bennett and Stan Lanning. The netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35, 2007.
19. Jun Wang, Arjen P De Vries, and Marcel JT Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 501–508. ACM, 2006.
20. Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday, and Christos Efstratiou. Developing a context-aware electronic tourist guide: some issues and experiences. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 17–24. ACM, 2000.
21. Liangjie Hong, Aziz S Doumith, and Brian D Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In Proceedings of the sixth ACM international conference on Web search and data mining, pages 557–566. ACM, 2013.
22. Nathan Srebro, Tommi Jaakkola, et al. Weighted low-rank approximations. In ICML, volume 3, pages 720–727, 2003.
23. Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In Proceedings of the third ACM international conference on Web search and data mining, pages 81–90. ACM, 2010.

24. Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In Proceedings of the TwentyFifth Conference on Uncertainty in Artificial Intelligence, pages 452–461. AUAI Press, 2009.
25. Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In Proceedings of the 34th ACM SIGIR Conference on Reasearch and Development in Information Retrieval. ACM, 2011.
26. Steffen Rendle. Factorization machines with libfm. ACM Transactions on Intelligent Systems and Technology (TIST), 3(3):57, 2012.
27. Steffen Rendle. Factorization machines. In Proceedings of the 10th IEEE International Conference on Data Mining. IEEE Computer Society, 2010.
28. Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. SIAM review, 51(3):455–500, 2009.
29. Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. Svdfeature: a toolkit for feature-based collaborative filtering. The Journal of Machine Learning Research, 13(1):3619–3622, 2012.
30. Tianqi Chen, Zhao Zheng, Qiuxia Lu, Weinan Zhang, and Yong Yu. Feature-based matrix factorization. arXiv preprint arXiv:1109.2271, 2011. 31
31. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.
32. Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434. ACM, 2008.
33. Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on, pages 263–272. IEEE, 2008.
34. Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In Algorithmic Aspects in Information and Management, pages 337–348. Springer, 2008.
35. Zeno Gantner, Steffen Rendle, and Lars Schmidt-Thieme. Factorization models for context-/time-aware movie recommendations. In Proceedings of the Workshop on Context-Aware Movie Recommendation, pages 14–19. ACM, 2010.
36. К. В. Воронцов. Математические методы обучения по прецедентам (теория обучения машин). Москва, 2011.

ДОДАТОК А. СЛАЙДИ ДО ПРЕЗЕНТАЦІЇ

1

ДОСЛІДЖЕННЯ І РОЗРОБКА СЕГМЕНТІВ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ НА БАЗІ ВЕБ-АНАЛІТИКИ

Канакін А.Е., КН-17дм

Об'єкт

процес підбору рекомендаційних характеристик для інтернет магазину.

Предмет

алгоритми та рекомендації щодо покращення результатів роботи.

Мета

вивчення методів та алгоритмів рекомендаційної системи та впровадження їх в існуючий інтернет проект.

2

Завдання

3

- Зробити аналіз усіх існуючих алгоритмів рекомендаційної системи.
- Провести серію дослідів, в ході яких виявити найкращий варіант системи для інтернет магазину, а також проаналізувати вхідні дані які маємо для правильної роботи алгоритмів.
- Обробити отримані результати, підвести підсумки, розкрити найбільш ефективні алгоритми та виявити найменш ефективні.
- Виконати необхідні введення в існуючу систему на основі зроблених досліджень в цій роботі.

Існуючи рекомендаційні системи

4

- Amazon
- Glovo
- Netflix
- Ebay
- Apple Music

Існуючи засоби веб-аналітики

5

- Google Analytics
- Yandex.Metrika
- WebTrends
- Webalizer
- AWStats

Шляхи вирішення задач

6

Основні підходи до побудови рекомендаційних систем:

- гібридний підхід;
- на підставі описів (content-based);
- колаборативна фільтрація (collaborative filtering);

Шляхи вирішення задач

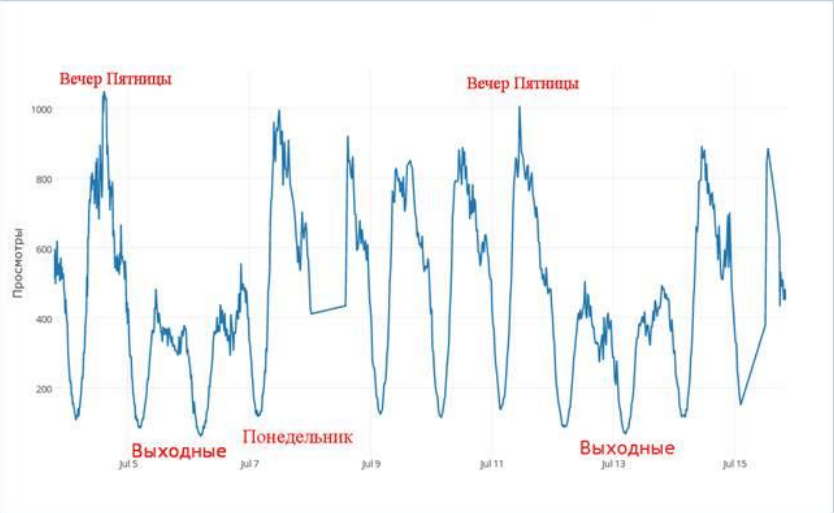
9

Інтеграція рекомендаційної системи з аналітикою

- Отримання даних з веб-аналітики
- Вибір найкращого методу рекомендації для задачі
- Рекомендаційна система та веб-аналітика разом

Результати

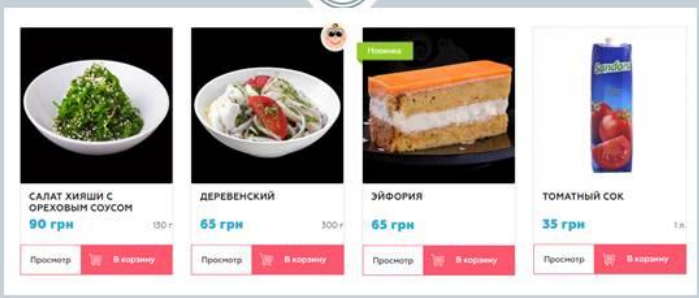
10



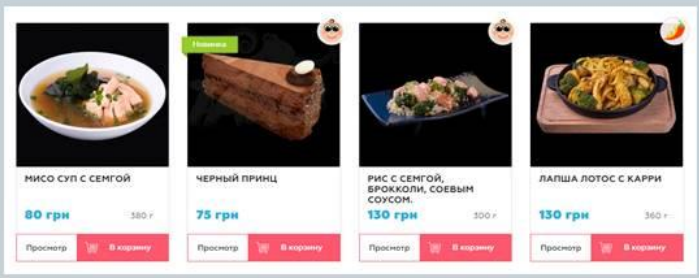
Відвідуваність по годинах

Результати

11



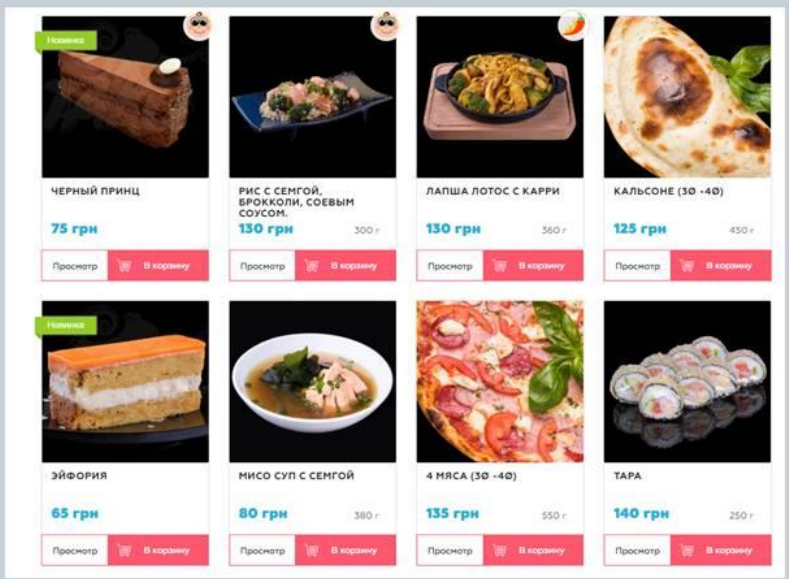
Рекомендації в першій половині робочого дня



Рекомендація в обідню перерву робочого дня

Результати

12



«Агресивна» рекомендація

Висновки

13

- Проаналізовано основні алгоритми для побудування рекомендаційної системи.
- Протестовані методи рекомендацій на власних завданнях
- Вирішені проблеми холодного старту
- Налагоджений зворотній зв’язок з користувачем
- Проаналізовано поведінку споживачів