Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*

(повна назва)

Кафедра *Системотехніки*

(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)

(освітньо-кваліфікаційний рівень)

ГЮІК.506120.012 ПЗ

(позначення документа)

Розробка та дослідження компонентів інформаційно-аналітичної системи електронних продажів

(тема роботи)

Виконав: студент II курсу, групи *ІТПм-19-1*

спеціальності 122 - *«Комп'ютерні науки та інформаційні технології»*

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма *Інформаційні технології проектування*

(повна назва освітньої програми)

Дудко В.Ю.

(прізвище, ініціали)

Керівник *проф. Нечипоренко А.С.*

(прізвище, ініціали)

Допускається до захисту

Зав. кафедри СТ _____ *проф. Гребеннік І.В.*

(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки
(назва вищого навчального закладу)

Факультет *Комп'ютерних наук*
(повна назва)

Кафедра *Системотехніки*
(повна назва)

Рівень вищої освіти *другий (магістерський)*

Спеціальність *122 «Комп'ютерні науки та інформаційні технології»*
(код і повна назва)

Тип програми *освітньо-професійна*
(освітньо-професійна або освітньо-наукова)

Освітня програма *Інформаційні технології проектування*
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри СТ _____
(підпис)

" _____ " . _____ 2020 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Дудко Владиславу Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка та дослідження компонентів інформаційно-аналітичної системи електронних продажів
затверджена наказом по університету від « *02* »11 20*20* р. № 1517 Ст.
2. Термін подання студентом роботи *12 грудня 2020 р.*
3. Вихідні дані до роботи *Розробити компонентиінформаційно-аналітичної системиелектронних продажівіз використанням методів транскодуваннямедіафайлів та прогнозу рекомендацій.*
Система повинна являти собою проект клієнтського web-додатка з інтерфейсом доступу до бази даних. Перелік використовуваних програмних засобів: ОС Windows 10, Docker,Kafka, Intiji Idea 2020
Технічне забезпечення: IBM-сумісний ПК з МП Core 2 Duo та вище.
4.Зміст пояснювальної записки (перелік питань, що потрібно розробити*)* *4.1 Вступ. 4.2 Оглядкомпонентів інформаційно-аналітичних систем. 4.3 Методи рекомендацій та транскодування. 4.4 Постановка задачі. 4.5 Дослідження методів та алгоритмів вирішення задачі. 4.6 Огляд існуючих системелектронних продажів 4.7 Проектування системи. 4.8 Практична реалізація методу транскодування та прогнозування рекомендацій. 4.9 Висновки*

5.Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, плакатів). *5.1 Контекстна діаграма IDEF0 інформаційної системи (1 аркуш формату А4). 5.2 IDEF0-діаграма декомпозиції (1 аркуш формату А4). 5.3 Діаграма варіантів використання. (1 аркуш формату А4). 5.5 Модель бази даних системи (1 аркуш формату А4).*

6. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		(підпис)	(дата)
Аналіз предметної області	проф. Нечипоренко А.С.		
Дослідження методів та алгоритмів вирішення задачі	проф. Нечипоренко А.С.		

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів атестаційної роботи	Термін виконання етапів роботи	Примітка
1.	Отримання завдання атестаційної роботи	02.11.2020	
2.	Аналіз завдання, літератури та аналогів з теми атестаційної роботи	02.11-12.11.2020	
3.	Формування постановки задачі	13.11 - 15.11.2020	
4.	Дослідження методів та алгоритмів вирішення задачі	16.04 - 20.11.2020	
5.	Розробка методів рекомендацій та транскодування	21.11 - 25.11.2020	
6.	Проектування та розробка інформаційно-аналітичної системи	26.11 - 03.12.2020	
7.	Аналіз результатів, отриманих за допомогою програмного засобу	04.12 - 05.12.2020	
8.	Оформлення пояснювальної записки та програмної документації	06.12 - 09.12.2020	
9.	Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту	10.12 - 12.06.2020	
10.	Представлення на рецензування	12.12.2020	
11.	Подання атестаційної роботи до ЕК	13.12.2020	

Дата видачі завдання *02 листопада 2020* р.

Студент _____ *Дудко В.Ю.*
(підпис)

Керівник роботи _____ *проф. Нечипоренко А.С.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до атестаційної роботи містить 75 аркушів, 23 рисунків, 1 таблицю, 3 додатків, 19 джерел. Графічна частина атестаційної роботи містить 6 плакатів.

ЕЛЕКТРОННА КОМЕРЦІЯ, ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА, АНАЛІТИЧНИЙ БЛОК, ТРАНСКОДИНГ, CINEMATCH, БАЗА ДАНИХ, СИСТЕМА УПРАВЛІННЯ БАЗОЮ ДАНИХ, WEB-СЕРВІС, РЕКОМЕНДАЦІЙНІ СИСТЕМИ,

Об’єктом дослідження є – процес транскодування медіафалів та процес прогнозу рекомендацій в рекомендаційних системах.

Предметом дослідження є метод транскодування, котрий застосовується для процесу перетворення медіафайлу з одного формату в інший, із можливістю його подальшого застосування на різних платформах і пристроях, та методи, засоби, і алгоритми прогнозування рекомендацій в рекомендаційних системах.

Метою магістерськьої атестаційної роботи є розробка та дослідження компонентів інформаційно-аналітичної системи електронних продажів

Методи дослідження – системний підхід, методи структурного аналізу і моделювання

Результати роботи – розроблені методи транскодування медіафайлівтапрогнозурекомендацийв рекомендаційних системаханалітичного блоку інформаційно-аналітичної системи, а також веб-сервіс відеохостингу з вбудованим рекомендаційним методом.

ABSTRACT

Explanatory note contains 75 sheets, 25 figures, 1 table, 3 annexes, 19 sources; graphic material – 6 p.

ELECTRONIC COMMERCE, INFORMATION AND ANALYTICAL SYSTEM, ANALYTICAL BLOCK, TRANSCODING, CINEMATCH, DATABASE, DATABASE MANAGEMENT SYSTEM, WEB SERVICE, RECOMMENDATION SYSTEMS.

The object of research is the process of transcoding media and the process of forecasting recommendations in recommendation systems.

The subject of the research is the transcoding method, which is used for the process of converting a media file from one format to another, with the possibility of its further application on different platforms and devices, and methods, tools and algorithms for predicting recommendations in recommendation systems.

The purpose of the master's certification work is to develop and study the components of the information-analytical system of electronic sales

Research methods - a systematic approach, methods of structural analysis and modeling

The results of the work - developed methods of the analytical unit of the information-analytical system, as well as a web service of video hosting with a built-in recommendation method.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 3
ВСТУП 4
1 ОГЛЯД ТА АНАЛІЗ ХАРАКТЕРИСТИК СУЧАСНИХ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ 5
1.1 Стримінгові сервіси 7
1.2 Компоненти інформаційно-аналітичних систем електронних продажів 7
1.3 Методи рекомендацій та транскодування 9
1.4 Недоліки існуючих методів рекомендацій та транскодування 11
1.5 Класифікація рекомендаційних систем 13
1.6 Огляд існуючих інформаційно-аналітичних систем 19
1.7 Дослідження методів та алгоритмів прогнозування рекомендацій 23
1.8 Постановка задачі 25
2 РОЗРОБКА КОМПОНЕНТІВ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ЕЛЕКТРОННИХ ПРОДАЖІВ ..... **Ошибка! Закладка не определена.**
2.1 Розробка системних вимог до інформативно-аналітичної системи ..... **Ошибка! Закладка не определена.**
2.2 Визначення функціональних вимог до інформаційно-аналітичної системи 28
2.3 Розробка моделі потоків даних компонентів інформаційно аналітичної системи 33
2.4 Розробка вимог до інтерфейсу web-сторінок клієнтської частини системи 37
2.5 Діаграма варіантів використання системи для визначення вимог до її інтерфейсу 37
2.6 Діаграми послідовності дій системи 40
3 ВДОСКОНАЛЕННЯ МЕТОДІВ ТРАНСКОДУВАННЯ МЕДІАФАЙЛІВ ТА ПРОГНОЗУВАННЯ РЕКОМЕНДАЦІЙ 42
3.1 Метод транскодування медіафайлів 42
3.2 Метод прогнозування рекомендацій 49
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДУ ТРАНСКОДУВАННЯ ТА ПРОГНОЗУВАННЯ РЕКОМЕНДАЦІЙ 51
4.1 Обґрунтування вибору мови та технології програмування 51
4.2 Обгрунтування вибору методів рекомендаційної системи 55
4.3 Обґрунтування вибору концепції аналітичної бази даних 58
4.4 Логічне та фізичне моделювання даних системи 64
4.3 Програмна реалізація функціоналу 69
ВИСНОВКИ 72
ПЕРЕЛІК ПОСИЛАНЬ 73
ДОДАТОК А Графічні маетеріали 81

ДОДАТОК Б Текст програми 94

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CRM – customer relationship management;

TS – Transcoding system

ІС – інформаційна система;

АІС – автоматизована інформаційна система;

БД – база даних;

СУБД – система управління базами даних;

IS – Information system;

SQL – Structured Query Language;

URL – Uniform Resource Locator;

UML – Unified Modeling Language;

RS – рекомендаційна система;

Collaborative filtering – колаборативна або спільна фільтрація;

Content-based filtering – фільтрація на основі вмісту або контентна фільтрація;

ВСТУП

Початок XXI сторіччя відзначився розвитком інформаційних технологій. Упроваджуються новітні технології, що кардинально змінює всі аспекти повсякденного життя.

Зараз користувачі можуть використовувати мережу Інтернет для задоволення всіх власних потреб. Використанням мережі Інтернет, сприяло перенесенню багатьох традиційних форм діяльності в мережевий простір. Найбільш успішним видом діяльності, який отримав широке поширення в мережі Інтернет, є електронна комерція.

Під терміном «електронна комерція» прийнято вважати, надання товарів і платних послуг через мережу Інтернет, що природно відкриває широкі можливості як для вже існуючих, так і для нових компаній. Для більш ефективної роботи e-commerce систем розроблена концепція CRM (Customer Relationship Management), яка використовується для автоматизованого управління відносинами з клієнтами. Програмні системи, що розроблені за цією концепцією, отримали назву «CRM-системи». Система CRM збирає всю актуальні дані про клієнтів. CRM-система аналізує всю корисну інформацію, що зберігається в базі даних CRM-системи. В атестаційній роботі розглядається рішення актуальної задачі розробки компонентів інформаційно-аналітичної системи.

Метою магістерськьої атестаційної роботи є розробка та дослідження компонентів інформаційно-аналітичної системи електронних продажів

Відповідно до мети роботи необхідно виконати такі завдання дослідження:

– аналіз існуючих інформаційно-аналітичних систем, в котрих впроваджені модулі аналізу та транскодування інформації;

– вдосконалення методів транскодуваннямедіафайлівта прогнозування рекомендацій;розробка стримінгового сервісу;
– реалізація та впровадження розроблених компонентів інформаційно-аналітичних систем до створеного стрімінгового сервісу.
– дослідження розроблених методів і алгоритмів на тестових даних

Об’єктом дослідження є – процеси транскодування медіафалів та прогнозу рекомендацій в рекомендаційних системах.

Предметом дослідження є метод транскодування, котрий застосовується для процесу перетворення медіафайлу з одного формату в інший, із можливістю його подальшого застосування на різних платформах і пристроях, та методи, засоби, і алгоритми прогнозування рекомендацій в рекомендаційних системах.

1 ОГЛЯД ТА АНАЛІЗ ХАРАКТЕРИСТИК СУЧАСНИХ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ

Протягом останніх десятиліть, із зростанням та розвитком технологій, зокрема Інтернету, наше оточення та методи взаємодії зі світом здебільшого все більш «цифрові». В інтернет ми кожен день робимо покупки, говоримо, дивимся новини, слухаємо музику, сплачуємо рахунки. Із цього виходить, що велікий обсяг інформації та даних є невід'ємною частиною нашого повсякденного життя. Сучасні користувачі стикаються з проблемами пошуку та вибору корисної для себе інформації. Іноді вони приймають рішення, не усвідомлюючи про існування інших, можливо кращих, альтернатив. Тому сьогодні велика кількість людей передає процес прийняття рішень веб-сервісам.

Через те, що немає обізнаності та постійне збільшення обсягу доступної інформації, яку надають сучасні технології, дуже велике значення набуває наявність методів рекомендації, вони спрощують процес прийняття рішень.

Останнім часом стає все більш популярним створення систем, які здатні вгадувати переваги і потреби користувачів і на основі цього пропонувати відповідні рішення. Такі системи, звані рекомендаційними, мають безліч застосувань. Що важливо, рекомендації формуються для конкретного користувача, виходячи з його особистих даних і інтересів.

Такий підхід значно спрощує користувачеві роботу з великим об'ємом даних, допомагає зменшити час пошуку потрібного рішення, забезпечуючи релевантність інформації. Однак найчастіше рекомендаційні системи розробляються для вирішення

задач конкретної області. Наприклад, Amazon спеціалізується на товарах, Netflix на фільмах.

Основні переваги рекомендаційних систем в веб сервісах:

Розуміння потреб користувача. Дані про інтереси користувача, які можуть збиратись та оброблятись явно системою, або прогнозуватись різними алгоритмами, наприклад алгоритм холодного старту. .

Збільшення задоволення користувача системи. Рекомендаційна система яка добре спроектована може покращити реалізацію потреб користувача та загальне враження від сервісу. Рекомендації в комбінації із хорошим юзер інтерфейсом будуть добре сприйняті користувачем системи . В свою чергу це призведе до збільшення та частоти використання сервісу. Рекомендаційна система підкаже абоненту цікаві фільми, навіть якщо вони вже давно вийшли з розряду новинок: таким чином великий відеокаталог перестає бути курній бібліотекою, а стає інтерактивною вітриною, гнучко підлаштовується під смаки і настрої абонентів.

Збільшення прибутку. Для комерційних проектів головною метою є можливість збільшити обсяг продажів. Цього можна досягнути за рахунок надання користувачам рекомендованих товарів. Є ймовірність, що получені рекомендації ана основі попередніх транзакцій будуть відповідати потребам користувача.

В даній роботі, в якості предметної області для дослідження компонентів інформаційно-аналітичних систем розглянемо стримінгові веб сервіси.

1.1 Стримінгові сервіси

Потокове передавання відео або точніше передавання відео - це спосіб передачі зображення і звуку, включаючи відео, не вимагаючи завантаження файлів з Інтернету. Стримінгові сервіси, такі як Netflix, Megogo та Prime Video, використовують цей метод для надання відео, які можна дивитись на різних типах пристроїв

Головний секрет технології потокового мультимедіа полягає в буферизації програються даних. Встановлений на настільному комп'ютері програмний медіаплеєр здійснює з'єднання з сервером і запитує потік. Сервер починає передавати медіапотоків, адресуючи його плеєру. При такій буферизації короткочасні затримки в потокової передачі, викликані перевантаженнями в мережі, не зроблять помітного впливу на якість програвання мультимедійної інформації. І чим більше буфер, тим менше вплив мережевих збоїв на якість передачі.

Потокові сервери здатні забезпечувати безперервний доступ до вже готовим файлів мультимедіа. Такий режим класифікується як надання аудіо- або відео за запитом. Інформація про стан активів даний момент події може бути передана і безпосередньо, за допомогою підключених до комп'ютера мікрофону або відеокамери, і потім у вигляді медіапотоку транслюватися на задану аудиторію. Цей режим передачі потоку називається Web-мовленням або технологією Webcast.

1.2

Інформаційно-аналітична система призначена для контролю і аналізу діяльності выдеохостингу, побудови звітів, а також моніторингу. Система забезпечує автоматичне поповнення сховища даних, безперервність і цілісність даних, алгоритмів їх обробки і оцінки.

Проблема створення автоматизованих систем аналізу та підтримки прийняття рішень особливо актуальна для багатьох подібних систем, що активно займаються управлінням і контролем діяльності.

Для ефективного вирішення цих завдань необхідна багатофункціональна інформаційна система, що покриває потреби як менеджерів вищої ланки, керівників, так і груп аналітиків, що займаються детальним аналізом і прогнозуванням.

Для збору і зберігання цієї інформації додається багато зусиль і витрачається багато коштів, але при цьому велика її частина не може бути використана тими, кому вона найпотрібніша, відповідальними за прийняття рішень. Найчастіше ця інформація доступна тільки тим підрозділам, в яких вона збирається і обробляється. Всі ці фактори призводять до розробки та впровадження систем підтримки прийняття рішень.

Таким чином, інформаційно-аналітична система призначена для вирішення завдань комплексного динамічного аналізу і його структурних підрозділів; моделювання потоків даних; оцінки наслідків прийнятих рішень; формування варіантів стратегій.

На відміну від інших областей, в яких використовуються системи рекомендацій, такі як товари, білети або готелі, рекомендації в відео області мають певні специфічні характеристики, які слід враховувати при створенні відео рекомендаційної системи. Деякі з цих особливостей у галузі відео впливають на застосування рекомендаційних систем та методів машинного навчання.

Серед цих особливостей можна виділити:

Наприклад, часто одні й ті ж популярні телешоу йдуть по різних каналах. В цьому випадку рекомендаційної системі доводиться розуміти дублювання інформації і вибирати рекомендацію на основі переваг абонента щодо каналів, часу початку передачі. Подібне дублювання інформації також виникає при наявності у абонента підписки на SD і HD версії каналів.

Безперервно працює зворотний зв'язок від абонентів, які взаємодіють з рекомендаціями, наповнює базу прецедентів алгоритмів машинного навчання і дозволяє рекомендацій змінюватися в залежності від неявних ознак зміни уподобань абонентів, таких як пору року, наближення свят або зміна складу сім'ї.

Інше поле рекомендацій, де працює штучний інтелект - мініатюра і слоган до кожного відео. Це призводить до того, що різним глядачам до одних і тих серіалів і фільмів транслюються абсолютно різні зображення.

На мініатюру можуть виділити захоплюючу сцену з фільму, наприклад автомобільну погоню, відомого актора, якого ви любите, або драматичну сцену. Програма сама вирішить, що показати саме вам. Це повинно змусити користувача зацікавитися відео.

Не менш важливою системою є система транскодування відеофайлу. Перш ніж фільм стане доступним для користувачів, сервіс повинен перетворити відео в формат, який найкраще підходить для вашого пристрою. Цей процес називається транскодуванням або кодуванням. Відео поставляється у форматі високої чіткості, який займає багато пам’яті. Крім того, сервіс повинен підтримувати багато різних пристроїв. Кожен пристрій має формат відео, який найкраще виглядає саме на цьому пристрою. Сервіс також створює файли, оптимізовані для різної швидкості мережі. Якщо ви дивитесь у швидкій мережі, ви побачите відео вищої якості, ніж у повільній мережі. А також залежить від вашої плану підписки.

1.3 Методи рекомендацій та транскодування

Постановка задачі прогнозування персоналізованих оцінок методом колаборативної фільтрації полягає в наступному. Нехай U-множина користувачів, $|U| = m$, I-множина предметів, $|I| = n$, D-множина дій надання персоналізованих рекомендацій. Потрібно здійснити прогноз персоналізованих рекомендацій $U \times I \rightarrow D$. Нехай R-множина оцінок користувачів Up -множина профілів користувачів, Ip -

множина профілів предметів, $Upi \in Up$ -вектор профілю і-того користувача $U_{pi=((i_1,r(U_{p1},j_1),((i_2,r(U_{pi},j_2),((i_n,r(U_{pi},j_n),\ ,r(u_{pi},j_k)}$ - оцінка і-того користувача для j-того предмета, $I_{\ pi} \in I_p$ - вектор профілю j-того предмета $I_{pi} = ((u_1, r(u_1, i_j), ((u_2, r(u_2, i_j), \dots, ((u_m, r(u_m, i_j), r(u_i, r_{pj}) = r_{ij}$- оцінка j-того предмета для і-того користувача $r(u_{pi}, j_k) = r(u_i, i_{pi})$.

Необхідно здійснити прогноз рекомендацій $\hat{r}_{\ ij} = Predict(u, i, \dots) \approx r_{ij}$, де $\hat{r}_{\ ij}$- прогнозоване значення оцінки, r_{ij} - оцінка, яка виставлена користувачем. Завдання системи полягає в тому, щоб досягнути мінімальної різниці між $\hat{r}_{\ ij}$і r_{ij}.

$$\hat{r}_{\ ij} - r_{ij} \Rightarrow min \tag{1.1}$$

В загальному функціонування рекомендаційної системи може бути описане наступним відображенням

$$RS_a: U_p * I_p \rightarrow \hat{R} \tag{1.2}$$

де $\hat{R}$ - множина прогнозованих значень оцінок предметів за якими можна визначити множину рекомендованих предметів. Цільова функція прогнозування рекомендацій для рекомендаційної системи представлена наступним рівнянням

$$F_{rs} = \min(\sum_{i=1}^{m} \sum_{j=1}^{n} \hat{r}_{\ ij} - r_{ij} \tag{1.3}$$

Рекомендаційна системи повинна забезпечити прогнозування дійсних уподобань користувачів. Узагальнена схема функціонування методу колаборативної фільтрації представлена на рисунку 1.6.

Рисунок 1.6 – Типова схема функціонування методу фільтрації

1.4 Недоліки існуючих методів рекомендацій та транскодування

Коли в системі вже багато користувачіві, з'являється проблема масштабованості. Наприклад, якщо буде 5 мільйонів користувачів і мільйон товарів та послуг, алгоритм рекомендацій стає дуже складний для розрахунків та оптимізації. Більщість систем треба моментально реагувати на запити від кожного користувача, незалежно від історії їх покупок і оцінок. Це потребує ще більшої масштабованості.

Наявність проблеми масштабованості вимагає від алгоритмів рекомендаційних систем можливість необмеженого збільшення числа, як користувачів, так і числа варіантів можливих рекомендацій. При цьому збільшення не повинно вимагати заміни в коді програми реалізації алгоритмів і повторної обробки всіх вже оброблених даних і допускати високу паралельність обчислень. Алгоритми повинні добре працювати з ростом розмірів цих множин до дуже великих значень ($10^5 - 10^8$) і не вимагати модернізації досить довгий час. Алгоритм повинен забезпечувати формування значень функції релевантності на основі емпіричного оцінювання. Побудова алгоритмів самоорганізації множин об'єктів і суб'єктів переваг в ході еволюції функції релевантності та подальше її ефективне використання на множині великих розмірів.

Проблема «холодного старту» полягає в тому, що при першому вході в рекомендаційну систему для нового суб'єкта невідомі значення рейтингів або функції релевантності для практично всіх об'єктів переваг. Те ж саме відноситься і до появи нового об'єкта. Для нього функція переваги також невідома для більшості суб'єктів. Алгоритм повинен забезпечувати формування значень функції релевантності для нового суб'єкта и нового об’єкту.

Холодний старт для користувачів можливий на основі демографічних даних, які самі користувачі вказують наприклад при регістрації.

Існують два основні підходи до застосування в рекомендаціях демографічної інформації про користувача:

- Експертним чином складаються стереотипи для різних демографічних категорій. Тобто експерт сам визначає що на холодному старті показувати кожної з категорій. Мінусом такого підходу є необхідність роботи експерта, при цьому користувачу будуть рекомендуватися тільки популярні сайти суб'єктивно підібрані експертом.

- Демографічні категорії визначаються автоматично шляхом виявлення кластерів користувачів зі схожими інтересами. Рекомендації будуються на основі того, які рейтинги проставляли користувачі з тієї ж категорії, тобто того ж віку, статі, місця розташування.

Для вирішення проблеми холодного старту для нових веб-сторінок застосовуються різні методи аналізу тексту та іншого контенту.

Основні методи семантичного аналізу тексту - це LDA і relevance feedback.

Загальна схема рекомендацій на основі текстового контенту сторінки полягає в тому, що спочатку по всіх сайтах збирається корисний контент. Слова в тексті проходять попередню обробку, тобто відкидаються стоп-слова і проводиться лематизація. Далі складається єдиний словник слів. Слова зважуються по $TF - IDF$ і для надто довгих текстів залишається тільки $top\ N$ найвагоміших слів.

В основу алгоритмів покладено механізми пошукової оптимізації. В процесі холодного старту користувач фізично здійснює переміщення по безлічі об'єктів і по ходу процесу збирається інформація про значення функції релевантності за всіма пройденими об'єктами. Швидкість збору інформації по максимальному числу об'єктів залежить від можливих траєкторій переміщення по множині, а значить від властивостей цієї множини. Далі, спільно обробляючи функції релевантності для всіх множин користувачів і для усіх об'єктів, можна знаходити точки максимальної релевантності для будь-якого користувача. Пошук точок максимальної релевантності здійснюється також пошукової процедурою, однак в цьому випадку не фізично користувачем, а алгоритмическим шляхом. Швидкість такого пошуку визначається

обчислювальною складністю алгоритму пошуку і залежить від властивостей простору пошуку.

Успіх і ефективність пошукових процедур на дискретних множинах залежить від допустимих переміщень від одного елемента множини до іншого. Такі властивості можна описати, додав на дискретній множині деяку структуру, тобто перетворивши просту дискретну множину в деякий простір. Безліч об'єктів після наділення його структурою можно називати простором пошуку або переваг. Множину користувачів - простором активностей. Простір переваг має мати властивість максимально ефективно вирішувати завдання переміщення від одного елемента до іншого для досягнення цілі: за найменше число кроків відвідати будь-який з елементів простору переваг на етапі холодного старту, і за найменше число кроків знайти максимально релевантний елемент, при старті від довільного елемента. На наступній формулі Показано, що середнє число кроків для досягнення будь-якої вершини з будь-якої початкової буде даватися наступною оцінкою:

$$M = D\sum_{k=1}^{N} k\frac{c}{k^{y}} = ClnN\sum_{k=1}^{N}\frac{1}{k^{y-1}} \leq C(\ln[N])^{2} \qquad (1.4)$$

Таким чином, показано, що при наділенні множини пошуку топологією графу, процедура пошуку в умовах "холодного старту" буде досить ефективною за кількістю кроків для вирішення завдання.

1.5 Класифікація рекомендаційних систем

В залежності від системи отримання рекомендацій методи
розподіляють на такі:

а) методи на основі змісту;

б) методи на основі колаборативної фільтрації;

в) гібридні методи;

1.5.1 Методи на основі змісту

Суть цього підходу полягає в тому, що ми співставляємо користувачів з тим контентом або товарами, які їм подобалися або були ними куплені. Тут важливі атрибути користувачів і продуктів..Для рекомендацій до фільмів використовують такі ознаки, як режисер, актори, тривалість фільму, жанр., щоб знайти схожість між фільмами. Крім того, ми можемо витягти такі характеристики, як оцінка настроїв і оцінки TF-IDF з описів фільмів і оглядів. Мета контент-орієнтованих методів - створити «профіль» для кожного користувача і кожного предмета.

Вектор предмета: це вектор довжини N. Він містить значення 1 для слів, які мають дуже високий бал за TF-IDF в цій статті, в іншому випадку значення 0.

	bacon	beans	beautiful	blue	breakfast	brown	dog	eggs	fox	green	ham	jumps	kings	lazy	love	quick	sausages	sky	toast	today
0	0.00	0.00	0.60	0.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.00	0.0
1	0.00	0.00	0.49	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.57	0.00	0.00	0.49	0.00	0.0

Вектор користувача: знову вектор розмірністю 1xN. Для кожного слова ми зберігаємо ймовірність появи слова в статтях, які вжив користувач. Зверніть увагу, що вектор користувача заснований на атрибутах елемента.

Побудувавши ці «профілі», ми обчислюємо подібності між користувачами і предметами. Предмети повинні бути рекомендовані користувачеві, якщо: вони мають найбільшу схожість з користувачем або мають велику схожість з іншими елементами, прочитаними користувачем. Є кілька способів зробити це. Давайте подивимося на 2 поширені методи:

– Косинусна схожість:

Щоб рекомендувати предмети, які найбільш схожі на ті, якими цікавився користувач, ми обчислюємо косинусну схожість між предметами, які користувач подивився, і іншими. Найбільш схожі будуть рекомендовані.

$$cosine(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \quad (1.5)$$

Щоб обчислити схожість між користувачем і предметом, ми просто беремо косинусну схожість між вектором користувача і вектором предмета.

– Схожість Жаккара

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (1.6)$$

Використовується для подібності предмет-предмет. Ми порівнюємо вектори елементів один з одним і повертаємо найбільш схожі предмети.

Подібність Жакара корисна тільки тоді, коли вектори містять бінарні значення. Якщо у них є ранжування або рейтинги, які можуть приймати більше двох можливих значень, схожість Жаккара не застосовується.

Ці методи корисні, коли у нас є безліч «зовнішніх» ознак, таких як погодні умови, ринкові чинники і т.д., які не є власністю користувача або продукту і можуть сильно варіюватися. Наприклад, ціна відкриття і закриття попереднього дня грає важливу роль у визначенні прибутковості інвестування в конкретну акцію. Це відноситься до класу supervised прикладів завдань, коли є мітки, які можуть позначати, чи сподобався користувачеві продукт, клікнув він на нього (0/1), або рейтинг, який користувач вказав цього продукту, або кількість одиниць, куплених користувачем. На рис 1.7 зображені переваги та недоліки методів на основі змісту.

Контент-ориентированные рекомендации

Преимущества	Недостатки
Независимость от данных других пользователей	Когда появляется новый пользователь с недостатком данных о его транзакциях, мы не можем качественно делать рекомендации.
Нет проблемы "холодного старта" для новых предметов, т.к. используя признаки предметов, мы можем легко находить похожие предметы	Формирование четких групп похожих продуктов может ограничить рекомендации других продуктов. Мы можем снова и снова рекомендовать лишь малое подмножество из всех продуктов
Результаты рекомендаций интерпретируемы	Если информация о продуктах ограниченна, трудно различать предметы и группировать их. В результате качество рекомендаций будет низким

Рисунок 1.7 – переваги та недоліки методів на основі змісту

1.5.2 Методи на основі колаборативної фільтрації;

Основне припущення підходу коллаборативной фільтрації полягає в тому, що якщо A і B купують аналогічні продукти, A, швидше за все, купить продукт, який купив B, ніж продукт, який купив випадкова людина. На відміну від контентно-орієнтованого підходу, тут немає ознак, відповідних користувачам або предметів. Все, що у нас є - це Матриця корисності.На рисунку 1.8 зображена матриця корисності. :

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Рисунок 1.8 – Матриця корисності

A, B, C, D - користувачі, а стовпці представляють медіафайли. Значення представляють оцінки, які користувачі дає фільму. В інших випадках ці значення могли б бути 0/1 в залежності від того, чи дивився користувач фільм чи ні. На рис 1.9 зображено принципи роботі колаборативної фільтрації.

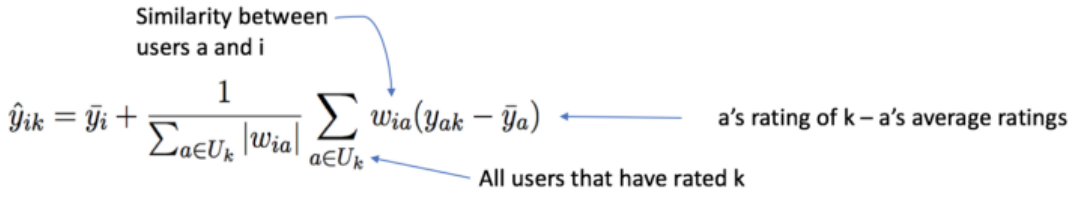
Рисунок 1.9 – Принципи роботи колаборативної фільтрації

Існує дві широких категорії, на які можна розділити колаборативну фільтрацію:
– Підхід на основі пам’яті

Для цього підходу запам'ятовується матриця корисності, і рекомендації складаються шляхом запиту даного користувача до іншої частини матриці корисності. Є медіафайли т і користувачі U, і ми хочемо дізнатися, наскільки користувач і любить кіно k.

$$\overline{Y_i = \frac{1}{|Ii|}\sum_{i \in I_i} Y_{ij}} \tag{1.7}$$

Це середній рейтинг, який сформований для користувача і на основі всіх медіафайлів, які він коли-небудь оцінював. Використовуючи це, ми оцінюємо рейтинг фільму k для користувачі і наступним чином:



(1.8)

Подібність між користувачами ***a, i*** можна обчислити з використанням будь-яких методів, таких як косинусна схожість, схожість Жаккара, коефіцієнт кореляції Пірсона.

Ці результати дуже легко отримувати та інтерпретувати, але як тільки дані стають занадто розрідженими, продуктивність погіршується. Підхід на основі моделі

Однією з найбільш поширених реалізацій підходу на основі моделі є матрична факторизація. В цьому випадку ми створюємо уявлення користувачів і предметів з матриці корисності. На рисунку 1.10 зображена матрична факторизація.:

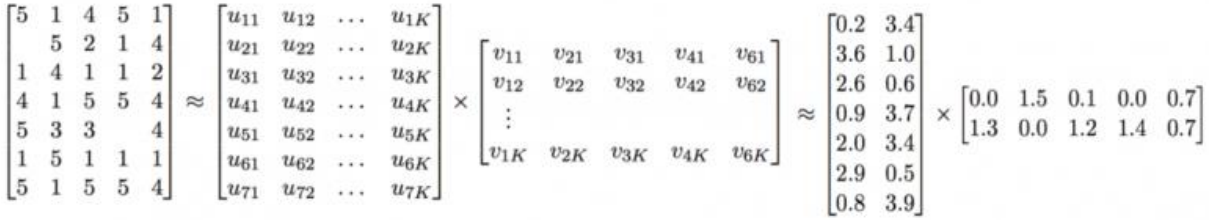


Рисунок 1.10 – Матрична факторизація

Таким чином, наша матриця корисності розкладається на U і V, де U представляє користувачів, а V представляє фільми в низькорозмірних просторі. Це може бути досягнуто за допомогою методів розкладання матриць, таких як SVD або PCA, або навчання двох векторних уявлень з використанням нейронних мереж за допомогою деякого оптимізатора, такого як Adam, SGD.

$$Y_{ij} = \dot{U}_i * U_j \tag{1.9}$$

Для користувача ***i*** та кожного медіафайлу ***j*** нам просто потрібно обчислити рейтинг у і рекомендувати медіафайли з найвищим прогнозованим рейтингом. Цей підхід найбільш корисний, коли у нас є тонна даних, і вони мають високу розрідженість. Матрична факторизация допомагає шляхом зниження розмірності, що прискорює обчислення. Одним з недоліків цього методу є те, що знижуватися інтерпритація, оскільки ми не знаємо, що саме означають елементи векторів користувачів / предмети.

– Кластеризація

Якщо дуже мало історичних, розмічених даних, ви можете кластеризувати спостереження на основі набору ознак, а потім призначити рекомендації для кластерів на основі міток, які є у об'єктів в цьому кластері.

Це рішення, звичайно, не дає кращих результатів відразу, але є гарною відправною точкою для таких випадків, поки не буде отримано достатньо даних. Кластеризація також може бути використана для створення мета-ознак для об'єктів. Наприклад, після кластеризації можна призначити значення від $1 - k$ в якості нового

елемента «кластер» для кожного спостереження, а потім навчити основну модель всіх функцій. Це може бути зроблено на рівні користувача або продукту.

– Статистичні метрики

Вони використовуються для оцінки точності методу фільтрації шляхом порівняння прогнозованих рейтингів безпосередньо з фактичним рейтингом користувачів. Середня абсолютна помилка MAE, среднеквадратическая помилка $RMSE$ і кореляція зазвичай використовуються в якості статистичних метрик. MAE є найбільш популярним і широко використовуваним - це міра відхилення рекомендації від фактичної вартості користувача. MAE і RMSE розраховуються наступним чином:

$$MAE = \frac{1}{N}\sum |predicted - actual| \quad \Big| \quad RMSE = \sqrt{\frac{1}{N}\sum (predicted - actual)^2} \tag{1.10}$$

Чим нижче значення MAE і RMSE, тим точніше механізм рекомендацій прогнозує призначені для користувача рейтинги. Ці метрики зручні, коли рекомендації засновані на прогнозуванні рейтингу або кількості транзакцій. Вони дають нам уявлення про те, наскільки точні наші прогнози і, в свою чергу, наскільки точні наші рекомендації.

1.6 Огляд існуючих інформаційно-аналітичних систем

Оскільки сьогодні стримінгові сервіси стають домінуючим та основним середовищем для прослуховування улюблених фільмів, ці сервіси потокової передачі відео можуть збирати велику кількість даних про своїх клієнтів. Такі потокові сервіси, як Netflix, Megogo або Prime Video, використовують ці дані для надання рекомендацій клієнтам. Ці системи рекомендацій є частиною більш широкого класу рекомендаційних систем, які фільтрують інформацію для прогнозування вподобань користувача.

Рекомендаційні системи поділяють на декілька основних класів, що використовують:

Коллаборативна фільтрація - рекомендації, основані на історії оцінок як самого користувача, так і інших користувачів. Цей підхід володіє теоретично високою точністю, але при цьому має одну немаловажну проблему - високий рівень входу.

Основні на вмісті - рекомендації, основані на даних, зібраних про кожен конкретний товар. Користувачі рекомендують об'єкти, подібні до них, які на ранній основі зацікавились або вже придбали. Великий плюс - можливість зацікавити нового користувача пропозиціями які вже робили споживачі системи. Для цього не потрібно довго збирати дані про товари які подобаються користувачам. Можливо, рекомендувати навіть ті об'єкти, які не отримали оцінку інших користувачів. Основні недоліки - сильна залежність від предметної області, зниження точності та обмеження рекомендацій.

Основні на знаннях - рекомендації на основі знань про предметну область. Цей тип рекомендацій має високу точність, пропонуючи користувачу, що йому потрібно. Крім цього, система вивчає та аналізує взаємозв'язок між об'єктами, виховує ряд додаткових опцій, що відносяться до індивідуальних властивостей конкретного користувача. До таких властивостей відносяться демографічні особливості (вихідні дані, що використовують найбільші соціальні мережі, такі як Facebook, LinkedIn,). Основної мінус - складність розробки та збору даних.

Гібридні - рекомендації, основані на комбінуванні коллоборативних і контентних підходів, що дозволяє уникнути більшості недоліків, цих систем. Основний недолік ц складність розробки.

1.6.1 Аналіз сервісу Netflix

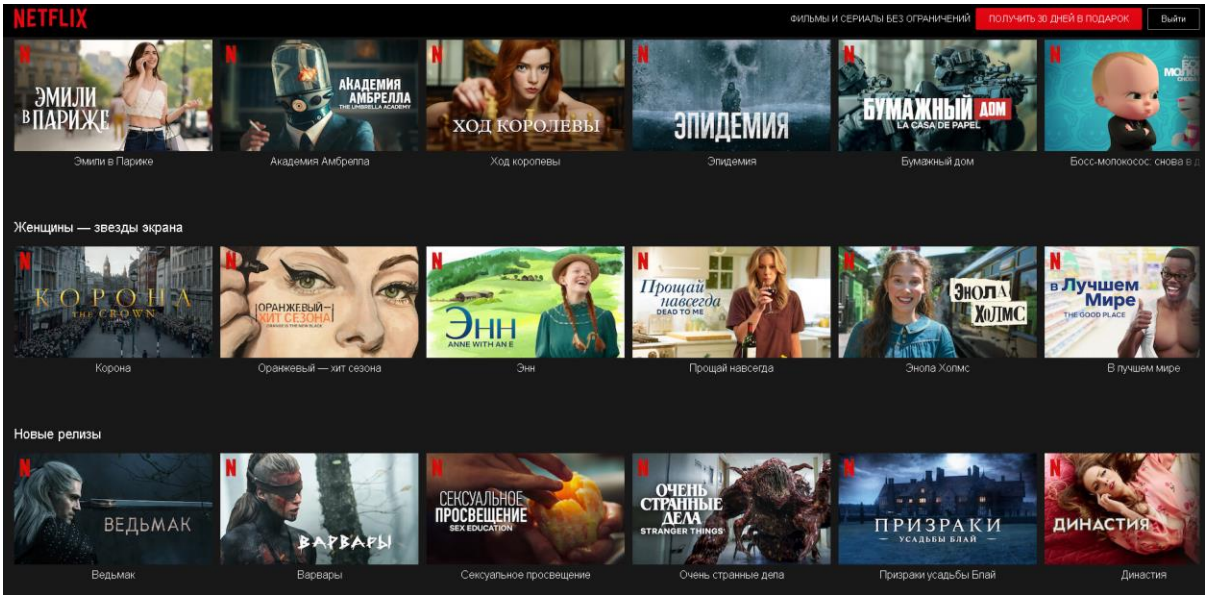


Рисунок 1.1 – «Стримінговий сервіс Netflix»

Netflix - це популярний в світі американський відео-сервіс, який продає інтернет-доступ до великої кількості фільмів і серіалів на основі потокового мультимедіа.

Однією з відмінних рис Netflix є те, що нові серіали виходять не по одній серії раз в тиждень, а відразу ж цілим сезоном в один день одночасно по всьому світу на різних мовах.

На всіх пристроях підтримується синхронізація перегляду. Тобто, можна подивитися 30 хвилин фільму на телевізорі, потім запустити Netflix на смартфоні і продовжити дивитися з того ж моменту.

В розділ Мій список можна додавати будь-які відео, щоб потім продовжити перегляд, а також для того, щоб Netflix пропонував вам більше подібного контенту.

Netflix рекомендуватиме фильми, які користувач ще не дивився, але часто дивляться інші користувачі зі схожими інтересами. Оскільки такий підхід використовує поведінку та вподобання користувачів, він є прикладом чистої колаборативної фільтрації

1.6.2 Аналіз сервісу Megogo

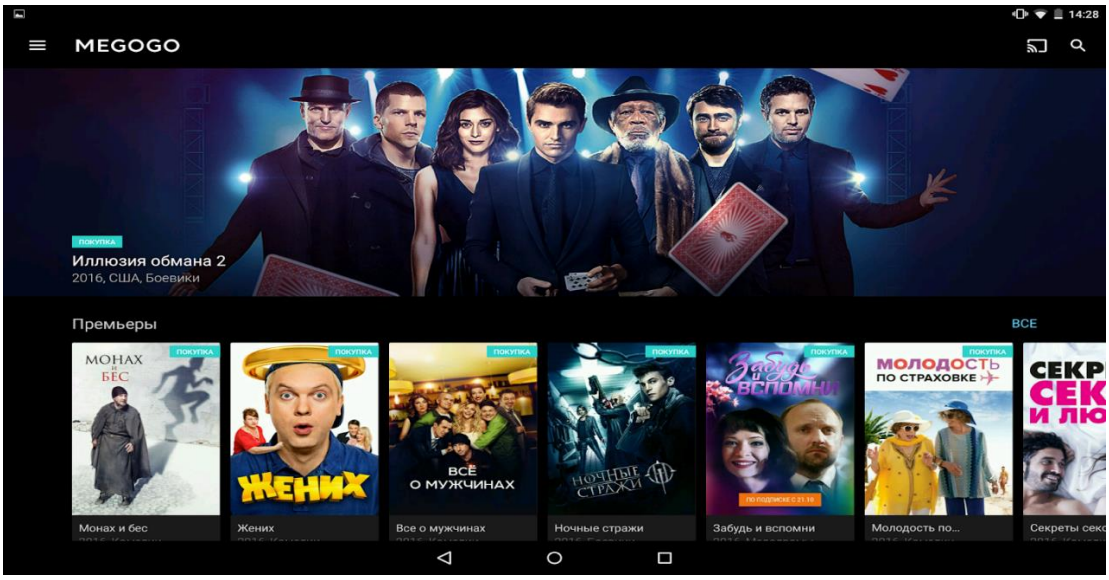


Рисунок 1.2 – «Стримінговий сервіс Megogo»

Сервіс дає доступ не тільки до кіно і серіалів, а й до телевізійним каналам. У користувачів також є можливість користуватися аудіокнигами, підкастами, 2D-фільмами з каталогу.

Відеоресурс є Мультиекранним і доступний на різних платформах на базі iOS, Android, Windows, а також на телевізорах з функцією Smart TV, мультимедійних плеєрах. Відеосервіс має власну студію сурдоперекладу. Він має в своєму розпорядженні спеціальний розділ на сайті з тисячами фільмів для людей з вадами слуху.

На основі оцінок користувачів і критиків (IMDb і «Кинопоиск») складається рейтинг, який полегшує вибір.

1.6.3 Аналіз сервісу Prime Video

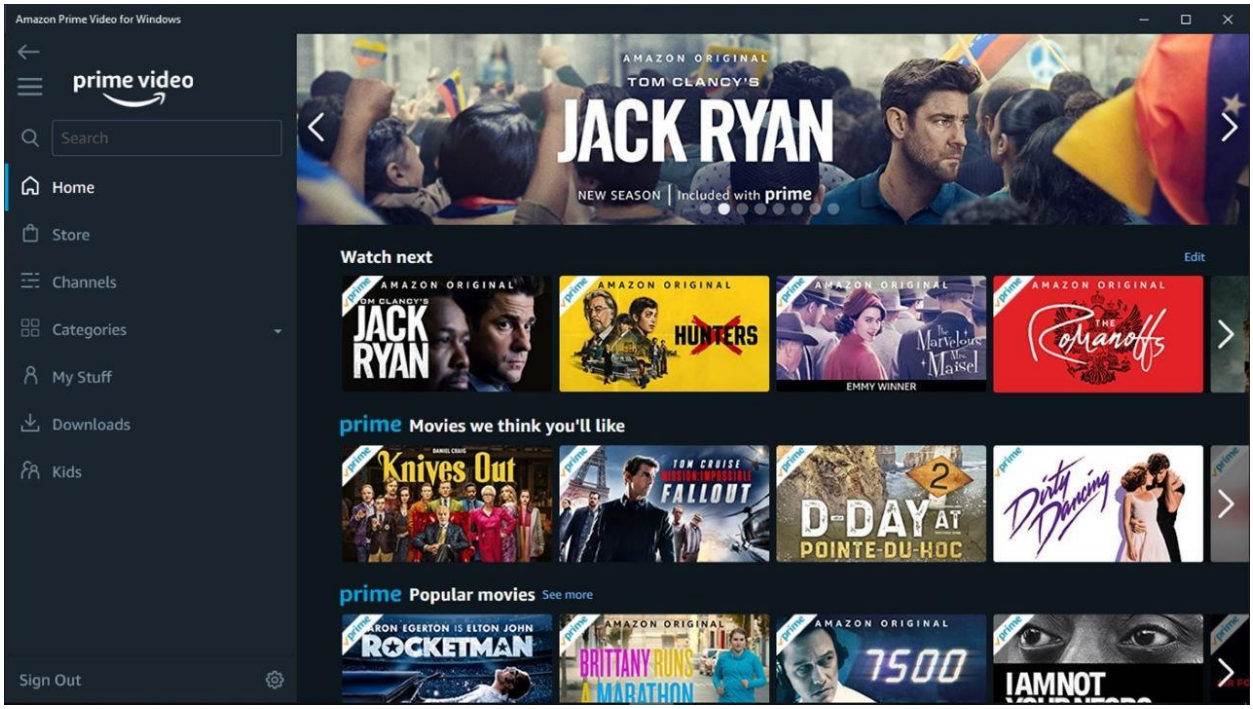


Рисунок 1.3 – «Стримінговий сервіс Prime Video»

Інтернет-відеосервіс компанії Amazon. Надає платний доступ до телевізійних передач, серіалів і фільмів власного виробництва (Amazon Studios) і інших виробників в рамках підписки Amazon Prime . В залежності від економічного становища країни умови підписки відрізняються. Застосовує контентний підхід до рекомендацій, за спільними властивостями власне самих відео, дозволяючи налаштувати вподобання користувача за жанрами.

1.7 Дослідження методів та алгоритмів прогнозування рекомендацій

Рекомендаційні системи це експертні системи, вони потрібні для рекомендації товарів та послуг. Головна проблема цих систем, зводиться до проблеми оцінювання рейтингу товарів, про які користувач ще не знає. Ця оцінка базується на рейтингу, який користувач оцінив інші

товари та на деякій додатковій інформації. Якщо визначити рейтинги для елементів, які ще не були рецензовані, то ми можемо порекомендувати користувачу елементи з найвищим передбачуваним рейтингом. На рисунку 1.4 зображено типову схему роботи систем рекомендацій.

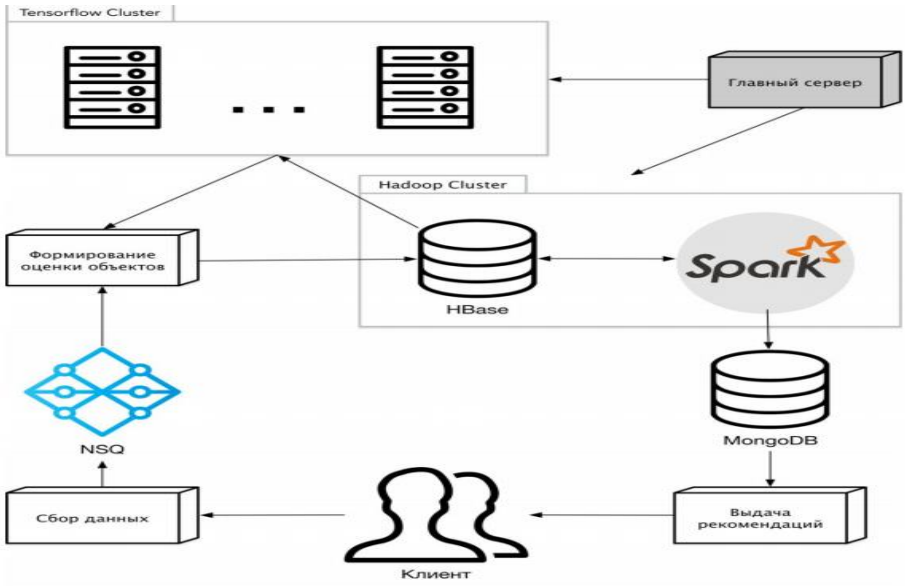


Рисунок 1.4 - «Схема роботи системи рекомендацій»

На вибір користувача щодо рекомендованих товарів та послуг впливають безліч факторів. Використовуючи інформацію, надану з різних джерел, наприклад, фотографії з тегами або відвідування користувачем різних сайтів сервісів, система намагається допомогти користувачу, та генерує рекомендації, які будуть корисними у процесі прийняття рішень та вибору.

Транскодування відео - це процес перетворення відеофайлів, які спочатку не підтримуються медіаплеєром, в сумісний формат.

У відео транскодування може виконуватися як в процесі пошуку файлів, так і для презентації. Наприклад, файли Cineon і DPX широко використовувалися як загальний формат для цифрового кіно, але розмір даних двогодинного фільму становить близько 8 терабайт. Такий великий розмір може збільшити вартість і

складність обробки файлів фільмів. Однак перекодування в формат без втрат має кращу продуктивність стиснення, і в багатьох випадках може стискати зображення до половинного розміру.

Передача - це процес, аналогічний Транскодуванню, в якому файли закодовані до нижчого бітрейту без зміни відеоформатів, це може включати перетворення частоти дискретизації, але може використовувати ідентичну частоту дискретизації з більш високим стисненням. Це дозволяє поміщати дані носії в менші місця для зберігання або по каналу з більш низькою пропускною здатністю. На рисунку 1.5 зображено типову схему роботи транскодування.

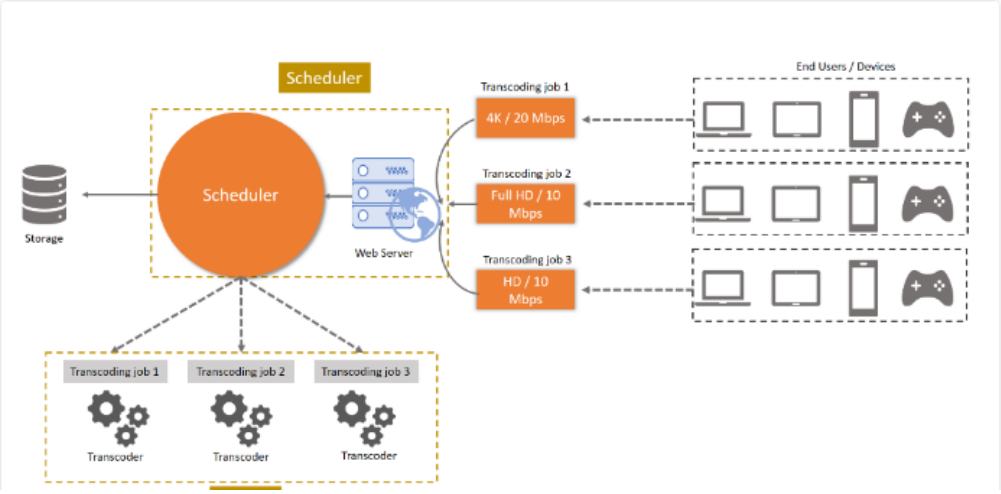


Рисунок 1.5 - «Схема роботи типової системи транскодування»

1.8 Постановка задачі

Предметною областю атестаційної роботи є методи розробки компонентів інформаційно-аналітичної системи, що автоматизують електронну комерцію стримінгового сервісу.

Метою магістерської атестаційної роботи є розробка та дослідження компонентів інформаційно-аналітичної системи електронних продажівна основі методів транскодування інформаціїта прогнозування рекомендацій.

Для розробки компонентів інформаційно-аналітичної системи необхідно виконати наступні завдання:

– провести дослідження методів побудови інформаційно аналітичних системіз використанням методівтранскодування та прогнозування рекомендацій;

– провести аналіз реалізованих систем;

– розробити системні вимоги;

– визначити функціональні вимоги;

– розробити модель потоків даних;

– розробити вимоги до клієнтської частини системи;

– розробити діаграму варіантів використання інформаційно-аналітичної системи для визначення вимог до клієнтської частини;

– розробити діаграми послідовності дій та кооперації для визначення порядку взаємодії елементів системи;

– розробити базу даних на платформі «PostgreSQL та MongoDB»;

–програмно реалізувати прототип стримінгового вебсервісу, який використовуватиме розроблені алгоритми рекомендацій та транскодування;

– провести тестування програмного продукту.

2 РОЗРОБКА КОМПОНЕНТІВ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ЕЛЕКТРОННИХ ПРОДАЖІВ

2.1 Розробка системних вимог до інформаційно-аналітичної системи

Предметною областю для проектування інформаційно-аналітичної системи була вибрана сфера відеохостингу. Алгоритм Cinematch буде впроваджений до інформаційно-аналітичної системи, як метод(или алгоритм лучше) рекомендацій. Розроблювана система має виконати завдання автоматизації роботи відеохостингу та використовувати рекомендаційний метод.

Завдяки використанню та впровадженнювдосконаленого методу прогнозуваннярекомендацій в інформаційній системі будуть пропонуватися товари, котрі можуть зацікавити кінцевих користувачів.

Інформаційна система призначена для продажу підписок на відеохостинг та ведення комерційного бізнесу в мережі Інтернет, для підвищення прибутку компанії. Виходячи з описаної інформації та цілей розробки даної системи були поставлені наступні вимоги до її проектування:

В системі мають бути присутніми два логічних блока: серверна частина і клієнтська частина.

Серверна частина повинна реалізовувати внутрішню роботу всієї системи, реалізовувати алгоритм реалізації рекомендацій в системі, відповідати на запити клієнтської частини та надавати їй інтерфейс для керування інформаційною системою. База даних серверної частини (СУБД) повинна реалізовувати внутрішні процедури збереження й доступу до даних.

Клієнтська ж система вирішує питання взаємодії системи з користувачами. Вона взаємодіє з серверною частиною, відправляючи на сторону сервера запити та отримує

від нього необхідні дані. Ці дані вона має показувати кінцевому користувачу в зручному для його розуміння вигляді, приховуючи технічні деталі функціонування інформаційної системи.

Інформаційна система, повинна бути масштабованою, тому що в майбутньому, кількість користувачів системи буде зростати.

Системні вимоги до системи автоматизації:

а) серверна частина інформаційної системи має бути розроблена з використанням платформи СУБД PostgreSQL;

б) клієнтська частина інформаційної системи має бути реалізована в вигляді сторінок з використанням мови програмування JavaScript та Java;

в) реалізований інтерфейс доступу має працювати під управлінням web-серверу Tomcat та Jetty)

2.2 Визначення функціональних вимогдо інформаційно-аналітичної системи

Щоб визначити функціональні вимоги до CRM системи, була розроблена її функціональна модель з використанням стандарту IDEF0. Цей стандарт дозволяє виділити основні бізнес-процеси в системі.

Стандарт IDEF0 – це методологія функціонального моделювання, використовуючи яку, система, що вивчається, постає у вигляді набору взаємопов’язаних функцій (функціональних блоків) [1].

Будь-яка схема IDEF0 розпочинається з концептуальної діаграми. Ця діаграма описує систему в узагальненому вигляді.

У графічних матеріалах зображені вхідні дані. Після надходження до входу системи, вони обробляються для отримання кінцевого результату. На виході контекстної діаграми зображується дані, що є результатом діяльності функцій системи. Контекстна діаграма також містить документи та інструкції, що необхідні для правильного управління процесами, а також механізми, що представляють собою ресурс/ресурси, для виконання основної бізнес-функції. Головною функцією системи є облік замовлень товарів.

Розроблена контекстна діаграма відображає погляд на функції системи з точки зору адміністратора інформаційної системи відеохостингу. Для того щоб більш детально зрозуміти, на які функції буде розбиватися основний бізнес-процес проводиться функціональна декомпозиція.

У графічних матеріалах зображена декомпозиція концептуальної діаграми. З її допомогою відбувається уточнення основної функції розробки компонентів інформаційно аналітичної системи для відеохостингу.

До декомпозиції увійшли такі підфункції, що реалізуються системою:

- з авторизація/реєстрація;
- перегляд каталогів медіафайлів;
- оформлення замовлень;
- адміністрування;
- перегляд каталогу підписок;
- аналітична обробка даних.

Функція «авторизація/реєстрація» потребує для свого виконання дані користувача (для реєстрації) або дані для авторизації (для входу в систему). Після її виконання на вхід функції «перегляд каталогу медіафайлів» будуть надходити дані про вже авторизованого юзера. Окрім того, на вхід функції будуть надходити дані про медіафайли та рекомендаційні медіафайли, а також дані для пошуку медіафайлів. В результаті виконання даної функції, на виході буде отримано каталог медіафайлів. Для функції «Оформлення замовлень» на вхід надходять дані про замовлення (тобто дані, що будуть вводитися в форму замовлень) а також дані про підписки в кошику користувача, результатом її роботи будуть дані замовлення. Функція «Адміністрування» отримує на вхід дані про медіафайли, дані про підписки та дані замовлення, а на виході виводить дані замовлень. На вхід функції «Аналітична обробка даних» надходять дані для оцінки медіафайлів та дані, що необхідні для управління аналізом. Результатом виконання будуть дані рекомендації медіафайлів і підписок, та дані аналітичних звітів.

У графічних матеріалах зображена декомпозиція функції системи «авторизація/реєстрація». З її допомогою відображаються ті основні процеси, що відповідають за забезпечення реєстрації нових користувачів системи та авторизації користувачів, що вже зареєстровані в системі.

Функція розбивається на наступні підфункції:

- заповнення форми реєстрації;
- перевірка на незайнятість логіну та паролю в БД;
- вивід форми про результати реєстрації;
- заповнення форми авторизації;
- перевірка на знаходження логіну та паролю в БД;
- вивід форми, про результати авторизації.

На рис. 2.4 зображена декомпозиція функції «Забезпечення перегляду медіафайлів». Дана функція відповідає за дії системи при перегляді асортименту медіафайлів, як авторизованими, так і не авторизованими користувачами. Крім того, демонструє коли будуть застосовуватися результати роботи рекомендаційного алгоритму, що буде розроблюватися для системи. Рекомендації будуть відображатися при перегляді усього асортименту медіафайлів, а також на сторінці перегляду інформації про медіафайл.

Функція розбивається на наступні підфункції:

- перегляд медіафайлів;
- вибрати медіафайл.

Для виконання функції перегляду медіафайлів необхідно отримати на її вхід дані про медіафайли, рекомендаційні медіафайли, та, при необхідності, дані для пошуку медіафайлів, в результаті виконання функції, буде отриманий каталог медіафайлів. Він буде надходити до входу функції «Вибрати медіафайл», на виході будуть отримані дані вибраного медіафайлу.В результаті виконання функції, будуть отримані дані про товари в кошику.

У графічних матеріалах представлена декомпозиція функції «Оформлення замовлення». Функція відображає один з найголовніших процесів в системі – оформлення замовлення. На діаграмі продемонстрована послідовність виконання процесів в системі, що необхідні для того, щоб забезпечити оформлення замовлення на підписку.

Функція розбивається на ряд підфункцій:

- заповнення даних форми для створення замовлення;
- виведення форми, що замовлення успішно оформлене замовлення;
- обробка замовлення.

Для виконання функції «Виведення даних в форму для створення замовлення» на вхід надходять дані для замовлення, про підписку в кошик користувача, та, безпосередньо дані про самого авторизованого користувача. Після виконання функції на виході отримуються дані замовлення. Коли вони надходять на вхід функції «Виведення форми, що замовлення успішно оформлене» система отримує нове замовлення, що йде на вхід до функції «Обробка замовлення» Результатом роботи цієї функції є дані оформленого замовлення.

У графічних матеріалах представлена декомпозиція функції «Адміністрування» системи. Дана функція демонструє процес управління адміністратором інформаційною системою. До процесів керування системою відноситься: оновлення асортименту медіафайлів, публікування актуальної інформації та актуалізація каталогу медіафайлів; створення звітів з продажу підписок, опираючись на підписки, що часто замовляють та тих, котрі замовляють рідко.

Дана функція розбивається на наступні функції:

- введення актуальної інформації про підписки та медіафайли
- зміна статусу замовлення
- ведення звітів з продажу підписок.

Дані про товари надходять до функції «Опублікування актуальної інформації про підписки та медіафайли» результатом виконання даної функції є контент, що буде демонструватися в інформаційній системі. Для виконання функції «Зміна статусу замовлення» на вхід повинні надходити дані замовлення, а результатом роботи даної функції будуть дані замовлень та дані для звітів. Функція «ведення звітів з продажу підписок» має на своєму вході дані для звітів та дані замовлення. На виході даної функції будуть дані звітів.

У графічних матеріалах представлена декомпозиція функції «Аналітична обробка даних». Вона демонструє те, як система оброблює дані для оцінки медіафайлів, щоб відібрати рекомендації медіафайлів. Спочатку дані для оцінки медіафайлів обробляються до необхідного вигляду, а далі використовується алгоритм для отримання рекомендацій товарів.

Функція «Аналітична обробка даних» розбивається на декілька функцій:

- аналіз медіафайлів та підписок;
- генерація рекомендацій.

Дані для оцінки медіафайлів та підписок, що надходять до входу функції «Аналіз медіафайлів та підписок» оброблюються в рамках даної функції. На виході отримуються оброблені дані, що необхідні для застосування другою функцією «Генерація рекомендацій», до входу цієї функції також надходять й дані для управління аналізом. На виході даної функції будуть дані про рекомендовані медіафайли та підписки та дані аналітичних звітів.

В рамках проведеного функціонального моделювання були уточнені функціональні вимоги до розроблювальної інформаційно аналітичної системи.

Функціональні вимоги до розроблюваної системи наступні:

- забезпечення авторизації;
- забезпечення реєстрації нових користувачів;
- перегляд асортименту товарів;
- редагувати інформації про товари;

- надання рекомендацій;
- забезпечення оформлення замовлень;
- забезпечення обробки замовлень;
- забезпечення можливості адміністрування;
- забезпечення публікування актуальної інформації;
- мати можливість змінювати статус замовлення.

2.3 Розробка моделі потоків даних компонентів інформаційно аналітичної системи

Для більш точного визначення вимог до інформаційної системи є створення діаграми потоків даних DFD.

DFD-діаграми забезпечують зручний спосіб опису потоків інформації, як між частинами розроблювальної системи, так і між системою та її зовнішнім середовищем [2].

Концептуальна DFD-діаграма відображає взаємодію системи з зовнішнім середовищем, тобто інформаційних потоків між системою і зовнішніми сутностями (такими як користувач системи, чи адміністратор), з котрими вона має бути зв’язана. У графічних матеріалах зображена концептуальна діаграма, що відображає погляд на функції системи з боку адміністратора.

На контекстній діаграмі демонструється зв’язок інформаційної системи з зовнішніми сутностями «Юзер» та «Адміністратор». Зовнішня сутність «Юзер» взаємодіє з системою шляхом надання на її входи таких потоків даних як: дані для авторизації, реєстрації та замовлення. В результаті виконання внутрішніх процесів в інформаційній системі, на виході користувач отримає дані замовлення, каталог медіафайлів та підписок та підписки, медіафайли, що рекомендує система. Адміністратор системи отримує дані здійсненого замовлення та дані, що необхідні для

звітів, а також каталог медіафайлів та підписок, що необхідний для редагування, видалення та додання нових.

У графічних матеріалах представлена декомпозиція концептуальної діаграми DFD. Вона дозволяє декомпозувати зовнішні потоки інформації концептуальної діаграми та зв’язати їх з внутрішніми потоками даних, що потрібні для функціонування системи.

Основний процес розбивається на наступні роботи/процеси:

- здійснення авторизації/реєстрації користувача;
- здійснення перегляду каталогу товарів;
- оформлення замовлення;
- адміністрування;
- аналіз даних.

У графічних матеріалах представлена декомпозиція функції «Здійснення авторизації/реєстрації користувача» системи.

Для виконання функції «Здійснення авторизації/реєстрації користувача» необхідні сховища даних (Data Store):

- сутність «users» з атрибутами: «first_name», «last_name», «patronymic», «birth_day», «login», «password».

Використання сутності «users» дає можливість зберігати інформацію зареєстрованого клієнта та звіряти співпадіння логіну та паролю з тими, що вже зберігаються в системі.

У графічних матеріалах представлена декомпозиція функції «Здійснення перегляду каталогу підписок та медіафайлів» системи.

Для виконання функції «Забезпечення перегляду каталогу товарів» необхідні наступні сховища даних (Data Stores):

- сутність «Basket» атрибути якої (id, id_item, id_order, count, price, id_service) дають можливість додавати товари до корзини, для можливості їх подальшого придбання, а також переглядати інформацію про підписку в корзині.

- сутність «Subscriptions» атрибути якої (id_subscription, title, description, price, id_category) дають можливість зчитувати інформацію про підписки, котрі маються в системі
- сутність «idFiles» атрибути якої (id_file, title, description, quantity, id_category) дають можливість зчитувати інформацію про медіафайли, їх кількість котрі маються в системі

У графічних матеріалах представлена декомпозиція функції «Оформлення замовлення» системи.

Для виконання функції «Оформлення замовлення» необхідні наступні сховища даних (Data Stores):

- сутність «orders» атрибути якої (id_orders, id_user, date, price, address, telephone, id_dostavka, id_status_order) дають можливість читати інформацію з таблиць БД та використовувати її з метою аналізу даних;
- сутність «basket» атрибути якої (id_basket, id_ subscription, id_orders, count, price, id_service) дають можливість додавати підписку до корзини, для можливості її подальшого придбання, а також переглядати інформацію про підписку в кошику.

У графічних матеріалах представлена декомпозиція функції «Адміністрування» системи.

Для виконання функції необхідні наступні сховища даних (Data Stores):

- сутність «idfiles» атрибути якої (id_dile, title, description, price, id_category) дають можливість зчитувати інформацію про медіафайли, котрі маються в системі
- сутність «subscriptions» атрибути якої (id_subscriptions, title, description, price, quantity, id_category) дають можливість зчитувати інформацію про підписки, їх кількість котрі маються в системі

- сутність «orders» атрибути якої (id_orders, id_user, date, price, adress, telephone, id_status_order) дають можливість читати інформацію з таблиць БД, та змінювати її ;

У графічних матеріалах представлена декомпозиція функції «Аналіз даних» системи.

Для виконання функції необхідні наступні сховища даних (Data Store):

- сутність «users», атрибути, якої («last_name», «first_name», «patronymic», «birth_day», «FKidGender» «login», «password») дають можливість отримати інформацію про користувача, та використовувати її для аналізу.
- сутність «orders» атрибути якої (id_order, id_user, date, price, adress, telephone, id_status_order) дають можливість читати інформацію з таблиць БД та використовувати її з метою аналізу даних;

В результаті проведення моделювання потоків даних, були уточнені функціональні вимоги до системи, що розроблюється, а також визначені сутності, що будуть використовуватися в структурі бази даних.

2.4 Розробка вимог до інтерфейсуweb-сторінок клієнтської частини системи

В інформаційній системі клієнтська частина реалізується в вигляді додатку для локальної мережі (web-сторінки глобальної мережі Інтернет). Необхідно уточнити вимоги до інтерфейсу клієнтської частини інформаційно-аналітичної системи відеохостингу.

Інтерфейс web-сторінок доступу до даних системи має забезпечувати виконання наступних аспектів:

- відкрити повний доступ адміністратору до всіх функцій системи;
- обмежити доступ до деяких функцій системи для незареєстрованого користувача. Він не повинен мати змогу купити підписку;

- відкрити доступ до таких функцій системи, як корзина покупця та оформлення замовлення в системі, для авторизованого користувача.

2.5 Діаграма варіантів використання системи для визначення вимог до її інтерфейсу

Діаграма варіантів використання (Use Case Diagram) – це графічне представлення всіх акторів, прецедентів та їх взаємодій у системі. У кожній системі, як правило, є головна діаграма прецедентів, котра відображає границі системи (акторів) та основну функціональну поведінку системи (прецеденти) [4].

У графічних матеріалах зображена діаграма варіантів використання розроблюваної інформаційно-аналітичної системи. Вона є послідовністю дій, котра виконується системою у відповідь на певну подію, ініційовану зовнішнім об’єктом (діючим актором).

На схемі варіант використання описує типову взаємодію між користувачем та системою, а кожне діюче лице представляє собою певну роль, яку грає користувач по відношенню до системи.

На діаграмі варіантів використання зображено три діючих актора: незареєстрований користувач, зареєстрований користувач, адміністратор.

Незареєстрований користувач – це користувач, що має обмежені права використання системи. Він може здійснювати наступні дії із системою: перегляд каталогу, авторизація та реєстрація.

Актор «Зареєстрований користувач» – це користувач, що має права для використання сервісами системи. Він може здійснювати всі дії, що й незареєстрований користувач, а також додатково: «Робота з корзиною» (додавання, видалення, перегляд одиниць), а також «Оформлення замовлення».

Актор «Адміністратор» – це користувач, що має розширені права використання системи. Він може здійснювати підготовку звітів, роботу з замовленнями, керування каталогами

Варіант використання «Перегляд каталогів» – забезпечує одну із важливих функцій системи: перегляд всіх медіафайлів та підписок, що є в системі.

Варіант використання «Пошук за категорією» являється розширенням для базового прецедента «Перегляд каталогів». Використовується користувачем в разі його бажання використати фільтри для пошуку медіафайлів, що належать певній категорії.

Варіант використання «Реєстрація» – забезпечує функцію реєстрації незареєстрованого користувача в системі, для того, щоб він мав більш розширений доступ до системи.

Варіант використання «Авторизація» – забезпечує процес вводу даних авторизації (логіна та пароля) для того, щоб користувач авторизувався в системі.

Варіант використання «Робота з корзиною» – забезпечує роботу користувача з корзиною, в якій розміщується підписка, яку користувач бажав би придбати.

Варіант використання «Додавання підписки в корзину» – прецедент, який включається в вигляді зіставного компонента в послідовність поведінки прецедента «Робота з корзиною». Прецедент «Додавання підписки в корзину» забезпечує можливість додати підписку до корзини користувача.

Варіант використання «Видалення підписки з корзини» також включається, як зіставний компонент до прецеденту «Робота з корзиною». Він забезпечує можливість видалення користувачем підписки, котра його не цікавить.

Варіант використання «Перегляд підписки, що є в корзині» включається, як зіставний компонент прецеденту «Робота з корзиною». Він забезпечує можливість перегляду користувачем споживчої корзини.

Варіант використання «Оформлення замовлення» – забезпечує можливість користувача системи оформити замовлення на придбання відповідної підписк

Варіант використання «Робота з замовленням» – забезпечує можливість керувати замовленням адміністратору: перевіряти, змінювати статус.

Варіант використання «Підтвердження замовлення» – забезпечує можливість адміністратора системи редагувати статус замовлення. Цей прецедент включається в базову послідовність дій прецедента «Робота з замовленням»

Варіант використання «Відхилення замовлення» – включається в базову послідовність дій прецедента «Робота з замовленням». Він забезпечує можливість відхилити замовлення, яке не було підтверджено користувачем.

Варіант використання «Керування каталогами» - адміністратор системи має слідкувати за публікацією актуальної інформації про підписки та медіафайли.

Варіант використання «Додавати нову підписку або медіафайл» – включається в базову послідовність дій прецедента «Керування каталогами». Він забезпечує можливість додавати інформацію про нову підписку або медіафайл.

Варіант використання «Редагувати інформацію про товапідписку або медіафайлр» – включається в базову послідовність дій прецедента «Керування каталогами». Він забезпечує можливість редагувати інформацію та дані про підписку або медіафайл.

Варіант використання «Видалити підписку або медіафайл» - включається в базову послідовність дій прецедента «Керування каталогами». Він забезпечує можливість видалення підписки або медіафайлу.

2.6 Діаграми послідовності дій системи

Діаграма послідовності визначає тимчасову послідовність повідомлень, що передаються, їх порядок, вид та тип повідомлення, що відбувається в рамках варіантах використання. [10]

У графічних матеріалах представлена діаграма послідовності дій для «Реєстрації» системи відеохостингу. Даний сервіс можуть використовувати

незареєстровані користувачі. Слід виконати наступні дії, для того, щоб зареєструватися: ввести html-запит, для того, щоб відкрилась головна сторінка. Після того, як відкриється головна сторінка, необхідно відкрити вкладку реєстрації. Після того, як вкладка відобразиться, необхідно ввести ім’я, прізвище, по-батькові, день народження, а також логін та пароль. Після цього натиснути клавішу «Регистрация». Дані будуть перевірені на предмет незайнятості логіну в базі даних. Якщо такий логін не використовується, користувач отримає повідомлення про успішну авторизацію.

У графічних матеріалах представлена діаграма послідовності дій для сервісу «Додавання підписки до корзини» інформаційної системи, що виконується зареєстрованим користувачем. Користувачу спочатку необхідно ввести html-запит для того щоб перейти на головну сторінку. Далі користувач відкриває каталог підписок – web-сторінку з інформацію про підписки в каталозі. Він вибирає з каталогу підписку та нажимає кнопку «Оформить». Після цього цей товар кладеться до корзини. Дані передаються до БД, а згодом товар опиняється в корзині користувача.

У графічних матеріалах представлена діаграма послідовності дій для сервісу «Видалення підписки з каталогу» інформаційної системи.

Для того, щоб адміністратор зміг видалити медіафайл з каталогу, спочатку йому необхідно ввести html-запит, для того, щоб відобразилась головна сторінка для адміністратора (адмін-панель). Після цього, адміністратор відкриває вкладку «Редактирование». Вона відображається. Адміністратор вибирає необхідний медіафайл, вона виділяється системою та натискає на кнопку «Видалити». Дані видаляються з бази даних, після чого користувач може побачити, що медіафайл видалився.

3 ВДОСКОНАЛЕННЯ МЕТОДІВ ТРАНСКОДУВАННЯ МЕДІАФАЙЛІВ ТА ПРОГНОЗУВАННЯ РЕКОМЕНДАЦІЙ

3.1 Метод транскодування медіафайлів

Процес перетворення потокового або аудіо чи відео одного кодека в інший. Формат файлу (контейнер) при цьому не змінюється.

Буває, що деякі пристрої (телевізори, смартфони, консолі) не підтримують формат аудіо / відео X, але підтримують формат аудіо / відео Y. Або ж новіші кодеки краще, оскільки забезпечують кращу ступінь стиснення.

Була виділена пам'ять для компонента AVFormatContext, який буде містити інформацію про формат (контейнер).

Тепер можна відкрити файл, прочитати його заголовок і заповнити AVFormatContext мінімальної інформацією про формат (зазвичай кодеки не відчиняються). Для цього використовується функція avformat_open_input. Він очікує AVFormatContext, ім'я файлу і два необов'язкових аргумента: AVInputFormat (якщо ви передасте NULL, FFmpeg визначить формат) і AVDictionary (які є опціями).

Щоб отримати доступ до водних потоків, нам потрібно прочитати дані з носія. Це робить функція avformat_find_stream_info. Тепер pFormatContext-> nb_streams буде містити кількість потоків, а pFormatContext-> streams [i] дасть нам і-й за рахунком потік (AVStream). Для кожного потоку ми збираємося зберегти AVCodecParameters, що описує властивості кодека, використовуваного і-м потоком:

Використовуючи властивості кодеків можемо знайти відповідний, запитуючи функцію avcodec_find_decoder, також можемо знайти зареєстрований декодер для ідентифікатора кодека і повернути AVCodec - компонент, який знає, як кодувати і декодувати потік:

За допомогою кодека виділяємо пам'ять для AVCodecContext, який буде містити контекст для нашого процесу декодування / кодування. Але потім потрібно заповнити

цей контекст кодека параметрами CODEC - робиться це за допомогою avcodec_parameters_to_context.

Після того, як ми заповнили контекст кодека, необхідно відкрити кодек. Викликаємо функцію avcodec_open2 і потім можемо її використовувати: AVCodecContext

```
pCodecContext = avcodec_alloc_context3(pCodec);
avcodec_parameters_to_context(pCodecContext, pCodecParameters);
avcodec_open2(pCodecContext, pCodec, NULL);
```

Тепер треба прочитати пакети з потоку і декодувати їх в кадри, але спочатку потрібно виділити пам'ять для обох компонентів (AVPacket і AVFrame). Далі треба відправити пакети до функції av_read_frame, поки у неї є пакети.

Можна відправити пакет необроблених даних (стиснений кадр) в декодер через контекст кодека, використовуючи функцію avcodec_send_packet.

Щод отримати кадр необроблених даних (нестислий кадр) від декодера через той же контекст кодека, треба використати функцію avcodec_receive_frame.

Коли проектується відеоплеєр, потрібно відтворювати кожен кадр в певному темпі, інакше важко отримати задоволення від відео або через те, що воно відтворюється занадто швидко, або занадто повільно.

Тому потрібно визначити якусь логіку для плавного відтворення кожного кадру. В цьому відношенні кожен кадр має часову мітку уявлення (PTS - від presentation timestamp), яка являє собою збільшення кількості, що враховується в змінної timebase, яка представляє собою раціональне число (де знаменник відомий як тимчасової масштаб - timescale), ділене на частоту кадрів (fps).

Для fps = 60/1 і timebase = 1/60000 кожен PTS буде збільшувати timescale / fps = 1000, тому реальний час PTS для кожного кадру може бути (за умови, що воно починається з 0) Тепер з pts_time можна знайти спосіб візуалізувати це синхронізованим зі звуком pts_time або з системним годинником. FFmpeg libav надає цю інформацію через свій API.

```
fps = AVStream->avg_frame_rate
```

Загальне використання методу слідує такому шаблону та архітектурі:

рівень протоколу - прийняття вхідних даних (наприклад, файл, але це може бути також завантаження по rtmp або HTTP);

рівень формату - демультиплексирование контенту, відкриття, в основному, метаданих і потоків;

рівень кодека - декодування даних для стислих потоків при необхідності;

рівень пікселів - також можливе застосування деяких фільтрів до необроблених кадрів (зміна розміру), необов'язковий пункт;

рівень кодека - кодування (перекодування або транскодування) необроблених кадрів;

рівень формату - мультиплексування (або ремультиплексування) необроблених потоків (стислих даних);

рівень протоколу - і, нарешті, мультиплексовані дані відправляються на вихід (в інший файл або, на сервер. На рис. 3.1 Зображена схема роботи по леєрам.

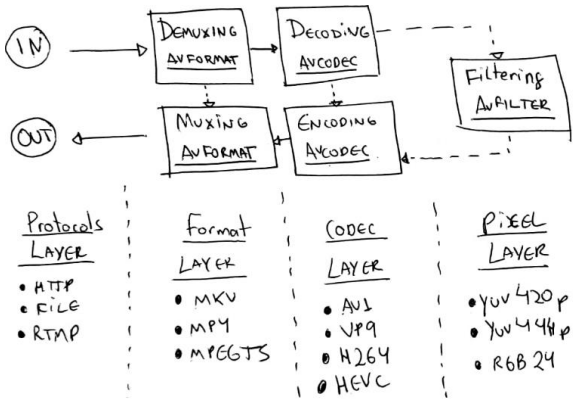


Рисунок 3.1 – Схема роботи по леєрам.

Будемо ремультиплексировати тільки потоки відео, аудіо та субтитри. Тому треба зафіксувати, які потоки будемо використовувати, в массив індексів:

```
number_of_streams = input_format_context->nb_streams;
streams_list = av_mallocz_array(number_of_streams, sizeof(*streams_list));
```

Після того, як виділили необхідну пам'ять, потрібно виконати цикл по всім потокам, і для кожного з них треба створити новий вихідний потік в нашому контексті вихідного формату, використовуючи функцію avformat_new_stream. Помічені всі потоки, які не є відео, аудіо або субтитрами, щоб була можливість пропустити їх.

Лістинг 3.1 – Фільтрація потоків

```
for (i = 0; i < input_format_context->nb_streams; i++) {
  AVStream *out_stream;
  AVStream *in_stream = input_format_context->streams[i];
  AVCodecParameters *in_codecpar = in_stream->codecpar;
  if (in_codecpar->codec_type != AVMEDIA_TYPE_AUDIO &&
      in_codecpar->codec_type != AVMEDIA_TYPE_VIDEO &&
      in_codecpar->codec_type != AVMEDIA_TYPE_SUBTITLE) {
    streams_list[i] = -1;
    continue;
  }
  streams_list[i] = stream_index++;
  out_stream = avformat_new_stream(output_format_context, NULL);
  if (!out_stream) {
    fprintf(stderr, "Failed allocating output stream\n");
    ret = AVERROR_UNKNOWN;
    goto end;
  }
  ret = avcodec_parameters_copy(out_stream->codecpar, in_codecpar);
  if (ret < 0) {
    fprintf(stderr, "Failed to copy codec parameters\n");
    goto end;
  }
}
```

Після цього можна копіювати потоки, пакет за пакетом, з нашого введення в наші вихідні потоки. Це відбувається в циклі, поки є пакети (av_read_frame), для кожного пакета потрібно перерахувати PTS і DTS, щоб нарешті записати його

(av_interleaved_write_frame) в наш контекст вихідного формату. Схема методу представлена у графічних матеріалах.

Лістинг 3.2 – контекст вихідного формату

```
while (1) {
  AVStream *in_stream, *out_stream;
  ret = av_read_frame(input_format_context, &packet);
  if (ret < 0)
    break;
  in_stream  = input_format_context->streams[packet.stream_index];
  if (packet.stream_index >= number_of_streams || streams_list[packet.stream_index] <
0) {
    av_packet_unref(&packet);
    continue;
  }
  packet.stream_index = streams_list[packet.stream_index];
  out_stream = output_format_context->streams[packet.stream_index];
  /* copy packet */
  packet.pts    =    av_rescale_q_rnd(packet.pts,    in_stream->time_base,    out_stream-
>time_base, AV_ROUND_NEAR_INF|AV_ROUND_PASS_MINMAX);
  packet.dts    =    av_rescale_q_rnd(packet.dts,    in_stream->time_base,    out_stream-
>time_base, AV_ROUND_NEAR_INF|AV_ROUND_PASS_MINMAX);
  packet.duration  =  av_rescale_q(packet.duration,  in_stream->time_base,  out_stream-
>time_base);
  packet.pos = -1;

  6c1
  ret = av_interleaved_write_frame(output_format_context, &packet);
  if (ret < 0) {
    fprintf(stderr, "Error muxing packet\n");
    break;
  }
  av_packet_unref(&packet);
}
```

AVFormatContext - це абстракція для формату медіа-файлу, тобто для контейнера (MKV, MP4, Webm, TS)

AVStream представляє кожен тип даних для даного формату (наприклад: аудіо, відео, субтитри, метадані)

AVPacket - це фрагмент стислих даних, отриманих з AVStream, які можуть бути декодовані за допомогою AVCodec (наприклад: av1, h264, vp9, hevc), що генерують необроблені дані, звані AVFrame.

Тепер налаштуємо декодер. AVFormatContext надасть нам доступ до всіх компонентів AVStream, і для кожного з яких можемо отримати їх AVCodec і створити конкретний AVCodecContext. І, нарешті, можемо відкрити даний кодек, щоб перейти до процесу декодування. AVCodecContext містить дані про конфігурацію мультимедіа, такі як швидкість передачі даних, частота кадрів, частота дискретизації, канали, висота.

Також потрібно підготувати вихідний медіа-файл для перетворення. Спочатку виділимо пам'ять для вихідного AVFormatContext. Створимо кожен потік в вихідному форматі. Щоб правильно упакувати потік, копіюємо параметри кодека з декодера. Встановлюємо прапор AV_CODEC_FLAG_GLOBAL_HEADER, який повідомляє кодувальнику, що він може використовувати глобальні заголовки, і, нарешті, відкриваємо вихідний файл для запису і зберігаємо заголовки:

Після того, як підготовлений декодер, але перед тим, як організувати вихідний мультимедійний потік, треба налаштувати код для кодування.

Треба створити екземпляр відео AVStream в методі avformat_new_stream;

далі треба мтворити контекст AVCodecContext на основі створеного кодека avcodec_alloc_context3;

встановити основні атрибути для сеансу транскодування;

відкрити кодек і зкопіювати параметри з контексту в потік (avcodec_open2 і avcodec_parameters_from_context).

Лістинг 3.3 – налаштування коду для кодування

```
AVRational input_framerate = av_guess_frame_rate(decoder_avfc, decoder_video_avs, NULL);
AVStream *video_avs = avformat_new_stream(encoder_avfc, NULL);

char *codec_name = "libx265";
char *codec_priv_key = "x265-params";
// we're going to use internal options for the x265
// it disables the scene change detection and fix then
// GOP on 60 frames.
char *codec_priv_value = "keyint=60:min-keyint=60:scenecut=0";

AVCodec *video_avc = avcodec_find_encoder_by_name(codec_name);
AVCodecContext *video_avcc = avcodec_alloc_context3(video_avc);
// encoder codec params
av_opt_set(sc->video_avcc->priv_data, codec_priv_key, codec_priv_value, 0);
video_avcc->height = decoder_ctx->height;
video_avcc->width = decoder_ctx->width;
video_avcc->pix_fmt = video_avc->pix_fmts[0];
// control rate
video_avcc->bit_rate = 2 * 1000 * 1000;
video_avcc->rc_buffer_size = 4 * 1000 * 1000;
video_avcc->rc_max_rate = 2 * 1000 * 1000;
video_avcc->rc_min_rate = 2.5 * 1000 * 1000;
// time base
video_avcc->time_base = av_inv_q(input_framerate);
video_avs->time_base = sc->video_avcc->time_base;

avcodec_open2(sc->video_avcc, sc->video_avc, NULL);
avcodec_parameters_from_context(sc->video_avs->codecpar, sc->video_avcc);
```

необхідно розширити цикл декодування для транскодування відео потоку:

– відправити порожній AVPacket декодера (avcodec_send_packet);

– отримати нестислий AVFrame (avcodec_receive_frame);

– далі почнеться перекодування необробленого кадру;

– треба відправити необроблений кадр (avcodec_send_frame);

– отримати стиснення, засноване на нашому кодеку AVPacket (avcodec_receive_packet);

– встановити позначку часу (av_packet_rescale_ts);

– записати у вихідний файл (av_interleaved_write_frame).

Код транскодування представлений у додатку.

3.2 Метод прогнозування рекомендацій

Алгоритм отримання даних:

Заводиться черга даних, в якій зберігаються ідентифікатори користувача;

Якщо черга порожня, то треба взяти id по порядку, котрий задан веб сервісом, якщо ні, то дастається черговий користувач із черги;

Якщо користувач був оброблений раніше, то він ігнорується. І переходимо до пункту 2;

По даному id користувача пдгружається html сторінка веб сервісу, котра співпадає цьому користувачу. Із сторінки буруться дані про користувача;

Підгружаються дані про користувачів, котрі робили відгуки похожі на данного користувача і також додаємо їх до черги;

Підгружаються дані про медіафайли на які зробів відгук данний користувач;

Підгружається html сторінка медіафайлів. Підгружається інформація про медіафайли;

Зробивши аналіз всіх медіафайлів переходимо до пункту 2.

Реалізація залежить від оцінок, які користувач ставив раніше (від середньої оцінки за всіма оціненим користувачем фільмів), середніх оцінок "схожих" користувачів і коефіцієнтів "схожості":

$$r_{u,i} = \dot{r} + \frac{\sum_{u \in U} simil(u,i)(r_{u,i} - \dot{ru})}{\sum_u simil(u,i)} \quad (3.1)$$

Для передбачення оцінок фільмів потрібні середні рейтинги за всіма оціненим фільмів користувачів. У кожного користувача, є закладка вибраних медіафайлів яку можна розпарсити і отримати дані зі списку. Зібрані дані можна знайти в файлі user_favorites.csv, а сам код збору у додатку. Для кожного зібраного медіафайлу проходимся по всім користувачам і рахуємо кількість появ його. Для кожного користувача рахуємо його схожість з усіма іншими користувачами, і якщо для іншого користувача подібність не нуль, то оновлюємо схожість input-користувача і відповідного медіафайлу. В кінці нормалізуємо і сортуємо в порядку спадання. Код методу представленій у додатку.

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДУ ТРАНСКОДУВАННЯ ТА ПРОГНОЗУВАННЯ РЕКОМЕНДАЦІЙ

4.1 Обґрунтування вибору мови та технології програмування

Для розробки інформаційно-аналітичної системи була вибрана мова програмування Java та технологія Spring.

Сьогодні продукти Java зустрічаються всюди. Завдяки надійності Java використовують у різноманітних сферах. Застосовується для розробки ПЗ в освіті, в науці, в приватному секторі, в розробці серверних додатках. Мова програмування Java має великий вибір бібліотек та дає можливість працювати з різними фреймворками.

Java - об'єктно-орієнтована мова, вона підтримує три основні принципи об'єктно-орієнтованого програмування: інкапсуляцію, успадкування і поліморфізм. [1].

Гіганти IT-індустрії, такі як Eclipse та NetBeans, зіграли значну роль у розвитку Java. Ці компанії розробили зручні інтегровані середовища розробки (Eclipse, IntelliJ IDEA), в котрих писати програмний код, на мові програмування Java, стало набагато простіше та швидше.

Крім IDE, Java має декілька інших інструментів наприклад для збірки: Maven і Ant, JConsole та VisualVM для моніторингу Heap.

Java має бібліотеки з відкритим кодом. Це дуже розвиває цю мову програмування. Наприклад Apache розробили величезну кількість бібліотек та фреймворків, що полегшують та оптимізують розробку.

Великою перевагою є захист від несанкціонованого доступу. Java аплікації не можуть викликати глобальні функції та отримувати доступ до відкритих полів аплікації. Це забезпечує в Java рівень безпеки, якого немає в деяких популярних інших

мовах програмування. Рівень безпеки забезпечує JVM , вона вбудована в операційну систему.

Я вибрав мову програмування Java, оскільки всі інструменти, котрі необхідні, а також багато корисних особливостей які дають можливість доволі просто розробити додаток, особливо Web-сервіси. Всі інструменти знаходяться в одному наборі, котрий підтримується.

Spring Framework —фреймворк з відкритим кодом та контейнер інверсії управління. Особливості Spring Framework можна використати будь-яким додатком Java.

Spring складається з модулів, які дають багато можливостей таких як: STests, MVC, управління транзакціями, SpringData, SpringSecurity.

Spring MVC (Model-View-Controller) - модуль на основі HTTP сервлета, він забезпечує зручне створення веб-додатків та RESTful сервісів.

Фреймворк Spring MVC розроблений на паттерні Model - View - Controller (Модель - Відображення - Контролер) за допомогою звязаних компонентів. Патерн MVC розділяє аспекти додатки (логіку додатку, UI, DataLayer), і забезпечує вільний зв'язок між цими компонентами.

MapReduce - це парадигма програмування, підходяща для обробки великих даних. На рис 4.1 зображена робота MapReduce.

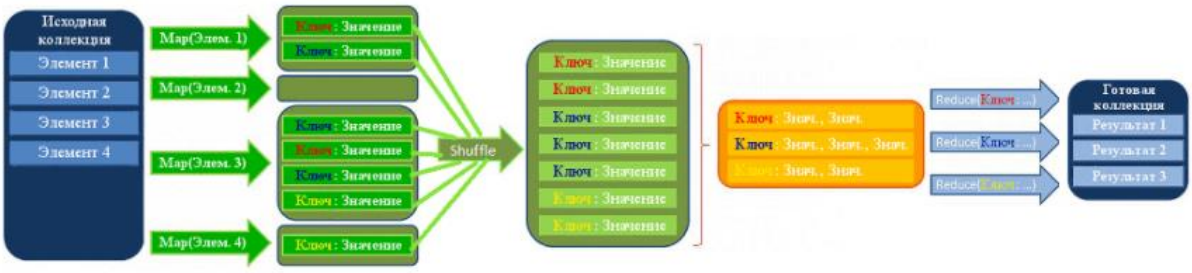


Рисунок 4.1 – Робота MapReduce в проекті

Map приймає на вхід список значень і якусь функцію, яку потім застосовує до кожного елементу списку і повертає новий список;

reduce (згортка) - перетворює список до єдиного атомарному значення за допомогою заданої функції, якій на кожній ітерації передаються новий елемент списку і проміжний результат.

Для обробки даних відповідно до обчислювальної моделі в MapReduce визначені обидві ці функції, вказані імена вхідних і вихідних файлів, а також параметри обробки.

Сама обчислювальна модель складається з 3-хшаговой комбінації вищенаведених функцій:

Map - попередня обробка вхідних даних у вигляді великого список значень. При цьому головний вузол кластера (master node) отримує цей список, ділить його на частини і передає робочим вузлам (worker node). Далі кожен робочий вузол застосовує функцію Map до локальних даних і записує результат в форматі «ключ-значення» в тимчасове сховище.

Shuffle, коли робочі вузли розподіляють дані на основі ключів, раніше створених функцією Map, таким чином, щоб всі дані ключа знаходяться на одному вузлі.

Reduce - паралельна обробка вузлом групи даних по порядку проходження ключів зєднання результатів на master ноді. Головний вузол отримує проміжні відповіді від робочих вузлів і передає їх на вільні вузли для виконання наступного кроку. На рис 4.2 зображеній алгоритм роботи MapReduce.

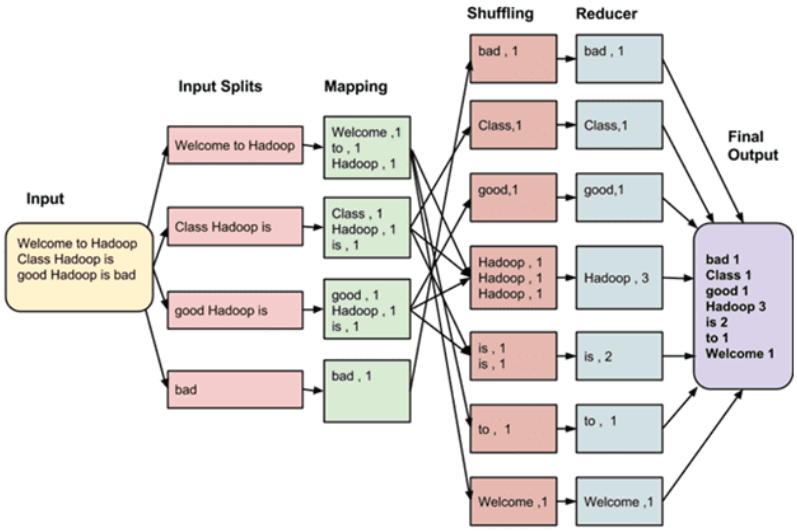


Рисунок 4.2 – Алгоритм роботи MapReduce

ApacheHadoopMapReduce - Hadoop MapReduce - це програмне середовище для простого створення програмного забезпечення, які обробляють величезні обсяги даних паралельно на великих кластерах стандартного устаткування надійним і відмовостійким способом. Завдання MapReduce розбивати вхідний набір даних на незалежні блоки, які обробляються завданнями карти повністю паралельно. Платформа сортує вихідні дані карт, які потім вводяться в завдання скорочення. Зазвичай і введення, і висновок завдання зберігаються в файлової системі. Фреймворк піклується про планування завдань, контролює їх і повторно виконує невиконані завдання. Платформа MapReduce складається з одного головного JobTracker і одного підлеглого TaskTracker на кожен вузол кластера. Провідний пристрій відповідає за планування задач компонентів завдань на ведених пристроях, моніторинг їх і повторне виконання невиконаних завдань. Підлеглі виконують завдання відповідно до вказівок ведучого. Додаток визначає місцеположення введення / виведення , надають карту і скорочують функції через реалізації відповідних інтерфейсів і / або абстрактних класів. Ці та інші параметри завдання становлять конфігурацію завдання. Потім клієнт завдання Hadoop відправляє завдання (jar) І конфігурацію в JobTracker, який потім бере на себе відповідальність за поширення програмного забезпечення,

конфігурації на ведені пристрої, планування завдань і їх моніторинг. Платформа MapReduce працює виключно з парами <ключ, значення>, тобто платформа розглядає вхідні дані для завдання як набір пар <ключ, значення> і створює набір пар <ключ, значення> в якості вихідних даних, імовірно різних типів. Класи ключів і значень повинні бути Серіалізуємі (Serializable) платформою і, отже, повинні реалізовувати інтерфейс Writable. Крім того, ключові класи повинні реалізувати інтерфейс WritableComparable, щоб спростити сортування фреймворком. В графічних матеріалах представлений приклад коду при аналізі контенту.

4.2 Обгрунтування вибору методів рекомендаційної системи

Один з найпопулярніших алгоритмів для вирішення проблем спільної кластеризації (і особливо для спільних рекомендаційних систем) називається матричної факторизації (MF). У своїй простій формі він передбачає матрицю оцінок, присвоєних точкам. На рис 4.3 зображено приклад матричної факторизації.

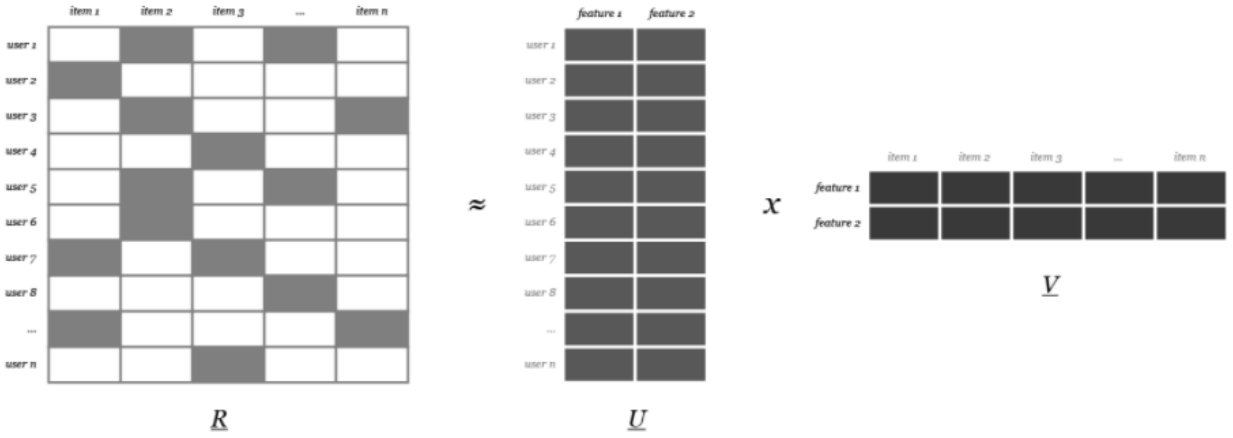


Рисунок 4.3 – Приклад матричної факторизації

У кожній непорожній ячейці міститься оцінка користувача певної одиниці контенту. Розмірність матриці в нашому випадку: дуже багато глядачів в рядках матриці на десятки тисяч одиниць контенту в стовбцях матриці.

Далі проводиться розкладання вихідної матриці на дві матриці U та V з деякою обраною кількістю чинників. Ці матриці являють собою стислу проекцію розподілу даних з деякими втратами. У нашому випадку фактори в стислій формі відображають все розмаїття переваг користувачів. Матриця U - проекція глядачів на переваги. Матриця V - проекція контент-об'єктів.

Матриці факторів повинні містити в собі такий набір значень, щоб множення U x V давало в результаті матрицю з близькими до вихідних значеннями.

Головна складність цього методу - знайти чинники. Для цього використовуємо машинне навчання у вигляді методу оптимізації Gradient Descent. Для його роботи необхідна диференційована метрика (функція втрат), яку ми будемо мінімізувати. Очевидно, що метрика повинна показувати помилку наближення множення матриць чинників до вихідної матриці. Базова формула виглядає так:

$$\min_{x_\star, y_\star} \sum_{r_{u,i} \text{ is known}} (r_{ui} - x_u^T y_i)^2 + \lambda(\|x_u\|^2 + \|y_i\|^2)$$

Формула дозволяє не тільки обчислити помилку, але також включає в себе L2 регуляризацію, сила впливу якої задається параметром λ.

Існує ряд алгоритмів пошуку чинників. Один з них - SGD (Stochastic Gradient Descent), що працює на семплах даних, забезпечуючи хорошу масштабованість, але не гарантує збіжність. Основний алгоритм який ми використовуємо - ALS (Alternating Least Squares), який також виконує градієнтний спуск. Його особливість полягає в тому, що кожна ітерація спуску відбувається в два етапи (приватне диференціювання

з почергової фіксацією матриць чинників користувачів і фільмів) і гарантує плавну збіжність.

Цей метод спрямований на заповнення відсутніх записів в матриці асоціації користувач-елемент. Spark.ml підтримує спільну фільтрацію на основі моделей, при якій користувачі і продукти описуються невеликим набором прихованих чинників, які можна використовувати для прогнозування відсутніх записів. spark.ml використовує алгоритм альтернативних найменших квадратів (ALS) для вивчення цих прихованих чинників. Реалізація в spark.ml має наступні параметри:

1) numBlocks - це кількість блоків, на які будуть розділені користувачі і елементи, щоб распаралелити обчислення (дефолтне значення10).

2) ранг - це кількість прихованих чинників в моделі (дефолтне значення 10).

3) maxIter - максимальна кількість ітерацій для запуску (дефолтне значення 10).

4) regParam вказує параметр регуляризації в ALS (дефолтне значення 1.0).

5) mplicitPrefs вказує, чи використовувати варіант ALS з явною зворотним зв'язком або варіант, адаптований для неявних даних зворотного зв'язку (за замовчуванням false, що означає використання явною зворотного зв'язку).

6) альфа - параметр, застосовний до неявного варіанту ALS, який регулює базову впевненість у спостереженнях за уподобаннями (дефолтне значення 1.0).

У багатьох реальних сценаріях використання зазвичай доступне лише для неявної зворотної зв'язку (наприклад, перегляди, кліки, покупки, оцінки). Підход, використовуваний у spark.ml для роботи з такими даними, береться із спільної фільтрації для наборів даних неявної зворотної зв'язку. По суті, замість того, щоб спробувати змоделювати матрицю оцінки напруги, цей підхід формує дані, як кількість, що представляє силу спостереження за діями користувача (наприклад, кількість кліків). Потім ці цифри пов'язані з рівнем впевненості у спостережуваних перевагах користувачів, а не з явними оцінками, присвяченими елементам. Модель моделі намагається знайти приховані фактори, які можна використовувати для прогнозування очікуваних переважних користувачів відповідно до елементів.

По дефолту Spark визначає прогнози NaN у віпадку ALSModel.transform, коли фактор користувача або елемент не присутній у моделях. В нашому випадку це корисно у системі, так як воно відображає нового користувача або елемент, і тому система може прийняти рішення про використання резервного копіювання для використання у прогнозі.

Однак це небажано під час перехресної перевірки, оскільки будь-які передбачені значення NaN приведуть до результатів NaN для метрики оцінки (наприклад, при використанні RegressionEvaluator). Це робить неможливим вибір моделі.

Spark дозволяє встановлювати для параметра coldStartStrategy значення «drop», щоб видалити будь-які рядки в DataFrame прогнозів, які містять значення NaN. Потім метрика оцінки буде обчислена для даних, відмінних від NaN, і буде дійсною. Використання цього параметра представлено в графічних матеріалах.

4.3 Обґрунтування вибору концепції аналітичної бази даних

Сховища виконують завдання накопичення відомостей про діяльність підприємства, інших інформаційних ресурсів з різних джерел, в тому числі баз даних, що відображають окремі бізнес-процеси, автоматизованих робочих місць, інформаційних систем та інших джерел інформації, в тому числі з глобальних інформаційних мереж, як наприклад Internet.

Виділяють такі характерні для них властивості:

- предметну орієнтованість;
- інтегрованість;
- незмінність;
- підтримка хронології.

Властивість предметної орієнтованості означає компоновку пулів інформації за певними предметним областям або цілям, що забезпечує підготовку і

прийняття відповідних рішень у відповідній системі DSS або використання

в будь-якому іншому вигляді додатку, наприклад для розробки планів розвитку підприємства, інвестиційних проектів, реінжинірингу і для будь-яких робіт, для виконання яких необхідні вихідні дані, накопичені в інформаційному сховищі.

Необхідно підкреслити, що зосереджені в сховищі дані підтримують аналітичну діяльність, а не функціональні або технологічні процеси. Чи не безпосередню виробничу діяльність, а аналіз даних.

Інтегрованість передбачає збір і доопрацювання (попередню обробку) інформації по певній предметній області з різних джерел і перетворення її в організований за заданими правилами, підлеглим певної мети, масив у вигляді гіперкуба або системи полікубів інформації. Користувачу легко використовувати спільне використання даних з різних секцій сховища, що відносяться до різних напрямків діяльності або функцій підприємства.

Незмінність полягає в тому, що в сховищі дані з різних джерел накопичуються, практично не піддаються будь-яким змінам, тільки в разі крайньої необхідності, в основному при виявленні спотворень. Вони нарощуються по заданому, чітко визначеним графіку. Користувач має тільки право читання інформації. Маніпуляції з вмістом сховища довіряються тільки адміністраторам.

Підтримка хронології полягає в обов'язковості прив'язки даних до часу.

Використовують наступні типи прив'язки до часу:

- на момент скоєння події або факту;
- до моменту фіксації його інформаційними засобами;
- комбіновані методи, що поєднують обидва підходи.

На рис 4.4 зображено структуру інформаційного сховища.

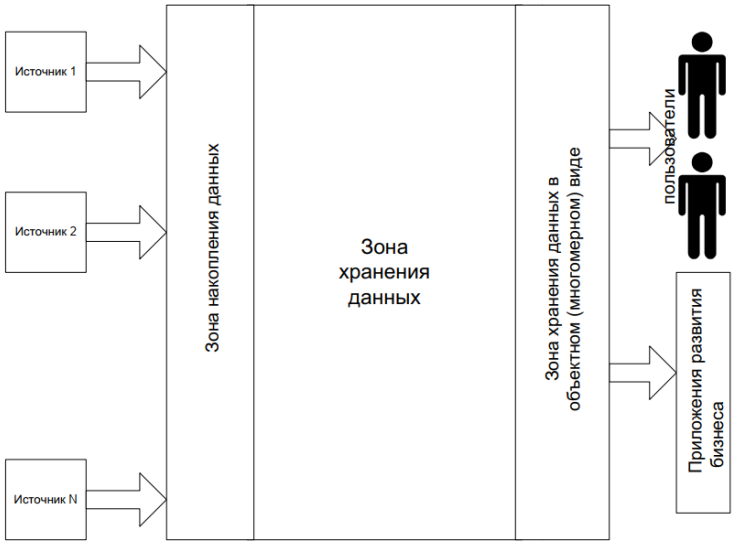


Рис 4.4 – структура інформаційного сховища

Дані в сховищі з джерел накопичуються протягом певного періоду часу в зоні накопичення. Протягом цього часу проводиться робота щодо забезпечення необхідної якості даних по описаним вище правилам. В процесі перекачування з джерел дані трансформуються в єдиний формат, перевіряються їх смислова узгодженість, наявність помилок, проводяться заходи щодо підвищення якості даних. При досягненні необхідного рівня якості та визначеного графіком роботи моментів часу дані переносяться в зону зберігання.

У зоні зберігання вони можуть бути представлені у вигляді реляційної або багатовимірної моделі (об'єктному вигляді). У разі використання реляційної моделі необхідно мати в сховище зону об'єктного представлення даних для досягнення рівня характеристик системи, що відповідає вимогам, що пред'являються до інформаційно аналітичних систем.

Реалізація може бути кількома способами.

В нашій задачі був вибір між двома. Концепція централізованого сховища даних та концепція розподіленого сховища.

Централізований підхід говорить, що є декілька джерел інформації - і створюється єдине центральне сховище даних. У первинних джерелах вся інформація

зберігається в первинному вигляді, в структурі даного джерела інформації або базі даних. Вся інформація яка поступає в систему повинна бути перетворена в прийняту в даній системі структуру. Транспортування даних з бази до системи, яка супроводжується аналізом, може бути організована по графіку і правилам аналізу зпринципів Інмона. На рис 4.5 представлена схема централізованого сховища.

Рисунок 4.5 –Централізоване сховище даних

Якщо реалізовувати цю концепцію треба мати дуже потужний компютер. В залежності від масштабів це може бути ПК з дуже високими характеристиками, або майнфрейми. Необхідні розвинені засоби телекомунікацій, щоб забезпечити обмін інформацією.

В нашій інформаційно-аналітичній системі потрібен протилежний підхід до зберігання даних котрий основується на розподілі функцій системи або угрупованні декількох баз даних навколо сховища інформації. Сховища можуть орієнтуватися на конкретну предметну область або на декілька. Система локальних сховищ діє в якості розподіленого сховища. Може бути і центральне сховище, але в цій структурі вимоги до розмірності легше.

Цей підхід передбачає передачу кожного запиту до джерела, обробку, аналіз, компанування даних «На льоту» і передача їх користувачу.

Такий підхід при економії ресурсів також має недоліки, до яких можна віднести:

- нормалізація даних в базах і тривалість доступу з центру час реакції такої системи набагато більше;

- повинна бути забезпечена відкритість кожного джерела інформації, так як відсутність може зірвати процес аналізу інформації;

- можлива неузгодженість даних з різних джерел інформації через різні формати представлення;

прив'язки до часу, зміна смислового значення даних;

- практична неможливість комплексного відображення інформації яка знаходиться в різних джерелах через різні механізми її зберігання - зробити один порядок для всіх дуже складно. На рис 4.6 представлено схему розподіленого інформаційного сховища.

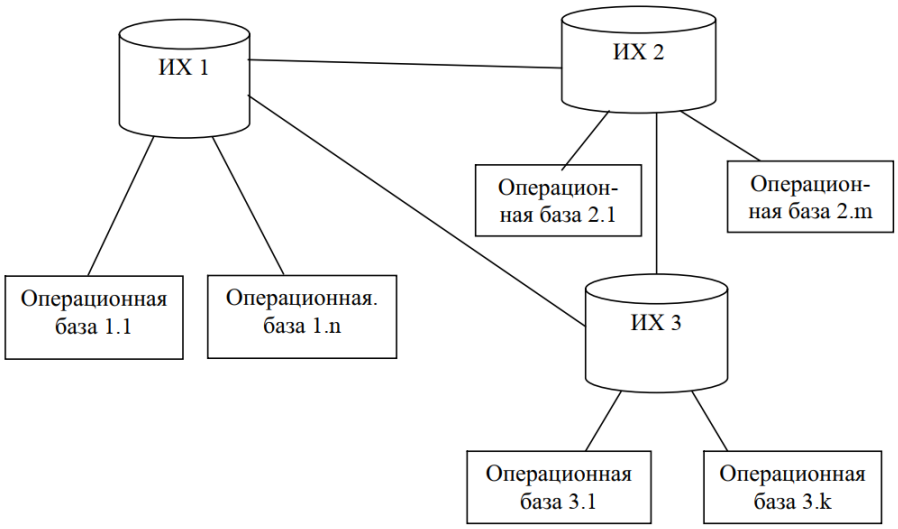


Рисунок 4.6 – Схема розподіленого інформаційного сховища

. Було вибрано цю концепцію тому що вона не потребує великих ресурсів для її використання а також її легше підтримувати чим монолітний варіант, так як у нас є багато відгалужень і якщо система дасть збій то вона зможе продовжити роботу без

частини яка вийшла з ладу. Для її реалізації були використані такі БД як реляційна БД PostgreSQL та нереляційна БД MongoDB.

База даних потрібна для збереження,аналізу, апдейту та обробки різноманітної інформації, здебільшого вони великі. Щоб здійснити доступ до бази даних та опрацьовувати дані, використовується система управління базами даних.

СУБ – це засоби, котрі необхідні щоб створити базу даних, підтримувати їх в працездатному стані для пошуку інформації. (це програмній засіб, для створення структури бази, її наповнення, редагування і відображення інформації.). [3]

Основними процесами бази даних є збереження інформації та надання доступу користувачу, додатку або сервісу до неї. База даних складається з двох частин: збереженої інформації та системою управління.

Основними функціями СУБД прийнято вважати наступні:

- управління даними;
- управління транзакціями;
- підтримка мов;
- управління буферами пам'яті.

Для розробки інформаційно-аналітичної системи для сервісу відеохостингу була вибрана СУБД PostgreSQL. Вона найбільш підходить для вибраної мови програмування та технології, що використовується.

PostgreSQL –система управління базами даних. В цій системі відкритий код. ЇЇ було створено в протилежність платним системам. Ця система використовується, для створення веб-сторінок, так як має підтримку різних мов програмування.

PostgreSQL підходить найбільше для маленьких та середніх додатків. PostgreSQL використовується як сервер, до нього приходять локальні або віддалені клієнти, також до дистрибутиву входить бібліотека внутрішнього сервера, вона дозволяє включати PostgreSQL до автоматизованих систем. Коди сервера компілюються на багатьох платформах. Найкраще вони працюють в UNIX-системах, в них є підтримка багатопоточності, що підвищує продуктивність.

Перевагами PostgreSQL є:

– масштабованість. Підтримує роботу бази даних великих розмірів, що підтверджують її реалізації в Apache, Google,. По документації, що є, існуть бази даних, які використовуються компанією PostgreSQL AB, мають по 60 млн. записів.;

– переносність. Працює на різноманітних платформах, серед яких Linux, Windows, Solaris, Mac OC. Окрім того, PostgreSQL працює на різних платформах;

– безпека. PostgreSQL має систему контроля доступу до даних, забезпечує шифрування даних при передаванні;

– швидка робота;

– зручність експлуатації. PostgreSQL досить зручно встановлюється в проект та його легко реалізувати;

– код цього проекту відкритий.

СУБД PostgreSQL це система клієнт-сервер, вона містить -SQL-сервер, який надає підтримку різних баз даних, а також декілька клієнтських додатків, бібліотек, механізми для адміну та широкий спектр програмних інтерфейсів (API). Сервер PostgreSQL представляється у вигляді багатопоточної бібліотеки, яку можна підключити до призначеного для користувача додатку і отримати компактний, більш швидкий і легкий в управлінні продукт. Доступна також велика кількість програмного забезпечення для PostgreSQL, в більшості - безкоштовного.

PostgreSQL підтримує різні типи даних (рядкові, з плаваючою точкою, цілі числа, дата й час), включаючи такі як Integer, Float, Double, Char, Varchar, Text.

4.4 Логічне та фізичне моделювання даних системи

Логічне та фізичне моделювання даних інформаційно-аналітичної системи сервісу відеохостингу проводилося з допомогою використання інструментального CASE-засобу «AllFusion ErWin Data Modeler». Даний засіб дозволяє автоматизувати

процес проектування реляційних баз даних. проектувати, документувати та супроводжувати бази даних, сховища даних та вітрини даних [4].

Методика створення моделі даних системи з використанням CASE-засобу «AllFusion ErWin Data Modeler» складається з наступних етапів:

– створення логічної моделі;

- створення фізичної моделі даних, на основі логічної;

Логічна модель даних інформаційно-аналітичної системи сервісу відеохостингу приведена на рис. 4.7.

Предметною областю системи являється діяльність відеохостингу. Сутності предметної області, котрі ввійшли в логічну модель представлені в таблиці 4.1

Таблиця 4.1 – Сутності логічної моделі даних системи

№	Найменування сутності	Найменування атрибута	Тип даних (домен)	Призначення
1.	idfile	idFile	Лічильник, цілочисловий	Первинний ключ
		Description	Текст, довжиною в 200 символів	Інформація про медіафайл
		Title	Текст, довжиною в 50 символів	Назва медіафайлу
		idCategory	Цілочисловий	Зовнішній ключ
		Image	Текст, довжиною в 100 символів	зображення медіафайлу

№	Найменування сутності	Найменування атрибута	Тип даних (домен)	Призначення
2.	Users	idUser	Лічильник, цілочисловий	Первинний ключ
	Users	FirstName	Текст, довжиною в 50 символів	Прізвище користувача
		LastName	Текст, довжиною 50 символів	Ім’я користувача
		Birthday	Дата	День народження користувача
		Login	Текст, довжиною в 15 символів	Логін для входу на сайт
		Password	Текст, довжиною в 20 символів	Пароль для входу на сайт
		idGender	Цілочисловий	Вторинний ключ, стать користувача
3.	Category	idCategory	Лічильник, цілочисловий	Первинний ключ
		Title	Текст, довжиною в 50 символів	Назва категорії
4.	Basket	idBasket	Лічильник, цілочисловий	Первинний ключ

№	Найменування сутності	Найменування атрибута	Тип даних (домен)	Призначення
	Basket	idItem	Лічильник, цілочисловий	Складовий ключ
	Basket	idOrders	Цілочисловий	Зовнішній ключ
	Basket	Price	Десятичний тип х фіксованою крапкою	Ціна ппідписки
	Basket	idService	Цілочисловий	Інформація про вибраний сервіс
5.	Orders	idOrders	Лічильник, цілочисловий	Первинний ключ
	Orders	Date	Дата	Дата оформлення замовлення
	Orders	Price	Десятичний тип	Ціна замовлення
	Orders	Email	Текст, довжиною в 40 символів	Електронна адреса користувача
	Orders	Telephone	Цілочисловий	Телефон користувача
	Orders	idStatusOrder	Цілочисловий	Зовнішній ключ
	Orders	idUser	Цілочисловий	Зовнішній ключ

№	Найменування сутності	Найменування атрибута	Тип даних (домен)	Призначення
6.	Status	idStatus	Лічильник, цілочисловий	Первинний ключ
	Status	Status	Текст, довжиною в 35 символів	Статус замовлення

Створення фізичної моделі бази даних – це другий етап в її проектуванні. Для створення фізичної моделі даних використовується конкретна реалізація СУБД. Для цього необхідно:

- вибрати платформу СУБД (MySQL);
- мати логічну базу даних.

Створення фізичної моделі БД інформаційно-аналітичної системи сервісу відеохостингу виконується за допомогою CASE-засобу AllFusion ErWin Data Modeler.

Фізична модель даних інформаційно-аналітичної системи сервісу відеохостингу представлена у графічних матеріалах.

4.5 Програмна реалізація функціоналу

4.5.1 Програмна розробка функції реєстрації в системі

У додатку Spring Boot MVC я використовував Spring Security для забезпечення аутентифікації та реєстрації користувачів. Код представлено у графічних матеріалах.

Користувачу буде запропоновано зайти в систему надавши ім'я (логін або email) і пароль. Ім'я користувача і пароль об'єднуються в екземпляр класу UsernamePasswordAuthenticationToken після чого він передається примірнику

AuthenticationManager для перевірки. У разі якщо пароль не збігається з назвою користувача буде викинуто виключення BadCredentialsException з повідомленням "Bad Credentials". Якщо аутентифікація пройшла успішно повертає повністю заповнений примірник Authentication.

4.5.2 Програмна розробка функції валідатора

Для забезпечення перевірки даних коли користувач авторизується розроблений метод який представлений у додатку. Якщо валідація пройшла успішно, метод продовжує роботу. Якщо ні – в додатку відбувається ексепшн. Якщо ви не знайдете цей виняток, то помилку впіймає бібліотека, поміняє flash-змінні повідомленням про помилку валідації і перенаправить користувача на головну сторінку з формою авторизації. При грамотному використанні валідації запитів в контроллерах завжди будуть тільки валідні дані.

3.5.3 Програмна розробка методу рекомендацій

Реалізовано дві рекомендаційні системи. Дані для створення методу були взяті замовні. Оскільки у нас немає ніякої інформації про переваги користувачів, ми ранжируємо мультимедійні дані по відстані в певному просторі (компоненти стискаються за допомогою TruncatedSVD), в загальному, KNN, тим самим надаючи, хоча і не найкращі рекомендації, але ефективність обслуговування і збір інформації для моделі зі спільною фільтрацією. Після збору необхідної кількості даних на сайт буде інтегрована друга модель, яка краще виконує завдання рекомендації. У графічних матеріалах надається програмна реалізація методу рекомендацій.

3.5.4 Програмна реалізаціяметоду транскодування

Реалізовано інструмент для кодування відео, отримання інформації та конвертації в інші формати. Залежно від контенту, ви можете використовувати Frame для зображень, а Video і Subtitle відповідно для роботи з відео-файлами і субтитрами. Конструктор в загальному вигляді виглядає так:

```
public < ... >::__construct(string $filePath, ServiceFactoryInterface $service = null)
```

$ FilePath - рядок, шлях до вихідного файлу, $ Service - параметр не обов'язковий. Екземпляр сервісу. По дефолту ServiceFactory. Ви можете використовувати свою сервіс-фабрику або змінити деякі опції.

Екземпляр формату реєструє хендлери, такий хендлер буде виконаний при настанні певної події. Методи addListener або addOneListener реєструють хендлер на подію. Залежно від події, обробник може вплинути на подальший хід виконання операції.

Подія before виконується перед початком кодування. Подальше виконання може бути зупинено. При багатопрохідному кодуванні хендлер буде викликаний перед кожним проходом. Подія спрацьовує мінімум 1 раз. Подія before.pass спрацьовує після before і не може скасувати процес кодування.

Подія failure операція завершилася з помилкою або не може коректно завершитися.

Замість прямого транскодування ви можете відправляти завдання в чергу, наприклад, на сервер черг. Ви можете використовувати опцію ServiceFactoryInterface OPTION_USE_QUEUE при створенні сервіс-фабрики.

Фільтри необхідні для зміни вихідного медіа контенту. Можуть мати один або кілька входів і виходів, а також бути організовані в ланцюжки для ізоляції деяких фільтрів один від одного.

Фільтри зображень реалізують інтерфейс FrameFilterInterface і можуть використовуватися спільно тільки з Foto.

Відео фільтри реалізують інтерфейси FrameFilterInterface або VideoFilterInterface, використовуються або з Frame або з Video. У додатку надається програмна реалізація методу транскодування.

ВИСНОВКИ

В рамках виконання атестаційної роботи були розроблені та впроваджені методи рекомендацій та транскодування на для інформаційно-аналітичної системи електронних продажів.

Під час аналізу предметної області була визначена сфера застосування інформаційно-аналітичних систем в електронній комерції, проведено аналіз вже існуючих інформаційно-аналітичних систем, а також визначено специфічні особливості для предметної області, що розглядається. В результаті проведеного аналізу була розроблена постановка задачі.

Для визначення того, який підхід найбільше підходить для розробки методу рекомендацій, було проведено аналіз та дослідження наступних існуючих методів побудови рекомендацій: фільтрації на основі контексту/вмісту, колаборативна фільтрація, гібридна фільтрація. В результаті дослідження було визначено, що використання методу колаборативної фільтрації є найбільш доцільним для специфіки розглянутої предметної області.

Для розробки методів інформаційно-аналітичної системи з урахуванням специфіки предметної області був застосований алгоритм cinematch. Розроблений метод опирається на особливості предметної області. Після розробки методу були описані основні бізнес-процеси та визначені функціональні вимоги до інформаційної завдяки методології IDEF0.

Для визначення того, які інформаційні потоки будуть проходити між системою та зовнішніми сутностями була розроблена модель потоків даних інформаційно-аналітичної системи системи, а для визначення взаємодії користувачів з системою – Use Case діаграма. Під час розробки інформаційного забезпечення був проведений обґрунтований аналіз вибору СУБД та спроектована база даних. Методи були розроблені на мові програмування Java з використанням фреймворків Apache.

ПЕРЕЛІК ПОСИЛАНЬ

1. Що таке електронна комерція? [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.interkassa.com/ua/blog/chto-takoe-elektronnaya-kommerciya-e-commerce-dlya-nachinayushchih/.

2. Інформаційно-аналітична система, её компоненты, классификация [Електронний ресурс] – Режим доступу до ресурсу: http://websekretar.chizh.ua/?p=548.

3. Метрики сходства и расстояния для науки о данных и машинного обучения [Електронний ресурс] – Режим доступу до ресурсу: https://www.machinelearningmastery.ru/similarity-and-distance-metrics-for-data-science-and-machine-learning-e5121b3956f8/.

4. Item-based collaborative filtering словами [Електронний ресурс] – http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/item based.html

5. Su X., Khoshgoftaar T.M. A Survey of Collaborative Filtering Techniques; Advances in Artificial Intelligence, 2009.

6. Гончаров М. Data Mining: Рекомендательные системы [Электронный документ] (http://kek.ksu.ru/EOS/WM/50_132-670.pdf)

7. Nechyporenko, A.S. A Lars-Based Method of the Construction of a Fuzzy Regression Model for the Selection of Significant Features / Yerokhin, A.L., Babii, A.S., Nechyporenko, A.S., Turuta, O.P. // Cybernetics and Systems Analysis. – 2016. – Vol. 52 (4). – P. 641–646.

8. Нейронные сети [Електронний ресурс] – Режим доступу до ресурсу: https://future2day.ru/nejronnye-seti/.

9. Похилько А. Ф. CASE-технология моделирования процессов с использованием средств BPWin и ERWin учебное пособие / А. Ф. Похилько, И. В. Горбачев. – Ульяновск: УлГТУ, 2008. – 120 с.

10. Стандарти IDEF. Методології SADT, DFD, BPMN, ARIS, UML. Порівняльний аналіз методологій моделювання. [Електронний ресурс] – Режим доступу до ресурсу: http://um.co.ua/8/8-16/8-162794.html.

11. Общая характеристика языка UML [Електронний ресурс] – Режим доступу до ресурсу: http://www.informicus.ru/default.aspx?SECTION=6&id=73&id=2.

12. Київська К. І. Побудова діаграми прецедентів (варіантів використання) [Електронний ресурс] / К. І. Київська. – 2018. – Режим доступу до ресурсу: http://org2.knuba.edu.ua/pluginfile.php/58191/mod_resource/content/1/2018metoda_labr_1.html.

13. PostgreSQL [Електронний ресурс]. – 1. – Режим доступу до ресурсу: https://coderlessons.com/tutorials/bazy-dannykh/uchebnik-postgresql/1-chto-takoe-postgresql.

14. Nechiporenko, A.S. Solution to the model problem of excitation of loaded conic slot antenna by method of singular integral equations / Doroshenko, V.A., Ievleva, S.N., Klimova, N.P., Nechiporenko, A.S., Strelnitsky, A.A. // Telecommunications and Radio Engineering (English translation of Elektrosvyaz and Radiotekhnika). – Vol. 75. – P. 1799–1812.