НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Кафедра автоматика та управління в технічних системах
(повна назва кафедри)

«На правах рукопису»
УДК _004.89_

«До захисту допущено»

Завідувач кафедри
_____ _О. І. Ролік_
(підпис) (ініціали, прізвище)
“___”_____2018 р.

Магістерська дисертація

зі спеціальності (спеціалізації) 126 «Інформаційні системи та технології»
(код і назва спеціальності)

на тему:___Інтелектуальна система підбору клієнтського контенту_____

_____

_____

Виконав: студент __6__ курсу, групи____ІА-73мп_____
(шифр групи)

_____Дячук Іван Сергійович_____ _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник_доц. каф. АУТС, к.т.н., доц. Дорогий Я. Ю._ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант_____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _доцент кафедри ІБ, к.т.н., доц. Демчинський В. В._ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

Національний технічний університет України
“Київський політехнічний інститут
імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки
(повна назва)
Кафедра автоматики та управління в технічних системах
(повна назва)
Ступінь вищої освіти –другий (магістерський)
(код, назва)
Спеціальність 126 «Інформаційні системи та технології»
(код, назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О. І. РОЛІК
(підпис) (ініціали, прізвище)

“___”_____2018_р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Дячуку Івану Сергійовичу

(прізвище, ім’я, по батькові)

1.**Тема дисертації** Інтелектуальна система підбору клієнтського контенту

Науковий керівник дисертації Дорогий Ярослав Юрійович, к.т.н., доцент

затверджені наказом по університету від “ 29 ” жовтня 2018 р.№ _____

2.Строк подання студентом дисертації “ 4 “ грудня 2018 р.

3. Об`єкт дослідження: рекомендаційні системи

4.Зміст пояснювальної записки**:** а) огляд рекомендаційних систем; б) розробка алгоритму, математичних методів та моделей; в) розробка технічного рішення, архітектури, проектування та реалізація; г) дослідження та тестування; д) розробка стартап-проекту.

5. Перелік завдань, які потрібно розробити: проаналізувати існуючі рішення для побудови рекомендаційних систем, проаналізувати методи фільтрації контенту, розробити модель прогнозування, розробити архітектуру системи підбору клієнтського контенту, провести дослідження ефективності розроблених моделей,

протестувати розроблену систему, провести маркетинговий аналіз стартап-проекту.

6. Перелік графічного (ілюстративного) матеріалу: структурна схема моделі на базі нейронних мереж, структурна схема моделі на базі матричного розкладу, структурна схема об'єднаної моделі, діаграма послідовності 1, діаграма послідовності 2, діаграма сценаріїв використання, структурна схема системи, діаграма розгортання.

7. Орієнтовний перелік публікацій:_____

_____

8. Консультанти розділів проекту:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

9. Дата видачі завдання "_29_" __жовтня__2018_р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів проекту	Примітка
1	Огляд існуючих рішень	10.11.2018	
2	Огляд та аналіз математичних моделей	15.11.2018	
3	Розробка математичних моделей та дослідження	20.11.2018	
4	Розробка структури системи	25.11.2018	
5	Розробка стартап-проекту	29.11.2018	
6	Оформлення текстової та графічної документації	2.12.2018	
7	Представлення до захисту	4.12.2018	

Студент _____ Дячук І.С
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Дорогий Я.Ю.
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 114 с., 21 рис., 40 табл., 9 додатків, 15 джерел.

Тема магістерської дисертації «Інтелектуальна система підбору клієнтського контенту».

Підґрунтям актуальності теми даної магістерської дисертації є швидкі темпи впровадження компаніями різного роду у свої системи управління відносинами з клієнтами, а також для рекомендацій різного роду контенту на інтернет платформах, яких дедалі стає все більше, рішень на базі алгоритмів машинного навчання, що зможуть спрогнозувати поведінку користувача для того, щоб надавати більш якісний сервіс.

Об'єктом дослідження є рекомендаційні системи.

Предметом дослідження є алгоритми прогнозування, способи їх комбінування та ефективність.

Метою роботи є підвищити ефективність персоналізації та точність рекомендацій та запропонувати підхід для вибору алгоритмів та їх комбінування при вирішенні проблем рекомендацій.

Результатом виконання даної кваліфікаційної роботи є запропонована модель алгоритму нейронної мережі для прогнозування клієнтських взаємодій, що враховує історичні дані минулих взаємодій користувача з групами елементів, а також програмний комплекс – інтелектуальна система підбору клієнтського контенту. Результати розробок та досліджень роботи були використані при розробці системи, що проваджена в експлуатацію, що підтверджує практичне значення одержаних результатів.

НЕЙРОННІ МЕРЕЖІ, ПРОГНОЗУВАННЯ, РЕКОМЕНДАЦІЇ, МАШИННЕ НАВЧАННЯ, КОНТЕНТ.

ABSTRACT

Master's thesis: 114 p., 21 figures, 40 tables, 9 appendixes, 15 sources.

Theme of the master's thesis “Intelligent system of selecting client content”.

The reason for the relevance of the topic of this master's thesis is the rapid pace of implementing solutions based on machine learning algorithms that can predict user behavior in order to provide a better service. These are getting implemented by companies for their customer relationship management systems, as well as for the recommendations of various kinds of content on the Internet platforms, amount of which is increasing.

The object of research: recommendation systems.

The subject of the research are the prediction algorithms, methods of their combination and their efficiency.

The aim of the work is to increase the effectiveness of personalization and accuracy of recommendations and to offer an approach for choosing algorithms and combining them in solving recommendations problems.

The result of this qualification work is the proposed model of the algorithm for the neural network to predict client interactions, which takes into account historical data of past user interactions with the groups of elements, as well as the software system – an intellectual system of selecting client content. The results of development and research work were used in the development of the system being put into operation, which confirms the practical value of the results.

NEURAL NETWORKS, PREDICTION, RECOMMENDATIONS, MACHINE LEARNING, CONTENT.

Зміст

Перелік умовних позначень, одиниць, скорочень і термінів

API – Application Programming Interface

CRM – Customer Relationship Management

DCG – Discounted Cumulative Gain

FFNN – Feed Forward Neural Network

GRU – Gated Recurrent Unit

HR – Hit Ratio

LSTM – Long Short Term Memory

MF – Matrix Factorization

NDCG – Normalized Discounted Cumulative Gain

NN – Neural Network

ReLU – Rectified Linear Unit

RNN – Recurrent Neural Networks

SQL – Structured Query Language

SVD – Singular Value Decomposition

ВСТУП

Інтернет вже давно перестав бути для людей чужим та ворожим, зараз вони дуже близько познайомилися і, хоча, раніше знайомство відбувалося здебільшого зі сторони користувачів: вони дізнавалися, як користуватися пошуковими системами, що можна знайти в інтернеті, що придбати, які можливості надає мережа, як користуватися певними ресурсами, обирали, які їм найбільш до вподоби та зручніші тощо, то тепер вже знайомство відбувається з обох сторін, і інтернет починає дізнаватися дедалі більше про самих користувачів, щоб зробити їх спільне проведення часу ще приємнішим та продуктивнішим.

Жага до знань у інтернету (інтернет ресурсів: сайтів, веб сервісів, банків, онлайн магазинів, соціальних мереж, стримінгових сервісів тощо) надалі зростає. Збір даних про користувачів та їх подальший аналіз з метою використання для покращення продажів, привернення та утримання користувачів, покращення якості сервісів стає новою задачею для будь-якого сучасного бізнесу, що прагне досягнути успіху та вижити в умовах жорсткої конкуренції.

Рекомендаційні системи – програми, які намагаються передбачити, які об'єкти (фільми, музика, книги, новини, веб-сайти) будуть цікаві користувачеві, маючи певну інформацію про його профілі. Сьогодні в процесі переходу бізнесу в онлайн режим, постає питання взаємодії систем з користувачем та адаптацію під його поведінку, вподобання інтереси. В офлайні обирати товар чи послуги людям зазвичай допомагають продавці, але коли тисячі людей взаємодіють з комп’ютером сидячи у себе вдома ситуація змінюється. Тепер компанії мають адаптуватися під поведінку людей в інтернеті аби зробити процес користування сервісами більш приємним та корисним. На сьогодні все більше компаній намагаються впровадити системи рекомендацій у свої сервіси, це дозволяє з більшою вірогідністю утримати клієнта та здійснити продаж. Використання систем рекомендацій виправдане для будь-яких компаній, що пропонують товари чи послуги, а також для сервісів, що пропонують платформи для розповсюдження реклами. Можливість таких систем збільшити прибутки компанії та лояльність

клієнтів, зростаючий попит на системи цього призначення, а також наявність невирішених проблем та задач зумовлюють актуальність досліджень у зазначеному напрямку.

Метою роботи є підвищити ефективність персоналізації та точність рекомендацій та запропонувати підхід для вибору алгоритмів та їх комбінування при вирішенні проблем рекомендацій.

Задачі. Для досягнення поставленої мети були визначені наступні завдання:

- дослідити існуючі алгоритми прогнозування клієнтських взаємодій;
- дослідити існуючі підходи фільтрування контенту в рекомендаційних системах;
- розробити комбіновану математичну модель для прогнозування користувацької взаємодії, що враховує історичні дані;
- розробити систему підбору клієнтського контенту на базі запропонованої моделі;
- дослідити ефективність запропонованого рішення.

Об'єктом дослідження є рекомендаційні та таргетингові системи.

Предметом дослідження є алгоритми прогнозування, способи їх комбінування та ефективність.

1 РЕКОМЕНДАЦІЙНІ СИСТЕМИ

1.1 Загальні положення

1.1.1 Визначення та застосування

Рекомендуючі або рекомендаційні системи (іноді платформи або двигуни) є підкласом систем інформаційної фільтрації, які намагаються передбачити "рейтинг" або "перевагу", яку користувач надасть предмету. Вони вирішують проблему перевантаження інформації, фільтруючи фрагменти важливої інформації з великої кількості інформації, обсяги якої динамічно зростають, відповідно до переваг, інтересів або поведінки користувачів. Такі системи мають можливість прогнозувати, чи надасть перевагу певний користувач даному елементу, базуючись на профілі користувача.

Системи рекомендацій корисні як постачальникам послуг, так і користувачам. Вони зменшують транзакційні витрати на пошук і відбір елементів в онлайн-магазині. Рекомендаційні системи також довели, що вони покращують процес прийняття рішень та якість. В електронній комерції системи рекомендацій збільшують доходи, оскільки вони є ефективним засобом продажу більшої кількості продуктів.

Системи рекомендацій використовуються в різних областях, пропонуючи фільми, музику, новини, книги, наукові статті, пошукові запити, соціальні теги та продукти в цілому. Є також рекомендаційні системи для фахівців, співавторів, жартів, ресторанів, одягу, фінансових послуг, страхування життя, романтичних партнерів (онлайн-знайомств) та сторінок Twitter та інших соціальних мереж.

Система рекомендацій будує стратегію прийняття рішень для користувачів у складних інформаційних середовищах. Крім того, з точки зору електронної комерції – це інструмент, який допомагає користувачам здійснювати пошук за запитами, які пов'язані з інтересом та перевагою користувачів. Рекомендаційні системи вирішують проблему перевантаження інформацією, з якою користувачі зазвичай стикаються, надаючи їм індивідуальний, винятковий контент та

рекомендації щодо послуг. Останнім часом були розроблені різні підходи для побудови систем рекомендацій, які можуть використовувати як спільну фільтрацію (collaborative filtering), фільтрування на основі змісту (content-based filtering) чи гібридну фільтрацію. Ці підходи будуть детальніше розглянуті в наступному підрозділі 1.2.

1.1.2 Стадії процесу надання рекомендацій.

До моменту отримання користувачем рекомендацій у системі відбувається певний процес, який можна розбити на 3 основні фази:

— Збір інформації;

— Тренування системи;

— Прогнозування та надання рекомендацій.

Основні фази також відображені на рисунку 1.1.

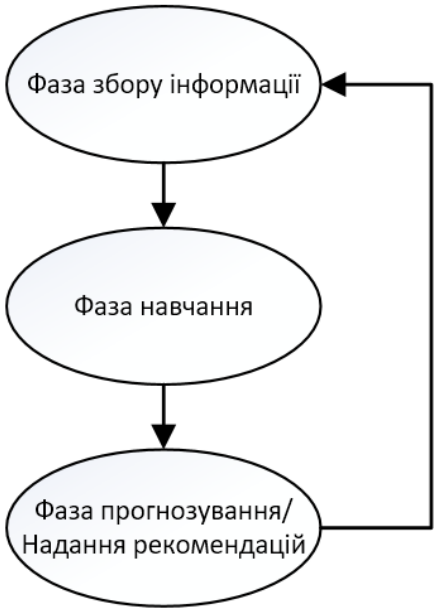


Рисунок 1.1 – Фази процесу рекомендацій

На протязі першої фази необхідно зібрати релевантну інформацію про користувачів для створення профілю або моделі користувача для майбутнього прогнозування. Така інформація включає атрибути користувача, його поведінку чи вміст ресурсів, до яких звертається користувач. Система не зможе надавати точні рекомендації, доки профіль/модель користувача не будуть добре

сконструйовані. Так званий рекомендаційний агент, яким виступає система, має знати якомога більше про користувача з самого початку, щоб забезпечити прийнятну точність рекомендацій.

Системи рекомендацій покладаються на різні типи вхідних даних, які зазвичай поділяють на явний та неявний зворотні зв'язки.

Явний зворотній зв'язок є найбільш зручним та високоякісним, з точки зору впливу на точність рекомендацій. Він являє собою безпосереднє введення користувачами даних, щодо їхньої зацікавленості чи вподобання певного елемента. Система зазвичай пропонує користувачеві через системний інтерфейс надавати оцінки для предметів для побудови та вдосконалення своєї моделі. Точність рекомендації залежить від кількості оцінок, наданих користувачем. Єдиним недоліком цього методу є те, що він потребує зусиль від користувачів, а також, користувачі не завжди готові надати достатньо інформації. Не зважаючи на те, що явний зворотній зв'язок вимагає від користувача більших зусиль, він розглядається як постачальник більш надійних даних, оскільки не передбачає виведення користувацьких переваг від дій, а також забезпечує прозорість у процесі рекомендацій, що призводить до дещо легшого сприймання рекомендацій та більшої впевненості в них.

Неявний зворотній зв'язок будується шляхом опосередкованого виведення користувацьких уподобань за допомогою спостереження за їх поведінкою та діями, такими як історія покупок, історія навігації та час, витрачений на деякі веб-сторінки, посилання, за якими слідують користувачі, зміст електронних листів, кліки клавіш та інше. Неявний зворотній зв'язок знижує навантаження на користувачів, виводячи їхні уподобання виходячи з їх взаємодії із системою. Хоча цей метод не вимагає зусиль від користувача, але він є менш точним. З іншої сторони, неявний зворотній зв’язок можна вважати більш об’єктивним, якщо взяти до уваги відсутність упередження, яке виникає у користувачів, які схильні реагувати соціально бажаним способом, для них в такому випадку не виникає жодних проблем із самообмеженням та збереженням образу для інших [1].

Можна зустріти системи, що використовують гібридний зворотний зв'язок,

що є поєднанням явного та неявного зворотного зв'язку. Сильні сторони як неявного, так і явного зворотного зв'язку можуть бути об'єднані в гібридну систему, щоб мінімізувати їх слабкі сторони та отримати найбільш ефективну систему. Цього можна досягти, використовуючи неявні дані як перевірку явно наданого рейтингу предметів, або дозволяючи користувачеві давати явний відгук лише тоді, коли він сам вирішує виказати явний інтерес.

Профіль користувача описує його просту модель. Успіх будь-якої системи рекомендацій багато в чому залежить від його здатності представляти поточні інтереси користувача. Незалежно від методів прогнозування для отримання відповідних та точних рекомендацій необхідні точні моделі.

На другій фазі – фазі навчання, застосовуються алгоритми навчання для фільтрації та експлуатації ознак користувача та зворотного зв'язку, отриманого на етапі збору інформації.

На останній стадії система рекомендує/передбачає, яким саме елементам користувач може віддати перевагу.

1.2 Підходи до вирішення задачі рекомендацій

Використання ефективних та правильних методів є дуже важливим для системи, що ставить своєю метою надавати точні та корисні рекомендації користувачам. Це пояснює важливість розуміння особливостей та можливостей різних методів рекомендацій. На рисунку 1.2.1 зображена топологія різних методів фільтрації рекомендацій.

Рисунок 1.2 – Підходи та техніки вирішення задачі рекомендацій (2-й рівень – підходи, нижче – техніки)

1.2.1 Фільтрування на базі змісту

Алгоритм фільтрування на базі змісту залежить від домену, і більше спирається на аналіз атрибутів елементів для побудови прогнозів. Коли рекомендуються такі елементи, як веб-сторінки, публікації чи новини, ця технологія є найбільш успішною. Фільтрування на базі вмісту будує прогнози використовуючи профілі користувачів, видобувши ознаки з елементів, які користувач раніше оцінював. Користувачеві рекомендуються ті елементи, що найбільш схожі з тими, які він позитивно оцінив. У фільтруванні на базі змісту для знаходження схожих елементів та створення корисних рекомендацій можуть використовуватися різні типи моделей, наприклад, моделі векторного простору, таку як Frequency Inverse Document Frequency (TF/IDF) або імовірнісні моделі, як наївний баєсів класифікатор, дерева рішень або нейронні мережі, що здатні змоделювати зв'язки між різними елементами. Ці методи надають рекомендації, вивчаючи базову модель із використанням статистичного аналізу або методами машинного навчання. При цьому для рекомендацій одному користувачеві профілі інших користувачів не використовуються. Також, метод фільтрація на базі змісту дає змогу корегувати рекомендації протягом дуже короткого часу, якщо профіль

користувача зміниться. Основним недоліком цього методу є необхідність в наявності детальних характеристик та глибоких знань про елементи.

Розгляньмо недоліки та переваги фільтрації на базі змісту. Методи фільтрації на базі змісту справляються з задачами краще за спільну фільтрацію. Вони мають можливість рекомендувати нові елементи, навіть якщо в базі відсутні дані про вподобання користувачів щодо цих елементів. Таким чином, навіть якщо база даних не містить, або містить небагато, інформації про переваги користувачів, це не впливає на точність рекомендацій. Також, якщо профіль користувача змінюються, система буде здатна змінити відкоригувати рекомендації за короткий проміжок часу. Фільтрація на базі змісту справляється з ситуаціями, коли користувачі не вподобали ті самі елементи, а лише ідентичні відповідно до їх властивостей та ознак. Користувачі можуть отримувати рекомендації не розділяючи доступ до свого профілю, і це гарантує конфіденційність. Техніка на базі змісту також дає можливість надати пояснення користувачеві щодо того, як були створенні його персональні рекомендації, що збільшує його лояльність до цих рекомендацій. Проте цей метод має й недоліки. Присутня залежність від метаданих елементів. Тобто необхідній детальний глибокий опис елементів і дуже добре сформований профіль користувача, перш ніж система здатна буде робити точні рекомендації. Це отримало назву обмеженого аналізу контенту. Отже, ефективність методів на базі змісту залежить від наявності описових даних. Інша серйозна проблема цього методу є занадто велика спеціалізація за змістом: рекомендації обмежуються лише подібними елементами до тих елементів, що раніше були відзначені користувачем.

1.2.2 Спільна фільтрація

Спільна або колаборативна фільтрація – підхід незалежний від домену, добре підходить для роботи з контентом, який неможливо легко та адекватно описати метаданими, наприклад, фільми та музика. Технологія спільної фільтрації створює базу даних (матрицю користувачі-елементи) в якій відображаються вподобання елементів користувачами. Далі знаходяться користувачі зі схожими інтересами та перевагами та робляться відповідні рекомендації. Такі схожі

користувачі об’єднуються групу під назвою «сусідство» («neighborhood»). Користувач отримує рекомендації щодо тих предметів, які він ще не встиг оцінити, але які вже були позитивно оцінені користувачами сусідами. Рекомендації на виході спільної фільтрації, можуть бути як у вигляді передбачень, так і у вигляді рекомендацій. Передбачення – це чисельне значення R_{ij}, яке відображає прогнозовану оцінку користувача i елементу j, тоді як рекомендація – це список N елементів, які мають найвищу вірогідність сподобатись користувачеві. Метод спільної фільтрації може здійснюватися двома способами: на основі пам'яті та на основі моделі.

1.2.2.1 Спільна фільтрація на основі пам'яті

Елементи, які користувач вже встиг оцінити раніше відіграють важливу роль у пошуку сусідів, які поділяє з ним ті самі смаки. Як сусід користувача буде знайдений, можна використовувати різні алгоритми поєднання смаків сусідів для створення рекомендацій. Методи спільної фільтрації на основі пам'яті завдяки своїй ефективності досягли значного успіху та широкого впровадження в реальні системи. Спільна фільтрація на основі пам'яті в свою чергу також може реалізовуватися двома способами: відштовхуючись від користувачів або від елементів. Перший спосіб розраховує подібність між користувачами, порівнюючи їх оцінки відносно одного і того ж елемента, а потім обчислює вірогідну оцінку користувача конкретного елемента як середнє значення оцінок елемента користувачами-сусідами, де вагами є схожість користувачів. Методи фільтрації, що відштовхуються від елементів надають свої прогнози використовуючи подібність елементів, а не подібність користувачів, створюючи модель подібності елементів на базі всіх елементів, оцінених користувачем, тобто визначається, наскільки подібні інші елементи до цільового елемента, а потім обирається k найбільш подібних елементів і обчислюється їх подібність. Прогноз здійснюється шляхом прийняття середньозваженого середнього рейтингу користувача цих подібних елементів k.

Для обчислення подібності між елементом/користувачем існує декілька рішень. Два найпопулярніших – кореляційні та косинусні. Коефіцієнт кореляції

Пірсона використовується для вимірювання ступеня лінійного відношення двох змінних і визначається як:

$$s(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i}-\overline{r_a})(r_{u,i}-\overline{r_u})}{\sqrt{\sum_{i=1}^{n}(r_{a,i}-\overline{r_a})^2}\sqrt{\sum_{i=1}^{n}(r_{u,i}-\overline{r_u})^2}} \quad . \qquad (1.1)$$

У вищевказаному рівнянні *s(a,u)* – відображає схожість між двома користувачами *a* та *u*, $r_{a,i}$ – це оцінка елементу *i* користувача *a*, $\overline{r_a}$ - це середня оцінка користувача по всіх елементах, а *n* – загальна кількість елементів у просторі користувач-елемент. Передбачення оцінки елемента складається з вагової комбінації оцінок обраних сусідів, що обчислюється як зважене відхилення від середнього значення сусідів. Загальний вигляд формули передбачення:

$$p(a,i) = \overline{r_a} + \frac{\sum_{i=1}^{n}(r_{u,i}-\overline{r_u})\times s(a,u)}{\sum_{i=1}^{n} s(a,u)} \quad . \qquad (1.2)$$

Косинусна подібність відрізняється від коефіцієнту Пірсона, оскільки вона є векторно-просторовою моделлю, яка базується на лінійній алгебрі, а не на статистичному підході. Косинусна подібність вимірює подібність двох n-мірних векторів на основі кута між ними. Міра подібності на основі косинів широко використовується в області аналізу інформації та аналізу текстів для порівняння двох текстових документів, де документи представлені векторами термінів. Схожість між двома елементами *u* та *v* можна визначити наступним чином:

$$s(\overrightarrow{u},\overrightarrow{v}) = \frac{\overrightarrow{u}\cdot\overrightarrow{v}}{|\overrightarrow{u}|*|\overrightarrow{v}|} = \frac{\sum_i r_{u,i}r_{v,i}}{\sqrt{\sum_i r_{u,i}^2}\times\sqrt{\sum_i r_{v,i}^2}} \quad . \qquad (1.3)$$

Міра подібності також називається метрикою подібності, і є методом, що використовуються для розрахунку оцінки, що виражає, наскільки подібні користувачі або елементи тісно пов’язані та відносяться один до одного. Ця оцінка потім може бути використана в якості основи генерації рекомендацій. Залежно від контексту використання метрику подібності можна також називати метрикою кореляції або метрикою відстані.

1.2.2.2 Спільна фільтрація на основі моделі

Ця методика використовує попередні рейтинги, щоб навчити модель, для підвищення ефективності спільної фільтрації. Побудова моделі може бути виконана з допомогою машинного навчання або методів вилучення даних (data-mining). Методи фільтрації на основі моделі можуть швидко генерувати набір рекомендацій завдяки тому, що вони використовують попередньо натреновану модель. Ці методи включають в себе методи зменшення розмірності, такі як сингулярний розклад матриці (SVD), Matrix Completion Technique, латентно-семантичний аналіз а також регресію та кластеризацію. Методи, що базуються на моделях, аналізують матрицю користувацького елемента для визначення зв'язків між елементами; вони використовують ці відносини, щоб порівнювати список найкращих рекомендацій. Методи, засновані на технічних рішеннях, вирішують проблеми ранжування, пов'язані з системами рекомендацій. Використання алгоритмів навчання також іноді змінюють манеру рекомендацій: із рекомендацій «що споживати» на рекомендації «коли споживати».

1.2.2.3 Недоліки та переваги спільної фільтрації

Спільна фільтрація має деякі переваги над фільтрацією на базі змісту, оскільки вона може виконуватись у доменах, в яких важко зібрати характеристики елементів, або їх вміст важко аналізувати (наприклад, думки та ідеали). Крім того, технологія спільної фільтрації дає змогу надавати випадкові рекомендації, тобто такі що рекомендовані елементи, релевантні для користувача, хоча в профілі користувача не знаходиться нічого, щоб його з ними пов’язувало б. Незважаючи на успіх методів спільної фільтрації, при їх широкому використанні було виявлено деякі потенційні проблеми.

1.2.2.4 Проблема холодного старту

Це стосується ситуації, коли система рекомендацій не має адекватної інформації про користувачів чи елементи, щоб зробити відповідні прогнози. Це одна з основних проблем, що знижують ефективність систем рекомендацій. Профілі нових користувачів або елементів будуть порожніми, оскільки оцінки від користувачів ще не встигли з’явитися – отже, їх смаки невідомі системі.

1.2.2.5 Проблема розріджених даних

Це проблема, яка виникає внаслідок відсутності достатньої інформації, тобто коли лише деякі з загальної кількості елементів, доступних у базі даних, були оцінені користувачами. Це призводить до розрідженої матриці користувач-елемент, нездатності вдало знайти сусідів і, нарешті, до створення поганих рекомендацій. Також, розрідженість даних завжди призводить до проблеми охоплення, що виражається у малому до відсотку елементів у системі, які можна вносити до рекомендацій.

1.2.2.6 Проблема масштабування

Це ще одна проблема, пов'язана з алгоритмами рекомендацій, що виникає оскільки обчислення зазвичай зростає лінійно разом з кількістю користувачів і елементів. Також метод рекомендацій, що є ефективним, коли кількість даних одна, може бути не в змозі надати задовільну кількість та якість рекомендацій при збільшенні обсягу набору даних. Таким чином, важливо застосовувати методи рекомендацій, які можуть успішно масштабуватися, оскільки кількість набору даних у базі даних скоріше за все буде зростати. Методи, що використовуються для вирішення проблеми масштабованості та прискорення генерації рекомендацій, базуються на методах зменшення розмірності, наприклад, метод сингулярного розкладу матриць (SVD), що здатний надавати надійні та ефективні рекомендації.

1.2.2.7 Проблема синонімії

Синонімія – це тенденція коли дуже подібні предмети, мають різні назви чи ідентифікатори. Більшість систем рекомендацій мають складнощі із розмежуванням між тісно пов'язаними предметами, такими як, наприклад, «дитячий одяг» та «дитяче вбрання». Спільна фільтрація зазвичай не знаходить відповідності між двома термінами, аби виявити їх схожість. Різні методи, такі як побудова тезаурусу та сингулярний розклад матриць (SVD), латентно-семантичний аналіз здатні вирішити проблему синонімії. Недоліком цих методів є те, що деякі додані терміни можуть мати відмінні значення від того, що зазначено, що іноді призводить до швидкої деградації якості рекомендацій.

1.2.3 Гібридна фільтрація

Гібридна фільтрації поєднує в собі різні методи рекомендацій для кращої оптимізації системи, та щоб уникнути деяких проблем і обмежень самостійних технологій систем рекомендацій. Ідея гібридної фільтрації полягає в тому, що створюється комбінація алгоритмів що забезпечує більш точні та ефективні рекомендації, ніж єдиний алгоритм, оскільки недоліки одного алгоритму долаються іншим алгоритмом. Використання кількох методів рекомендацій може прикрити слабкі сторони індивідуальної техніки в комбінованій моделі. Поєднання підходів можна виконувати будь-яким із таких способів як: окрема реалізація алгоритмів та об'єднання їх результатів, використання окремих фільтрів на основі вмісту у спільній фільтрації, чи навпаки використання окремих фільтрів спільної фільтрації у підході на базі змісту, або ж створити єдину систему рекомендацій, яка використовує одночасно обидва підходи.

1.2.3.1 Зважена гібридизація

Зважена гібридизація поєднує результати різних рекомендацій для створення списків рекомендацій або прогнозування шляхом інтегрування балів кожного з методів, що використовуються, лінійною формулою. Спочатку прогнози обох методів отримують рівну вагу, але ваги налаштовуються, в залежності від того які прогнози підтверджуються. Перевага зваженої гібридизації полягає в тому, що протягом процесу всі сильні сторони системи рекомендацій використовуються прямолінійним чином.

1.2.3.2 Гібридизація з перемиканням

Система здійснює перемикання методів рекомендацій за евристикою, що відображає спроможність робити хороші рекомендації. Гібридизація з перемиканням має можливість уникати проблем, специфічних для одного способу, наприклад проблему нового користувача для рекомендацій на основі вмісту, перейшовши систему рекомендацій на спільній фільтрації. Перевага цієї стратегії полягає у здобуванні системою чутливості до сильних і слабких сторін обраних методів рекомендацій. Основним недоліком перемикання є те, що це зазвичай призводить до ускладнення процесу рекомендації, тому що повинен бути

визначений критерій перемикання, який зазвичай вимагає більшої кількості параметрів системи рекомендацій. Наприклад, спочатку можуть використовуватися рекомендації на базі змісту, коли система ще не готова робити гарні рекомендації на базі спільної фільтрації, а коли система отримує достатню кількість відгуків, то відбудеться перемикання на спільну фільтрацію.

1.2.3.3 Каскадна гібридизація

Каскадний метод гібридизації застосовує ітеративний процес уточнення. Рекомендації однієї методики вдосконалюються іншою методикою. Перша методика рекомендації видає перелік грубих рекомендацій, які у свою чергу уточнюються та вдосконалюються наступною методикою рекомендацій. Технологія гібридизації дуже ефективна і толерантна до шуму через її характер ітерації «від грубого до точного».

1.2.3.4 Змішана гібридизація

Змішана гібридизація поєднує результати рекомендацій різних методів одночасно, замість того, щоб мати лише одну рекомендацію. Кожен елемент має кілька прогнозованих оцінок, пов'язаних з ним, що надійшли від різних методів рекомендацій. У змішаній гібридизації продуктивність індивідуальних методів не завжди впливають на загальну ефективність.

1.2.3.5 Гібридизація з комбінацією ознак

Ознаки обраховані, однією методикою рекомендацій, потрапляють в іншу як вхідні данні. Наприклад, оцінка подібності користувачів, що є ознакою спільної фільтрації, використовується в техніці рекомендації на базі змісту при зважені оцінок між подібними елементами. Перевагою цієї методики є те, що вона не завжди залежить виключно від даних стосовно відношення між користувачами.

1.2.3.6 Гібридизація з доповненням ознак

Ця методика має туж саму стратегію, що й гібридизація з комбінацією ознак, але також вимагає від кожної наступної частини системи доповнення масиву ознак новими. Гібридизація з доповненням ознак примножує переваги гібридизації з комбінацією ознак, оскільки ще додається невелика кількість ознак до основного набору.

1.2.3.6 Гібридизація на мета-рівні

Внутрішня модель, створена за допомогою однієї методики рекомендацій, використовується як вхідна для іншої. Зроблена модель завжди багатша з точки зору інформації. Гібриди мета-рівня здатні вирішити проблему розрідженості, що іноді виникає у методі спільної фільтрації, використовуючи модель, отриману за допомогою першої методики, як вхідну для другої методики.

1.3 Математичні методи та моделі

1.3.1 Розклад матриць (Matrix factorization)

Розклад матриці – це клас алгоритмів спільної фільтрації, що серед іншого також використовується і в системах рекомендації. Алгоритми розкладу матриць працюють шляхом розбиття матриці взаємодії користувача та елемента (матриця користувач-елемент) на добуток двох прямокутних матриць меншого розміру. Сімейство цих методів стало широко відомим після змагань від компанії Netflix, під час яких показало свою ефективність [2]. Починаючи від моменту публікації роботи Функа в 2006 році [2], почали з’являтися безліч підходів на базі матричних розкладів для систем рекомендацій.

Ідея матричного розкладу полягає в тому, щоб представити користувачів та елементи у латентному просторі ознак меншої розмірності.

При використанні підходу матричного розкладу, зазвичай кожен користувач і елемент пов'язується з вектором дійсних чисел, що представляють латентні ознаки. Нехай, p_u і q_i позначають латентний вектор користувача u та елемента i відповідно, тоді метод матричного розкладу оцінює взаємодію y_{ui} як скалярний добуток p_u і q_i:

$$y_{ui} = \mathrm{f}(\mathrm{u}, \mathrm{i}|p_u, q_i) = {p_u}^T q_i = \sum_{k=1}^{K} p_{uk} q_{ik}, \qquad (1.4)$$

де K відображає розмірність латентного простору. Отже, підхід матричного розкладу моделює двосторонню взаємодію прихованих ознак користувача та елемента, беручи до уваги, що кожен параметр прихованого простору є незалежним один від одного, і лінійно поєднує їх з такою ж вагою. Таким чином,

матричний розклад можна розглядати як лінійну модель взаємодії латентних ознак.

	i1	*i2*	*i3*	*i4*	*i5*
u1	1	1	1	0	1
u2	0	1	1	0	0
u3	0	1	1	1	0
u4	1	0	1	1	1

Рисунок 1.3 – Матриця користувач-елемент

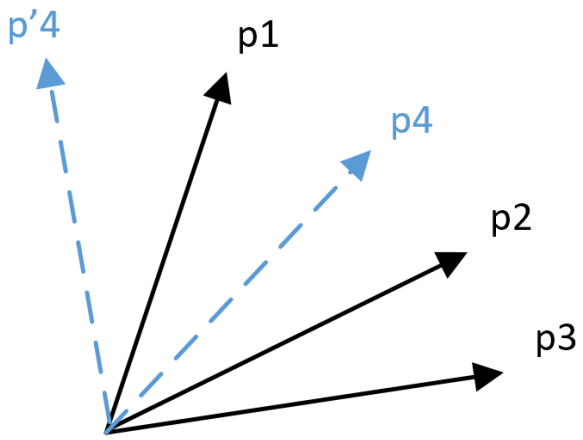


Рисунок 1.4 – Простір латентних ознак

На рисунку 1.4 видно, як функція скалярного добутку може обмежити виразність матричного розкладу. Для зрозуміння цього прикладу, потрібно взяти до уваги дві речі. По-перше, оскільки матричний розклад переносить користувачів та елементи в один і той самий латентний простір, подібність між двома користувачами може бути виміряна за допомогою скалярного добутку, або косинусом кута між їх латентними векторами (припускаючи, що це одиничні вектори). По-друге, в цьому прикладі без втрати загальності використовується коефіцієнт Джакарда, як підтвердження подібності користувачів, що мають бути відтворені матричним розкладом. Коефіцієнт Джакарда визначається як:

$$s_{ij} = \frac{|R_i| \cap |R_j|}{|R_i| \cup |R_j|}, \qquad (1.5)$$

де R_i ,R_j – це множини елементів, з якими взаємодіяли користувачі *i* та *j* відповідно. З рисунка 1.3 видно, що $s_{23}(0,66) > s_{12}(0,5) > s_{13}(0,4)$. Тоді геометричне відношення латентних векторів користувачів p_1, p_2 та p_3 може бути зображене так, як зображено на рисунку 1.4. Тепер також можна сказати, що $s_{41}(0,6) > s_{43}(0,4) > s_{42}(0,2)$. Однак, якщо p_4 буде розміщений найближче до p_1, це призведе до того, що p_4 буде знаходитись ближче до p_2, ніж до p_3, що може призвести до значних відхилень при прогнозуванні.

Цей приклад вказує на можливі обмеження методу матричного розкладу для систем рекомендацій, через використання простого та фіксованого скалярного добутку для моделювання складної функції взаємодії користувача з елементом. Ця проблема може бути частково вирішена збільшенням розмірності латентного простору, проте це в свою чергу може зашкодити загальності моделі та призвести до перенавчання, особливо при високій розрідженості даних.

1.3.2 Neural networks

Нейронні мережі – це набір алгоритмів, що були спроектовані на базі уявлень про роботу людського мозку, та призначені для розпізнавання різного роду шаблонів та залежностей. Ці алгоритми обробляють та сприймають інформацію через призму своєрідного машинного сприйняття, маркування та розбиття (кластеризації) сирих вхідних даних.

Нейронні мережі допомагають при вирішенні широкого класу задач кластеризації, класифікації та прогнозування.

Завдання класифікації залежать від промаркованих наборів даних, тобто кожен об’єкт даних має містити позначку свого класу, щоб нейронна мережа могла вивчити зв'язки між класами та об’єктами даних. Такий підхід також називають навчанням з учителем. Прикладами класифікації є розпізнавання обличь, голосів, спаму тощо.

Кластеризація чи групування – це по суті виявлення подібності елементів. Ця задача не потребує маркування об’єктів даних, і такий підхід називають навчанням без учителя. До цього класу задач відносять пошук схожих документів, відео, музики тощо.

При вирішенні задачі класифікації нейронні мережі здатні виявляти залежності між, наприклад, класом об’єкту та яскравістю певних пікселів на зображенні. Таким же чином вони здатні знаходити кореляційні залежності між минулими подіями та майбутніми.

Зазвичай нейронна мережа складається з декількох рівнів. Кожен рівень в свою чергу являє собою групу штучних нейронів, в середині яких і відбувається безпосереднє обчислення. Загальна схема штучного нейрону зображена на рисунку 1.5. Кожен нейрон можна сприймати, як певний тригер, що спрацьовує при достатньо сильному стимулі, поріг спрацювання зазвичай змінюється під час навчання. Нейрони можуть мати декілька входів та виходів. Кожному входу присвоєна певна вага, що корегується під час роботи алгоритму. Можливість змінювати ваги вхідних сигналів до нейронів, а також поріг спрацювання нейрону і забезпечує здатність алгоритму до навчання.

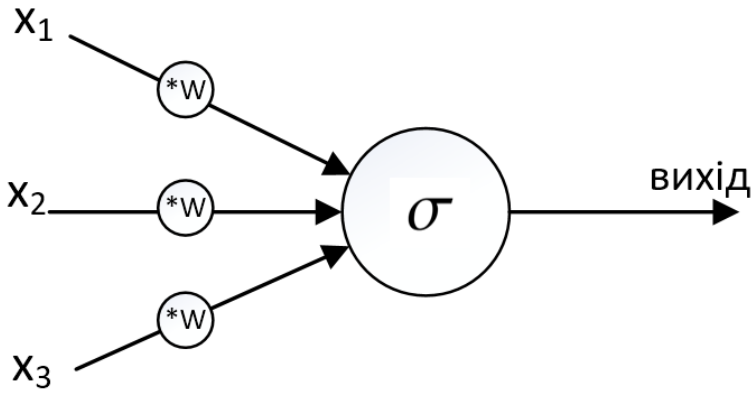


Рисунок 1.5 – Штучний нейрон.

Функцію, що визначає вихідне значення нейрона також називають активаційною функцією. Нейрон, що видає на виході лише 0 або 1, також називають перцептроном. Його активаційну функцію можна визначити наступним чином:

$$\text{вихід} = \begin{cases} 0, & \text{якщо } w \cdot x + b \leq 0 \\ 1, & \text{якщо } w \cdot x + b > 0 \end{cases}, \qquad (1.6)$$

Де b – це так званий биас, такий що b= −threshold (threshold – поріг спрацювання нейрону),а x та w – це вхідні значення та їх ваги відповідно.

Проте зараз частіше використовують сигмоїдальні нейрони [3, 4], бо бінарна природа значень значно обмежує нейронну мережу. Різниця полягає в активаційній функції. Для сигмоїдального нейрона вона буде наступною:

$$\sigma(z) = \frac{1}{1+e^{-z}}, \qquad (1.7)$$

де $z = w * x + b$. Так само як і перцептрон, сигмоїдальний нейрон має вхідні та вихідні значення, проте вони можуть набувати будь-якого значення між 0 та 1.

Навчання нейронних мереж зазвичай відбувається під час зворотнього проходу, при використанні градієнтного спуску. Маючи функцію втрат, яку необхідно мінімізувати, обчислюються градієнти та вказують в якому напрямку треба змінювати параметри нейронної мережі.

Існує багато архітектур нейронних мереж, що по різному об’єднують різні рівні різних нейронів. Нейронні мережі в залежності від їх архітектури розділяють на певні групи або класи, наприклад рекурентні або згорткові. Кожен клас архітектур може показувати різні результати на певних класах задач, тому важливо підібрати правильну архітектуру. Для отримання оптимальних результатів моделі декількох нейронних мереж об’єднують різними способами, або будують свої модифіковані архітектури.

Важливо зазначити, що на сьогоднішній день нейронні мережі, як основний засіб для побудови прогнозів та рекомендацій в системах рекомендацій, не набули широкого використання, та використовувалися здебільшого для генерації супутньої інформації, тоді як основним методом був матричній розклад.

1.4. Формування вимог до системи

Проаналізувавши існуючі рішення та підходи вирішення до задачі рекомендації контенту, можна сформувати вимоги до системи, яку можна буде розглядати, як майбутній продукт. Такий продукт має бути цікавим як для

потенційних покупців такої системи, так і для кінцевих користувачів, що будуть безпосередньо отримувати рекомендації. Тому такий продукт має бути економічно вигідно впроваджувати, та надавати корисний та зручний функціонал, що буде конкурентним з якісної та функціональної точки зору. Щоб це забезпечити потрібно виконати певні вимоги, щодо архітектурного рішення, реалізації, функціональних та нефункціональних вимог.

До функціональних вимог інтелектуальної системи підбору клієнтського контенту можна віднести наступне:

- можливість зареєстрованому користувачеві отримувати рекомендації;
- можливість зареєстрованому в системі клієнта користувачеві авторизуватися в системі підбору контенту;
- можливість зареєстрованому користувачеві переглядати інформацію про елементи, якщо клієнт надає таку інформацію;
- можливість зареєстрованому користувачеві фільтрувати підібраний контент по категоріям;
- можливість зареєстрованому користувачеві додавати в збережене елементи контенту, для майбутнього швидкого перегляду;
- можливість представнику клієнта переглядати елементи, що найчастіше потрапляли до рекомендованого контенту;
- можливість представнику клієнта фільтрувати елементи, що найчастіше потрапляли до рекомендованого контенту по категоріях.

До нефункціональних вимог інтелектуальної системи підбору клієнтського контенту можна віднести наступні пункти:

- система має легко масштабуватися;
- система має давати легко та швидко нарощувати функціонал;
- мають враховуватися історичні дані;
- архітектура системи має бути достатньо гнучкою та дозволяти впроваджувати та інтегрувати систему з різноманітними системами клієнтів з мінімальними витратами.

1.5. Висновки

У першому розділі було розглянуто рекомендаційні системи, як інтелектуальні системи підбору клієнтського контенту. Були описані основні принципи побудови та роботи таких систем. Також описано основні існуючі підходи до вирішення задачі підбору контенту та техніки, що найчастіше використовуються.

Проаналізовані проблеми, які виникають при розробці подібних систем, та можливі методи рішення.

На базі проробленого аналізу можна зробити висновок, що системи рекомендацій або підбору клієнтського контенту набувають широкої популярності серед провайдерів послуг та роздрібних торгівців в мережі, все більше розроблюється подібних систем, тому дослідження та розробки у цьому напрямку є актуальними та затребуваними.

Також можна зробити висновок, що область для досліджень ще не вичерпала себе та є широкий обсяг направлень для подальших досліджень та експериментів з різноманітними видами алгоритмів прогнозування та можливими їх комбінаціями, що можуть призвести до покращення результатів. Основним математичним методом на якому базуються сучасні системи рекомендацій є матричний розклад для спільної фільтрації, а нейронні мережі здебільшого використовуються для генерації супутньої інформації.

Зрозуміло, що системи рекомендацій та підбору контенту є індивідуальними для кожного домену та навіть бізнесу та його набору даних, адже саме від даних потрібно відштовхуватись під час розробки алгоритмів прогнозування та рекомендацій. Беручи це до уваги, були сформовані вимоги до системи, що зможе надавати потрібний функціонал компаніям клієнтам, та при цьому не втрачати своєї загальності та легко впроваджуватися поверх різних систем, з мінімальними витратами. Розрахований функціональні вимоги є мінімальними, для легшого першого контакту з клієнтами та мінімальної ціни реалізації, з розрахунком на довготривалі стосунки та подальше розширення

функціоналу, можливість для чого повинна бути прорахована в архітектурі системи.

2 РОЗРОБКА АЛГОРИТМУ, МАТЕМАТИЧНИХ МЕТОДІВ ТА МОДЕЛЕЙ

Для реалізації «Інтелектуальної системи підбору клієнтського контенту» пропонується об’єднати декілька підходів в єдине ціле для отримання оптимальних показників продуктивності та якості рекомендацій. Пропонується поєднати підхід на базі матричного розкладу з моделлю моделями на базі нейронної мережі. Основною задачею моделі в даній системі є прогнозування входження підкатегорії товарів до наступного замовлення користувача, врахувавши при цьому історичні дані.

Було обрано декілька найбільш відповідних архітектур нейронних мереж, як початкова точка. Далі буде запропоновані модифікації цих моделей та можливий спосіб їх об’єднання для побудови оптимального рішення.

2.1 Загальний фреймворк

Для побудови рішення задачі рекомендації шляхом нейронних моделей та моделювання взаємодії користувача з групою елементів буде використано багаторівневу структуру мережі.

На вхід моделі будуть подаватися ідентифікатор користувача, ідентифікатор підкатегорії елементу та інформація про декілька попередніх замовлень (фіксована кількість). Інформація про одне замовлення включатиме характеристики замовлення, а також ідентифікатори елементів, що до нього входили; також ідентифікатори всіх категоріальних ознак цих елементів. Тобто будуть вектори v_u, що представляє користувача, та декілька векторів, що представляють кожне із замовлень серед w_n по w_{n+m}. Кожен вектор замовлення в свою чергу складається з вектору елементів $v_{n_{i-s}}$, (i та s – перший та останній номери елементів замовлення відповідно) векторів категоріальних ознак та реальних ознак замовлення.

Наявність інших ознак окрім ідентифікаторів елементів протидіє проблемі спільної фільтрації – проблемі холодного старту, використовуючи ознаки змісту для представлення користувачів та елементів.

Категоріальні ознаки та ідентифікатори за допомогою вбудованих шарів мережі (embedded layers) будуть обернені у вектори фіксованої розмірності, що будуть являти собою латентні ознаки. Це повнозв'язні рівні, що перетворюють розріджене представлення, закодованих за допомогою унітарного коду, категоріальних ознак та ідентифікаторів у щільний вектор. Наприклад, отримане на виході вбудованого (embedded) рівня векторне представлення ідентифікатора користувача може розглядатися як вектор латентних ознак користувача в розрізі латентної моделі. Те саме актуальне і для категоріальних ознак замовлення та для ідентифікаторів і категоріальних ознак елементів.

Потім вектори латентних ознак будуть об'єднані з вектором реальних ознак та передані далі. Тобто будуть такі вектори: один, що описує користувача u, вектор, що описує підкатегорію та декілька векторів, що описують історію нещодавніх n замовлень користувача, тобто одразу декілька замовлень – послідовність.

Реальні ознаки разом з латентними ознаками потім йдуть на вхід до першого рівня нейронної мережі багаторівневої архітектури, що виконує роль нейронної моделі спільної фільтрації, та має перетворити вхідні ознаки на прогнози взаємодії користувача, тобто спрогнозувати вірогідність появи елементу даної категорії у наступному замовленні. Кожен рівень мережі нейронної моделі може бути налаштованим під визначення певних латентних структур взаємодії користувача.

Вихідні сигнали з нейронів останнього рівня нейронної моделі потрапляють на вихідний рівень, що видає прогнозовану оцінку взаємодії $\hat{y}_{ui}$. Навчання відбувається за рахунок мінімізації точкової втрати між $\hat{y}_{ui}$ та цільовим значенням y_{ui}. Варто зазначити, що $\hat{y}_{ui}$ це прогнозоване значення ймовірності входження підкатегорії до наступного замовлення користувача, така ж сама модель може застосовуватися для прогнозування більш та менш абстрактних категорії елементів чи навіть самих елементів, проте задля прогнозування наступних

елементів, що входитимуть до наступного замовлення, доведеться зробити кількість обчислень прогнозованої ймовірності рівну кількості існуючих елементів, що буде займати велику кількість обчислювальних ресурсів та часу.

Тепер можна сформулювати прогнозуючу модель наступним чином:

$$\hat{y}_{ui} = f(P^T v_u^U, Q^T v_{n_i}^I, Q^T v_{n_{i+1}}^J, \dots, Q^T v_{n+1_i}^I, Q^T v_{n+1_{i+1}}^J \dots, B^T w_n | P, Q, B, \dots, \theta_f), \qquad (2.1)$$

де $P \in R^{M \times K}$ та $Q \in R^{N \times K}$ відображають матрицю латентних ознак для користувачів та елементів відповідно, M та N – це кількість всіх існуючих користувачів та елементів, K – це розмірність латентного простору, $B \in R^{N \times K}$ – це матриця латентних ознак категоріальної ознаки замовлення, а θ_f позначено параметри моделі. Повна версія цієї формули мала б ще містити аналогічне представлення для кожної категоріальної ознаки елемента, та замовлення. Оскільки функція взаємодії визначена як багаторівнева нейронна мережа, то вона може бути також виражена:

$$f\left(P^T v_u^U, Q^T v_i^I\right) = \varphi_{\text{вихід}}(\varphi_X(\dots \varphi_2(\varphi_1(P^T v_u^U, Q^T v_{n_i}^I, \dots, B^T w_n \dots)) \dots)), \quad (2.2)$$

де $\varphi_{\text{вихід}}$ та φ_x це функції відображення для вихідного та x-того рівня нейронної мережі спільної фільтрації, а X – це загальна кількість нейронних рівнів у загальній моделі.

2.2 Побудова моделі нейронної мережі

Дана версія системи має на меті підбір контенту для клієнтів, що займаються роздрібною торгівлею, та врахування історичних даних про замовлення користувача (тобто взаємодію з декількома елементами одночасно – групову взаємодію), тому потрібно побудувати таку модель, що могла б враховувати попередні замовлення користувача, виявляти залежності між минулими замовленнями та наступними. Задача ускладняється можливою різною кількістю елементів у замовленні.

Пропонується розглядати поведінку користувача, як послідовність замовлень. Враховуючи це можна констатувати, те що історичні данні а також і порядок, і періодичність замовлень мають відігравати певну роль при прогнозуванні наступного замовлення користувача.

Існуючі системи рекомендацій зазвичай використовують алгоритми, що прогнозують оцінку даного користувача стосовно одного елементу за раз. Вони використовують одиничні оцінки окремих взаємодій користувача з елементом та можливо доповнюють це певними ознаками користувача та елемента. Проте такий підхід до побудови прогнозувань не зовсім відповідає природі взаємодії, що є одночасною взаємодією із багатьма елементами, тому запропоновано розглянути взаємодію користувача як комбіновану групову взаємодію, та аналізувати дані одразу, щодо всіх елементів у замовленні.

Серед чисельних варіацій архітектур нейронних мереж вибір було зроблено на користь рекурентних нейронних мереж, а саме двох їх варіацій мережі довгої короткочасної пам’яті – LSTM (Long Short-Term Memory) та мережі вентильних рекурентних вузлів – GRU (Gated Recurrent Units). Цей вибір зроблено в силу того, що саме ці мережі здатні враховувати історичні данні, та вже встигли позитивно себе проявити у вирішеннях задач де важлива послідовність інформації та історичні данні. На рисунку 2.1 зображена різниця між звичайними мережами прямого розповсюдження та рекурентними мережами.

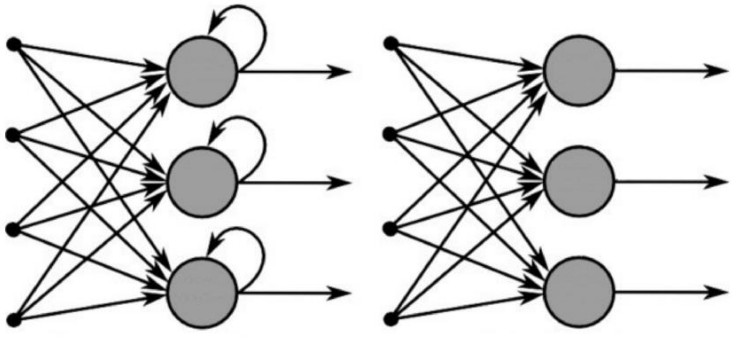


Рисунок 2.1 – Рекурентні (зліва) та звичайні FFNN мережі (справа)

Рекурентні нейронні мережі (RNN) з’явилися ще в 80-х роках, але змогли проявити свій дійсний потенціал лише за останні кілька років. Сама ідея

рекурентних нейронних мереж є дуже багатообіцяючою, адже це єдині мережі, що можуть запам’ятовувати минулі події зберігаючи свої вихідні значення. RNN завдяки внутрішній структурі пам’яті, можуть відзначати важливі властивості минулих подій та зберігати їх для подальшого використання. Ця властивість робить їх найкращим вибором для задач в яких використовується послідовна інформація. Послідовну інформацію можна визначити, як інформацію у якій зв’язки між зрізами інформації у різні моменти часу важливіші за інформацію, що несе одиничний зріз інформації в конкретний момент часу.

Рекурентні нейронні мережі можуть використовувати історичну інформацію завдяки своїм петлям. Вузол рекурентної мережі представлений на рисунку 2.2.

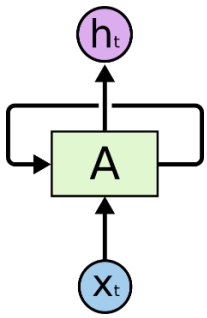


Рисунок 2.2 – Вузол рекурентної мережі [5]

З рисунку 2.2 видно, що вузол *A* нейронної мережі отримує на вхід значення x_t та на вихід віддає значення h_t, а петля дозволяє передавати інформацію від одного кроку до іншого.

Якщо розгорнути петлю, то можна представити вузол рекурентної мережі, як послідовність вузлів звичайної мережі прямого проходження. Це зображено на рисунку 2.3.

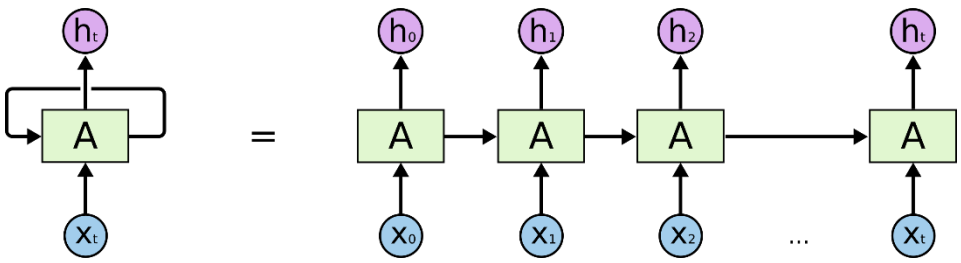


Рисунок 2.3 – Розгорнутий вузол рекурентної мережі [5]

Ланцюгова природа рекурентних мереж робить їх ідеальними для обробки послідовностей даних.

Незважаючи на всі свої чесноти та переваги рекурентні мережі стикаються з декількома проблемами. До проблем рекурентних мереж відносяться: проблема довготривалих зв'язків [6], а також проблема затухаючого та вибухаючого градієнту.

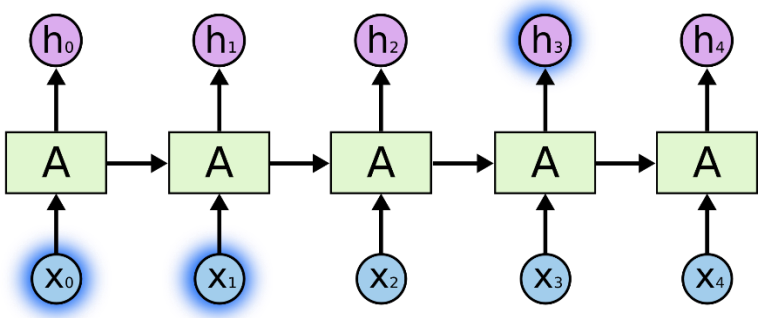


Рисунок 2.4 – Залежність між мало-віддаленими у часі подіями у рекурентних мережах [5]

Стосовно проблеми довготривалих зв'язків, то, хоча в теорії рекурентні мережі мають справлятися з цим, практика показала, що рекурентні мережі здатні впоратися та показати гарні результати, лише при інформації в якій наявні залежності між подіями, що відбулися в короткі проміжки часу (рисунок 2.4), а не коли діло заходить до виявлення залежностей між віддаленими у часі подіями (рисунок 2.5).

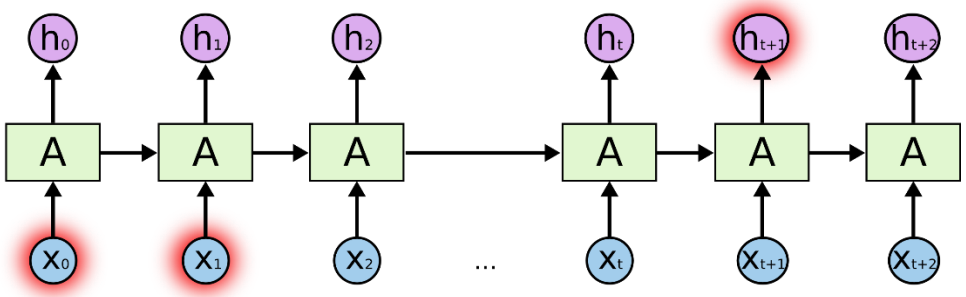


Рисунок 2.5 – Залежність між віддаленими у часі подіями у рекурентних мережах [5]

Проблема затухаючого градієнту виникає тоді коли функції активації насичуються та їх похідна наближається до нуля. Це призводить до того, шо навчання мережі майже не відбувається. Проблема вибухаючого градієнту – це протилежна проблема, з занадто великими значеннями градієнту та неконтрольованим процесом навчання. Тому потрібно ретельно обирати функції активації та обережно ставитися до ініціалізації параметрів мережі.

Проблему довготривалих залежностей та частково проблеми затухаючого та вибухаючого градієнту вирішують модифікації рекурентних мереж LSTM та GRU.

Мережі довготривалої короткочасної пам’яті – це підвид рекурентних мереж, що здатні завдяки внутрішній будові своїх вузлів запам’ятовувати довготривалі залежності [7]. Таку здатність вони мають завдяки своєму внутрішньому стану та механізмам запам'ятовування та забування інформації.

Внутрішня будова вузла звичайної рекурентної мережі та LSTM зображена на рисунку 2.5.

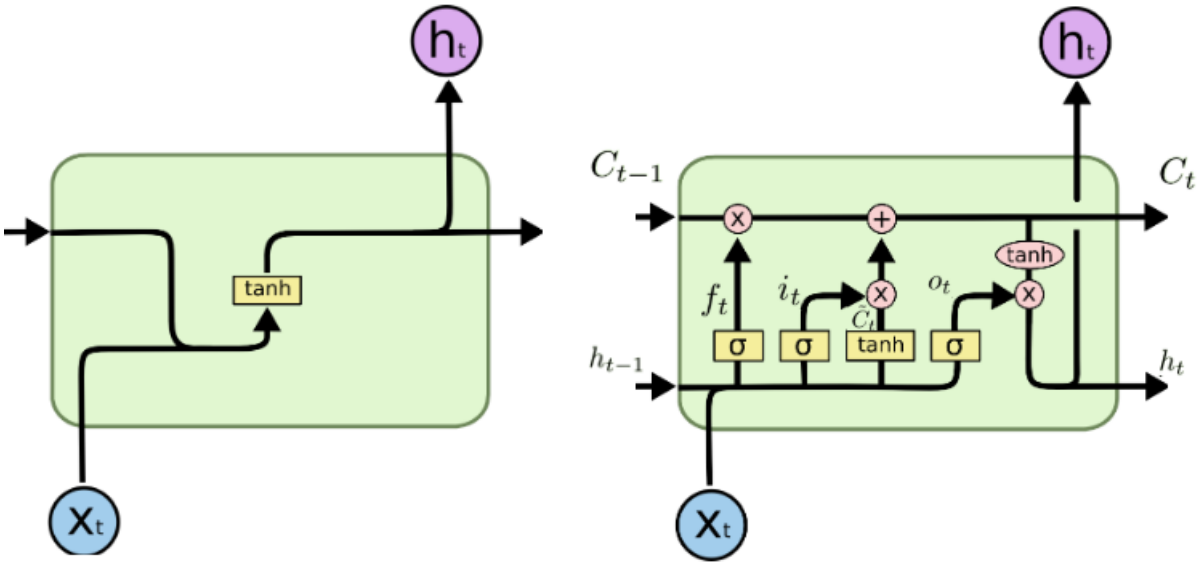


Рисунок 2.5 – Внутрішня будова вузла RNN (зліва) та LSTM (справа) [5]

Верхня горизонтальна лінія у вузлі LSTM C_t це стан вузла. Саме цей стан переносить довготривалу історичну інформацію від кроку до кроку. Окрім стану до наступного кроку також передається вихідне значення вузла. Також з рисунку

видно, що наявні два механізми: забування частини інформації зі стану та додавання нової інформації.

За перший механізм забування відповідає по-елементний добуток C_{t-1} та f_t, що є виходом шару, що отримує на вхід вихідні значення з минулого кроку, нові вхідні значення та навчається, яку краще інформацію забувати.

Механізм додавання нової інформації складається з двох частин: у першій сигмоїдний шар визначає, які значення та в якому обсязі ми будемо додавати, а в другій частині інший тангенціальний шар створює вектор нових потенційних значень. Потім результати об'єднуються у по-елементному добутку та додаються до стану вузла.

На вихід вузла подаються вхідні значення, що проходять через сигмоїдальний шар та по-елементно домножуються на стан вузла приведений до значень від -1 до 1 за допомогою *tanh*.

Також це можна описати наступними формулами:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (2.3)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (2.4)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \quad (2.5)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad (2.6)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (2.7)$$

$$h_t = o_t \odot \tanh(C_t), \quad (2.8)$$

Мережі вентильних рекурентних вузлів – це модифікація LSTM. Внутрішня будова вузла GRU зображена на рисунку 2.6.

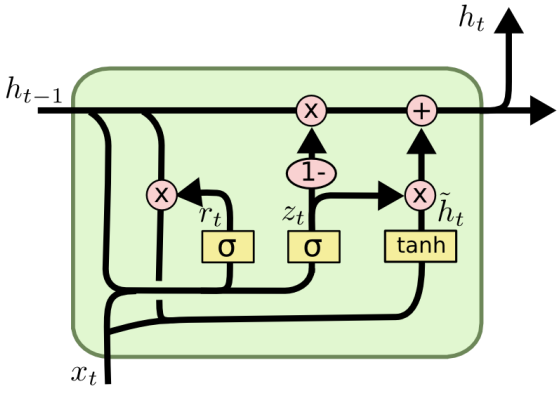


Рисунок 2.6 – Внутрішня будова вузла GRU [5]

В вузлах цієї мережі поєднані механізми забування та запам’ятовування нової інформації і єдиний механізм оновлення, також вихідним значенням вузла GRU є стан вузла, але він формується іншим способом [8]. Формули, що описують вузол мережі вентильних вузлів приведені нижче:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]), \quad (2.9)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \quad (2.10)$$

$$\tilde{h}_t = tanh(W \cdot [r_t \odot h_{t-1}, x_t]), \quad (2.11)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, \quad (2.12)$$

Оскільки саме ці підвиди архітектур рекурентних мереж мають властивості, що дозволяють їм краще поратися із проблемами рекурентних мереж та виявляти довготривалі залежності, та ідеально підходять для роботи з послідовностями інформації, було вирішено спробувати їх для побудови математичної моделі прогнозування наступних замовлень користувачів (мережі GRU показали трохи гірші результати, тому були використані саме LSTM).

Структурна схема моделі нейронної мережі представлена у додатку А. Модель має розгалуженні входи: ознаки підкатегорії, реальні ознаки користувача, категоріальні ознаки користувача, масиви реальних ознак елементів та масиви категоріальних ознак елементів, що входили до кожного з останніх п’яти замовлень, а також реальні та категоріальні ознаки кожного з п’яти замовлень. Кожна категоріальна ознака проходить через вбудований (embedded) шар та перетворюється у масив латентних ознак. Далі латентні ознаки об’єднуються з реальними ознаками. Плоскі вектори латентних та реальних ознак елементів, що входили до кожного з п’яти замовлень далі складаються у послідовність із п’яти векторів. Одночасно з цим плоскі вектори латентних та реальних ознак самих замовлень також складаються у послідовність із п’яти векторів. Вектор латентних та реальних ознак користувача розмножується до п’яти штук для того, щоб об’єднати вектор користувача з вектором кожного замовлення. Далі всі три матриці, кожна з п’яти векторів конкатенуються та отримується єдина послідовність з 5 векторів, кожен з яких представляє собою користувача, а також

реальні та категоріальні ознаки одного з п’яти замовлень та кожного з елементів цього замовлення. Ця послідовність відправляється на вхід шару LSTM з активаційною функцією ReLU. Потім йде об’єднання з плоским вектором латентних ознак підкатегорії, а далі йдуть чотири Dense шари (звичайні шари нейронів в нейронній мережі), з активаційною функцією ReLU, розміри яких зменшуються вдвічі з кожним наступним шаром, що виявляти абстрактні залежності. Останній вихідний Dense шар використовує сигмоїдальну активаційну функцію. У додатку К представлена таблиця розшифрування входів моделей, зображених у додатках А, Б та В. Варто також зауважити в даній реалізованій системі не використані дані реальних ознак, проте це не впливає на загальний підхід до побудови моделі, просто у моделі немає цих входів. Наявність реальних ознак також впливає на реалізацію та потребує підлаштовувати розмірності латентних просторів вбудованих рівнів. Варто також зауважити, що схема наведені у додатках А, Б, В мають трохи скорочений вигляд через брак місця, тобто для кожного об'єкту даних зображено узагальнені входи, як, наприклад – категоріальні ознаки елементу, яких насправді може бути декілька, але для спрощення усі вони не були зображені на схемі.

2.3 Модель матричного розкладу

Підхід матричного розкладу набув найширшого використання в системах рекомендацій. Як вже зазначалося в попередньому розділі, на даний момент він де-факто є основним підходом до вирішення задачі спільної фільтрації в рекомендаційних системах. Тому було вирішено не ігнорувати цей підхід, а використати його та посилити. Цей підрозділ розкриває як можна відтворити підхід матричного розкладу за допомогою нейронної мережі.

Оскільки засобами нейронних мереж можна запрограмувати будь-який алгоритм, то й підхід матричного розкладу для спільної фільтрації не є виключенням.

Нехай на вхід ми отримуємо вектори p_u та q_i, що відображає реальні та латентні ознаки користувача та підкатегорії відповідно. Повертаючись до

загального фреймворку, припустимо що ми маємо справу із взаємодією лише з одним елементом а не групою, і не маємо ознак замовлення, тоді формулу 2.1 можна переписати наступним чином:

$$\hat{y}_{ui} = f(P^T v_u^U, Q^T v_i^I | P, Q, \theta_f), \tag{2.13}$$

де $P \in R^{M \times K}$ та $Q \in R^{N \times K}$ – це матриця латентних ознак користувача та підкатегорії елементу відповідно. Тоді припустимо, що p_u – це $P^T v_u^U$, а q_i – це $Q^T v_i^I$. Визначимо функцію відображення для першого рівня нейронної моделі спільної фільтрації:

$$\varphi_1(p_u, q_i) = p_u \odot q_i, \tag{2.14}$$

де $\odot$ – це знак по-елементного добутку двох векторів p_u та q_i. Одразу після першого рівня ми потрапляємо до вихідного рівня:

$$\hat{y}_{ui} = a_{\text{вихід}}(h^T(p_u \odot q_i)), \tag{2.15}$$

де $a_{\text{вихід}}$ та h – це активаційна функція та крайові ваги вихідного рівня відповідно. Якщо як активаційну функцію $a_{\text{вихід}}$ використати тотожне відображення (identity function), а h зробити вектором із одиниць, то це повністю відтворить звичайну модель матричного розкладу. Проте якщо дозволити нейронній мережі підібрати значення h під час навчання, це дозволить вирахувати та використати значення справжньої значущості латентних вимірів. Вибір нелінійної активаційної функції $a_{\text{вихід}}$ узагальнить модель та зробить її більш виразною, ніж звичайна лінійна модель підходу з матричним розкладом.

В нашому випадку на вхід ми отримуємо не вектор q_i, що відображає реальні та латентні ознаки підкатегорії елементу. Дана модель при самостійному використанні не може врахувати історичні дані замовлень користувача, тому далі у наступному підрозділі буде описано використаний спосіб об’єднання її з моделлю на базі нейронної мережі задля використання історичних даних, та покращення результатів прогнозувань.

Структурна схема моделі підходу матричного розкладу приведена у додатку Б. Ця модель також має розгалужені входи: ідентифікатор підкатегорії, категоріальні ознаки користувача та реальні ознаки користувача. Категоріальні ознаки користувача за допомогою вбудованих шарів перетворюються у вектори латентних ознак та об'єднуються з реальними ознаками користувача. Категоріальні ознаки підкатегорії так само за допомогою вбудованих шарів перетворюються у вектори латентних ознак. Потім два плоских вектори йдуть на шар MF де відбувається їх по-елементне множення. Далі йдуть два Dense шари з активаційною функцією гіперболічного тангенсу та вихідний шар з сигмоїдальною функцією. У додатку К представлена таблиця розшифрування входів моделей, зображених у додатках А, Б та В.

2.4 Поєднання нейронного та матричного підходів

Для отримання кращих результатів у прогнозуванні складної функції взаємодії користувача та підкатегорій елементів та врахування історичних даних, щодо взаємодій користувача з групами елементів пропонується об'єднати дві моделі, одну з лінійним ядром на базі матричного розкладу та іншу з нелінійним ядром на базі багаторівневої нейронної мережі, в одну, щоб вони могли підсилити одна одну.

Щодо використання вбудованого (embedded) рівня буде використано окремі вбудовані рівні для кожної моделі, щоб можна було налаштувати розмірність латентного простору для кожної моделі окремо, бо оптимальні значення розмірності можуть відрізнятися для кожної.

Варто також зазначити, що на вхід моделі матричного розкладу буде подаватися лише вектори, що описують користувача і вектор латентних ознак підкатегорії, без значень, що стосуються реальних та категоріальних ознак замовлень, та історичної послідовності, бо цей метод не потребує цих даних, і вони можуть виявитися збитковими та деструктивними.

Далі описується задіяний спосіб об'єднання моделей.

Для об’єднання двох моделей буде виконана конкатенація їх останніх прихованих рівнів, а також вектору латентних ознак підкатегорії та передання на FFNN. Зображення об’єднаної моделі представлено у додатку В.

Цю модель можна умовно розділити три частини: на дві частини до конкатенація та частину після. Перша з частин до конкатенації це модель, що відтворює підхід матричного розкладу. Перша частина моделі відрізняється від моделі описаної в попередньому підрозділі, тим що результат по-елементного добутку векторів користувача та підкатегорії не йде на вхід шарів FFNN до об'єднання з результатами інших частин. В іншому все теж саме модель має розгалужені входи: ідентифікатор підкатегорії, категоріальні ознаки користувача та реальні ознаки користувача. Категоріальні ознаки так само за допомогою вбудованих шарів перетворюються у вектори латентних ознак. Єдине, що присутні додаткові вбудовані шари для категоріальних ознак користувача та підкатегорії для можливості окремого формування латентних векторів ознак для моделі матричного розкладу та передачі на вхід шарів нейронних мереж.

Друга частина моделі являє собою модель схожу на описану в підрозділі 2.2, за виключенням того, що після шару LSTM результат, не йдуть шари FFNN. Друга частина моделі має розгалужені входи: реальні ознаки користувача, категоріальні ознаки користувача, масиви реальних ознак елементів та масиви категоріальних ознак елементів, що входили до кожного з останніх п’яти замовлень, а також реальні та категоріальні ознаки кожного з п’яти замовлень. Категоріальні ознаки так само перетворюються у вектори латентних ознак та об’єднуються з реальними. На вхід шару LSTM так само подається послідовність з п’яти векторів, кожен з яких описує даного користувача, одне з п’яти його останніх замовлень та елементи цього замовлення.

Результат з другої частини (вихід шару LSTM) об’єднується з результатом першої частини, що представляє модель матричного розкладу, а також з латентним вектором ознак підкатегорії. Об’єднаний вектор потрапляє на чотиришарову FFNN мережу, кожен шар якої використовує активаційну функцію ReLU, та розмір скорочується вдвічі з кожним наступним шаром. Останній

вихідний Dense шар має за активаційну функцію сигмоїдальну логістичну функцію. У додатку К представлена таблиця розшифрування входів моделей, зображених у додатках А, Б та В.

2.5 Вибір активаційних функцій

Одним з важливих етапів розробки нейронної мережі є вибір функції активації нейронів. Вид функції активації багато в чому визначає функціональні можливості нейронної мережі і метод навчання цієї мережі. Класичний алгоритм зворотного поширення помилки добре працює на двошарових і тришарових нейронних мережах, але при нарощуванні більшої кількості шарів мережі, моужть виникати проблеми з навчанням. Це проблеми загасання чи вибуху градієнтів. У міру поширення помилки від вихідного шару до вхідного на кожному шарі відбувається множення поточного результату на похідну функції активації. Похідна у традиційній сігмоїдальній функції активації менше одиниці на всій області визначення, тому після декількох шарів помилка стане близькою до нуля. Якщо ж, навпаки, функція активації має необмежену похідну (як, наприклад, гіперболічний тангенс), то може статися вибухове збільшення помилки по мірі поширення, що призведе до нестійкого процесу навчання.

Поганою властивістю сигмоїдальної функції при насичені функції та приближенні до 0 або 1, градієнт стає близьким до 0. Це призводить до дуже повільного процесу навчання, адже при зворотньому розповсюджені помилки локальний градієнт множиться на загальний, і близький до нуля локальний градієнт фактично прирівнює до нуля загальний.

В цій роботі для побудови моделі використовуються активаційні функції сигмоїди здебільшого для вихідних нейронів, через діапазон значень від 0 до 1, гіперболічний тангенс для деяких внутрішніх шарів та вхідних нейронів, та ReLU для глибоких прихованих шарів, графіки функцій зображено на рисунках 2.7, 2.8 та 2.9 відповідно.

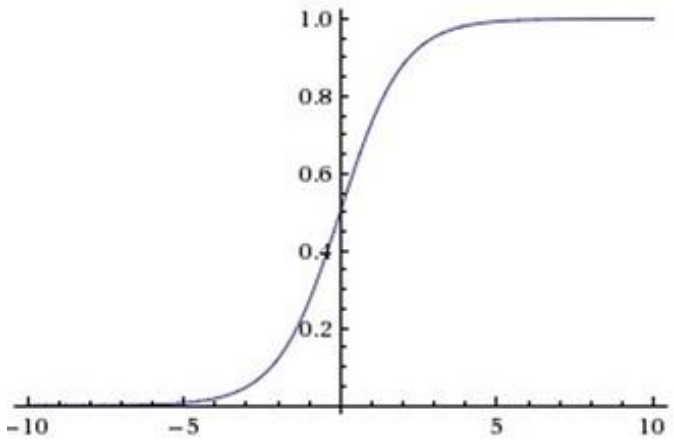


Рисунок 2.7 – Графік сигмоїдальної логістичної функції

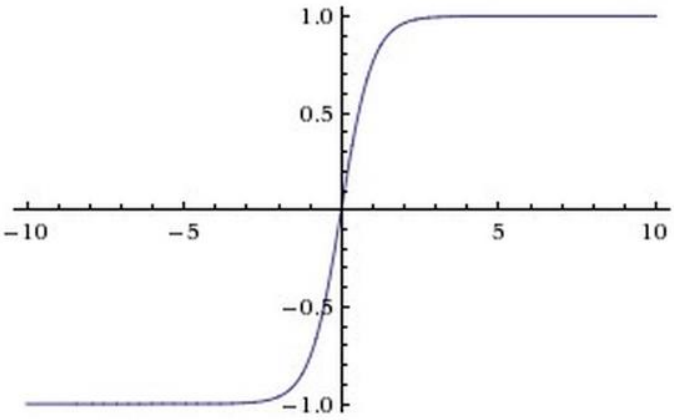


Рисунок 2.8 – Графік функції гіперболічний тангенс

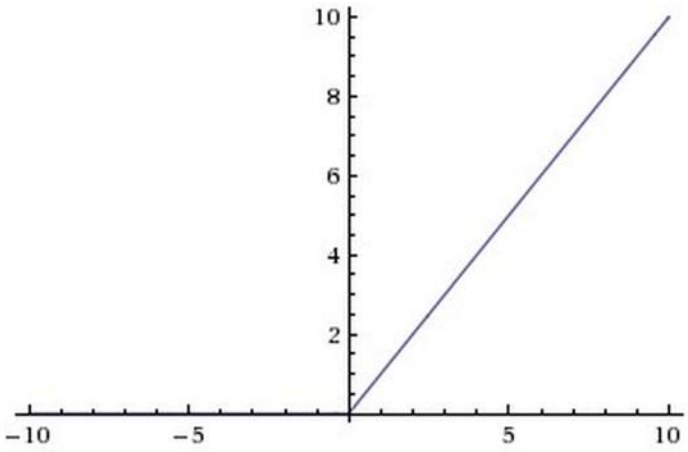


Рисунок 2.9 – Графік функції ReLU

Такий вибір зумовлений властивостями цих функцій. ReLU позбавлена тяжких в обчисленні операцій, призводить до швидкого навчання, а головне позбавлена проблеми затухання та вибуху градієнта, хоча і не завжди надійна. Симетричні

активаційні функції, типу гіперболічного тангенса забезпечують більш швидку збіжність, ніж стандартна логістична функція.

2.6 Функція втрати

В даній поставленій задачі прогнозовані значення мають двійкову природу, тобто ми маємо вгадати, відбудеться взаємодія користувача з елементом чи ні, а це означає що можна це розглядати, як задачу класифікації, де є лише два класи: взаємодія відбудеться – значення 1, взаємодія не відбудеться – значення 0.

Враховуючи це, як функцію втрати обґрунтовано буде обрати бінарну перехресну ентропію. Втрата крос-ентропії вимірює точність класифікаційної моделі, вихід якої є значенням вірогідності від 0 до 1. Втрата крос-ентропії збільшується, коли прогнозована ймовірність віддаляється від фактичного значення. Наприклад, прогнозування ймовірності 0,3, коли фактичне спостереження вказує на 1 – погане, і призводить до великого значення втрати. Ідеальна модель буде мати значення втрати рівне нулю.

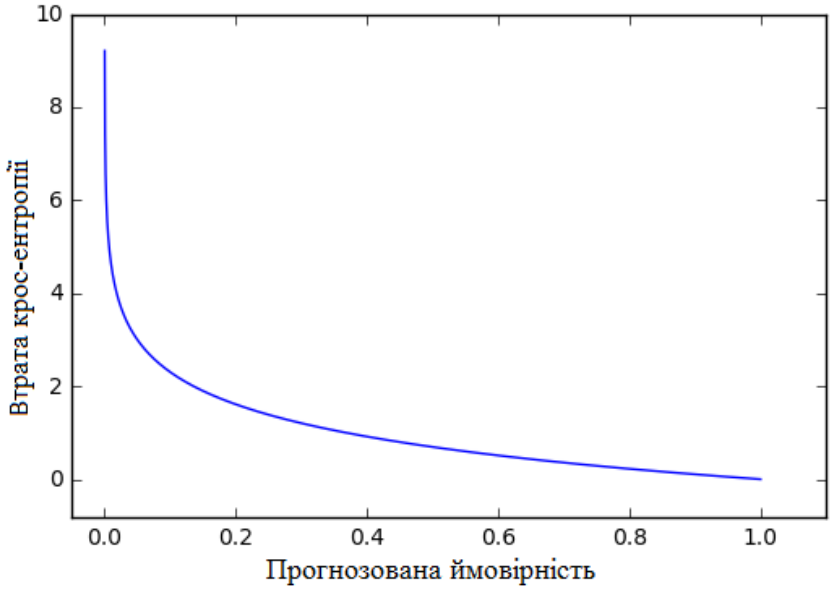


Рисунок 2.10 – Втрата крос-ентропії, при бажаному значенні = 1

Рисунок 2.10 зображує, як буде змінюватися значення втрати, при зміні прогнозованої ймовірності в ситуації, коли взаємодія між користувачем та елементом відбудеться, тобто бажане значення = 1.

Програмно це можна виразити, як представлено на рисунку 2.12:

```
def CrossEntropy(predicted, target):
    if target == 1:
        return -log(predicted)
    else:
        return -log(1 - predicted)
```

Рисунок 2.12 – Програмне представлення крос-ентропії

А також математично, як у формулі 2.16:

$$\sum_{(u,i)\in y\cup y^{-}} y_{ui}\log(\hat{y}_{ui}) + (1-y_{ui})\log(1-\hat{y}_{ui}) \quad , \qquad (2.16)$$

де y та y^{-} – це набір відомих позитивних взаємодій та негативних відповідно.

2.7 Висновки

У другому розділі були розглянуті модель матричного розкладу та модель на базі нейронної мережі для прогнозування взаємодії користувача. Було враховано вимогу, щодо врахування історичних даних взаємодій користувача з групами елементів. Для цього було запропоновано використати підвид рекурентних нейронних мереж довготривалою короткочасною пам’яттю ,що дозволяє виявляти залежності між подіями віддаленими у часі. Також було розглянуто нейронну модель матричного розкладу, та запропоновано її адаптацію для вирішення даної задачі.

У розділі було також синтезовано об’єднану модель на базі модулі матричного розкладу та моделі нейронної мережі задля отримання оптимальних результатів та врахування історичних даних.

Також було обрано функцію втрат для тренування моделей нейронних мереж. А також були розглянуті та обрані активаційні функції для вузлів нейронних мереж.

Запропоновані та описані моделі для побудови інтелектуальної системи підбору клієнтського контенту представлені також у додатках А, Б та В, а у додатку К представлена таблиця розшифрування входів моделей, зображених у додатках А, Б та В.

3 РОЗРОБКА ТЕХНІЧНОГО РІШЕННЯ, ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ

3.1 Опис роботи системи

Інтелектуальна система підбору клієнтського контенту являє собою комплексну систему, якою будуть користуватися представники клієнтських компаній та безпосередні користувачі, для яких буде підбиратися контент.

Передбачається, що система інтегрується з сторонніми клієнтськими системами. Інтеграція необхідна для отримання та попередньої обробки необхідних даних, що потім слугуватимуть вхідними для моделей прогнозування. Тому в системі присутній механізм отримування даних від клієнтських сервісів та баз даних. Також через програмний інтерфейс системи сторонні клієнтські системи можуть отримувати перелік підібраного контенту для користувачів.

Базуючись на даних про користувачів та елементи, отриманих від сторонніх систем, тренується модель прогнозування. Її параметри зберігаються в базі. Потім при ініціалізації сервісу генерації рекомендацій, тобто наборів рекомендованого підібраного контенту, цими параметрами буде проініціалізована модель цього сервісу, і саме вона буде робити прогнози. Оскільки через можливе горизонтальне масштабування, може бути запущено декілька одиниць цього сервісу, а трафік запитів рівномірно розподілений між ними, кожна модель буде мати різний стан, через проходження через неї різних даних. Моделі будуть щодня в моменти найменшої активності актуалізуватися новими, підготовленими в оффлайн режимі параметрами моделі, що отримала на вхід повний обсяг всіх взаємодій користувачів у правильній послідовності. Такий підхід використаний задля того, щоб була змога масштабувати компонент системи, що відповідає за прогнозування.

Підбір контенту користувачам відбувається за рахунок отриманих прогнозів від математичної моделі прогнозування, що намагається вгадати вірогідність успішної взаємодії користувача з певними елементами або групами

елементів, маючи доступ до інформації щодо минулої взаємодії користувача з елементами.

До списку додається декілька елементів зі списку збереженого контенту, якщо такий є у користувача. Також задля отримання кращого показника покриття до списку підібраного контенту додаються декілька елементів, що зазвичай найрідше рекомендуються користувачам.

Варто також зазначити, що система має механізм отримання та зберігання даних ідентифікації користувачів з клієнтських систем, для їх авторизації у системі без додаткової реєстрації.

Система матиме обробляти запити, що надходитимуть з веб-інтерфейсу браузера та мобільних додатків. Ці запити можуть стосуватися авторизації чи реєстрації, тоді система має відповідно обробити вхідні данні перевірити пароль, електронну адресу та авторизувати чи зареєструвати користувача. Після успішної реєстрації користувач автоматично авторизується у системі.

Далі авторизований користувач отримує доступ до інтерфейсу, будь-то веб-інтерфейс чи мобільний додаток, що дозволяє переглядати персонально підібраний контент та відповідну інформацію про нього. Якщо це окремий додаток чи інтерфейс інтелектуальної системи підбору контенту, то у користувача також є можливість додати певний елемент у список збереженого, для швидкого доступу у майбутньому. Після додавання у список збереженого користувач може відфільтрувати його по категоріям або видалити з нього елементи, виконавши певні маніпуляції з інтерфейсом. Якщо клієнт компанія бажає використовувати підібраний контент для відображення на інтерфейсах своїх систем, тоді сторонні клієнтські системи чи інтерфейси можуть звертатися до програмного інтерфейсу системи для отримання переліку підібраного контенту, для подальшої обробки та відображення. Система отримавши запит від авторизованого користувача має надіслати йому персонально підібраний контент. Якщо користувач використовує інтерфейс інтелектуальної системи підбору клієнтського контенту то він може відфільтрувати контент по категоріях. Авторизований представник компанії може переглянути використовуючи інтерфейс системи елементи, що найчастіше

рекомендуються користувачам та фільтрувати цей список по категоріях. Також він здатен змінювати кількість рекомендованих елементів, що відображаються користувачу. Авторизовані користувачі та представники клієнтів можуть здійснити вихід із системи. Після чого потрібно буде знов здійснити авторизацію для користування системою. Також у додатках Г та Д наведені діаграми послідовностей, що описують процес отримання рекомендації та оновлення моделі у сервісі генерації рекомендацій. Детально цей та інші структурні компоненти системи описані у підрозділі 3.4. Детальний опис кожного сценарію використання наведений у підрозділі 3.2.

3.2 Сценарії використання системи

При описі сценаріїв використання системи визначаються три актори: неавторизованій користувач, авторизований користувач та представник клієнта. Після проходження авторизації неавторизований користувач стає одним з двох останніх. Діаграма сценаріїв використання наведена у додатку Е.

Таблиця 3.1 Сценарій використання – «Авторизація»

Сценарій використання	Авторизація
Опис	Для отримання доступу до функціоналу, необхідно пройти процес авторизації у системі, завдяки якому система може віднести користувача до певної групи, що має певні права та можливості.
Актори	Неавторизований користувач
Припущення	Отримання доступу до інтерфейсу системи
Кроки	Натиснути на посилання, що веде до форми авторизації, проходження сценаріїв з заповненням форми авторизації
Нефункціональні вимоги	немає

Таблиця 3.2 Сценарій використання – «Реєстрація»

Сценарій використання	Реєстрація
Опис	Щоб система могла ідентифікувати користувача та в майбутньому авторизувати необхідно надати свої ідентифікаційні дані.
Актори	Неавторизований користувач
Припущення	Отримання доступу до інтерфейсу системи
Кроки	Натиснути на посилання, що веде до форми реєстрації, виконати дії з заповнення форми
Нефункціональні вимоги	немає

Таблиця 3.3 Сценарій використання – «Введення паролю»

Сценарій використання	Введення паролю
Опис	Введення ідентифікаційних даних.
Актори	Неавторизований користувач
Припущення	Отримання доступу до інтерфейсу системи
Кроки	Натиснути на посилання, що веде до форми реєстрації або авторизації
Нефункціональні вимоги	немає

Таблиця 3.4 Сценарій використання – «Введення підтвердження для паролю»

Сценарій використання	Введення підтвердження для паролю
Опис	Введення ідентифікаційних даних.
Актори	Неавторизований користувач
Припущення	Отримання доступу до інтерфейсу системи

Продовження таблиці 3.4

Кроки	Натиснути на посилання, що веде до форми реєстрації
Нефункціональні вимоги	немає

Таблиця 3.5 Сценарій використання – «Введення електронної пошти»

Сценарій використання	Введення електронної пошти
Опис	Введення ідентифікаційних даних
Актори	Неавторизований користувач
Припущення	Отримання доступу до інтерфейсу системи
Кроки	Натиснути на посилання, що веде до форми реєстрації або авторизації
Нефункціональні вимоги	немає

Таблиця 3.6 Сценарій використання – «Перегляд персонально підібраного контенту»

Сценарій використання	Перегляд персонально підібраного контенту
Опис	Користувач може споглядати відображений на інтерфейсі персонально підібраний контент та відповідну інформацію про нього
Актори	Авторизований користувач
Припущення	Попередня авторизація у ролі користувача
Кроки	Отримати доступ до інтерфейсу
Нефункціональні вимоги	Релевантний контент

Таблиця 3.7 Сценарій використання – «Перегляд персонально підібраного контенту»

Сценарій використання	Перегляд персонально підібраного контенту
Опис	Користувач може споглядати відображений на інтерфейсі персонально підібраний контент та відповідну інформацію про нього
Актори	Авторизований користувач
Припущення	Попередня авторизація у ролі користувача
Кроки	Отримати доступ до інтерфейсу
Нефункціональні вимоги	Релевантний контент

Таблиця 3.8 Сценарій використання – «Вибір фільтру по категорії»

Сценарій використання	Вибір фільтру по категорії
Опис	Маючи список контенту, можна його відфільтрувати вказавши певну категорію
Актори	Авторизований користувач, Представник клієнта
Припущення	Попередня авторизація у ролі користувача або представника клієнта
Кроки	Отримати доступ до інтерфейсу та переліку елементів контенту
Нефункціональні вимоги	немає

Таблиця 3.9 Сценарій використання – «Додати в збережені»

Сценарій використання	Додати в збережені
Опис	Маючи список контенту, можна додати певні елементи до списку збережених, де згодом можна буде швидко знайти ці елементи

Продовження таблиці 3.9

Актори	Авторизований користувач
Припущення	Попередня авторизація у ролі користувача
Кроки	Отримати доступ до інтерфейсу та переліку елементів контенту, натиснути на елемент інтерфейсу, що відповідає за збереження
Нефункціональні вимоги	немає

Таблиця 3.10 Сценарій використання – «Видалити зі збережених»

Сценарій використання	Видалити зі збережених
Опис	Маючи список збереженого контенту, можна видалити з нього певні елементи, якщо вони вже не потрібні
Актори	Авторизований користувач
Припущення	Попередня авторизація у ролі користувача, наявність в списку збережених хоча б одного елементу
Кроки	Отримати доступ до інтерфейсу та переліку збережених елементів контенту, натиснути на елемент інтерфейсу, що відповідає за видалення
Нефункціональні вимоги	немає

Таблиця 3.11 Сценарій використання – «Перегляд елементів, що найчастіше рекомендуються»

Сценарій використання	Перегляд елементів, що найчастіше рекомендуються

Продовження таблиці 3.11

Опис	Відображення списку елементів контенту відсортованих в залежності від частоти потрапляння до підібраних для користувачів списків контенту, тобто найпопулярніші елементи
Актори	Представник клієнта
Припущення	Попередня авторизація у ролі представника клієнта
Кроки	Отримати доступ до інтерфейсу
Нефункціональні вимоги	немає

Таблиця 3.12 Сценарій використання – «Перегляд елементів, що найчастіше додаються в збережені»

Сценарій використання	Перегляд елементів, що найчастіше додаються в збережені
Опис	Відображення списку елементів контенту відсортованих в залежності від частоти потрапляння користувацьких списків збережених елементів
Актори	Представник клієнта
Припущення	Попередня авторизація у ролі представника клієнта
Кроки	Отримати доступ до інтерфейсу
Нефункціональні вимоги	немає

Таблиця 3.13 Сценарій використання – «Перегляд інформації про елемент»

Сценарій використання	Перегляд інформації про елемент
Опис	Відображення детальної інформації та описання елементу контенту
Актори	Авторизований користувач
Припущення	Попередня авторизація у ролі користувача
Кроки	Отримати доступ до інтерфейсу, вибір конкретного елементу
Нефункціональні вимоги	немає

Таблиця 3.14 Сценарій використання – «Зміна кількості рекомендованих елементів»

Сценарій використання	Зміна кількості рекомендованих елементів
Опис	Зміна кількості набору елементів контенту, що підбираються для кожного користувача
Актори	Представник клієнта
Припущення	Попередня авторизація у ролі представника клієнта
Кроки	Отримати доступ до інтерфейсу, маніпуляції з елементами інтерфейсу, що відповідають за зміну кількості елементів
Нефункціональні вимоги	немає

Таблиця 3.15 Сценарій використання – «Вихід»

Сценарій використання	Вихід

Продовження таблиці 3.5

Опис	Вихід з системи; після виходу, щоб отримати доступ до функціоналу системи потрібно буде знову пройти авторизацію
Актори	Представник клієнта, Авторизований користувач
Припущення	Попередня авторизація у ролі представника клієнта або користувача
Кроки	Маніпуляції з елементами інтерфейсу, що відповідають за вихід з системи
Нефункціональні вимоги	немає

3.2 Вибір архітектури системи

Для якісної та вчасної реалізації системи завжди дуже важливо детально проробити архітектуру системи, що далі буде використовуватися спеціалістами з розробки, підтримки системи і також виступатиме важливим артефактом в процесі комунікації із клієнтами. Опис архітектури важливий для вдалого планування ресурсів, для планування процесу розробки, знадобиться під час експлуатації та процесу документування. Вкрай необхідно мати опис архітектури системи для подальшого процесу розробки, розширення функціоналу, модифікації чи модернізації системи.

На сьогоднішній день до програмного забезпечення та інформаційних систем висувається чимало вимог. Вимоги щодо швидкості розробки також вимагають дуже високого темпу, часто потрібно поставити новий функціонал за лічені тижні. В умовах жорсткої конкурентної боротьбі, швидкого технологічного розвитку та масової діджиталізації всіх сфер навантаження на системи зростає разом із вимогами до їх швидкодії. Застосування мають бути готові до швидкого масштабування при швидкому рості користувачів, що може статися в дуже короткі терміни. Усе це разом із високим рівнем складності сучасних виробничих систем вимагає корегувати процеси розробки та саму структуру та архітектуру

програмних систем та ретельно її продумувати, щоб встигати в строки та будувати конкурентне програмне забезпечення та комплексні інформаційні системи.

До недавнього часу в розробці програмного забезпечення панувала монолітна архітектура, проте останнім часом все більше набирає популярності мікро-сервісна архітектура та підхід до побудови програмних систем.

Розгляньмо монолітну архітектуру програмного забезпечення. При монолітній архітектурі програмне забезпечення постачається як єдиний розгортаємий програмний артефакт.

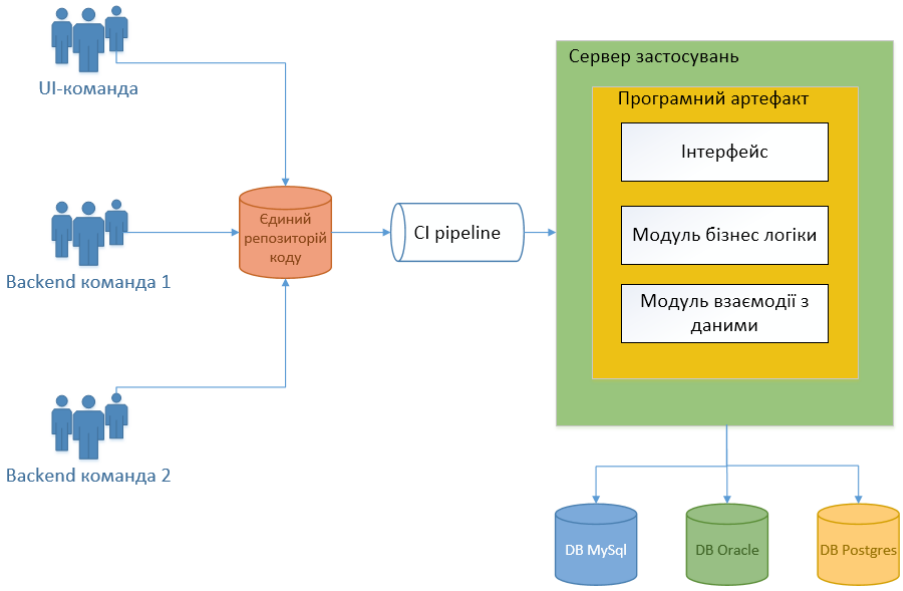


Рисунок 3.1 – Розробка монолітного програмного забезпечення

Вся логіка щодо обробки UI-запитів, бізнес логіка, та доступ до бази даних збирається в єдиний артефакт та розгортається на сервері застосувань. Проте в той час як розгортатися буде лише один артефакт, перед цим над його розробкою працюють здебільшого декілька команд і працюють над різними його частинами, що відповідають за окремі шматки функціональності та логіки. Єдина кодова база, сильно ускладнює та сповільнює процес розробки, а також управління. Також важливо зазначити, що цей єдиний артефакт програмного забезпечення матиме доступ до усіх задіяних джерел даних, що у свою чергу також призводить до складнощів адміністрування, підтримки, а також до можливого неправильного

використання програмних компонентів та джерел даних. Узагальнений процес розробки монолітного програмного забезпечення зображений на рисунку 3.1.

Проблема монолітного програмного забезпечення, що при нарощуванні нового функціоналу та загальної складності системи, витрати на організацію робочого процесу, комунікації між командами значно зростають.

Мікро-сервісна архітектура долає багато недоліків монолітного способу побудови програмного забезпечення. В основі концепції мікро-сервісної архітектури лежить розбиття застосування на невеликі слабо зв'язані частини, що виступатимуть у ролі розподілених сервісів. Кожен мікро-сервіс є компонентом системи, що відповідає за невелику вузько визначену частину функціональності. Такими компонентами легко управляти та розробляти, бо можна побудувати незалежні процеси розробки для кожної команди.

Узагальнений процес розробки мікро-сервісного програмного забезпечення зображений на рисунку 3.2.

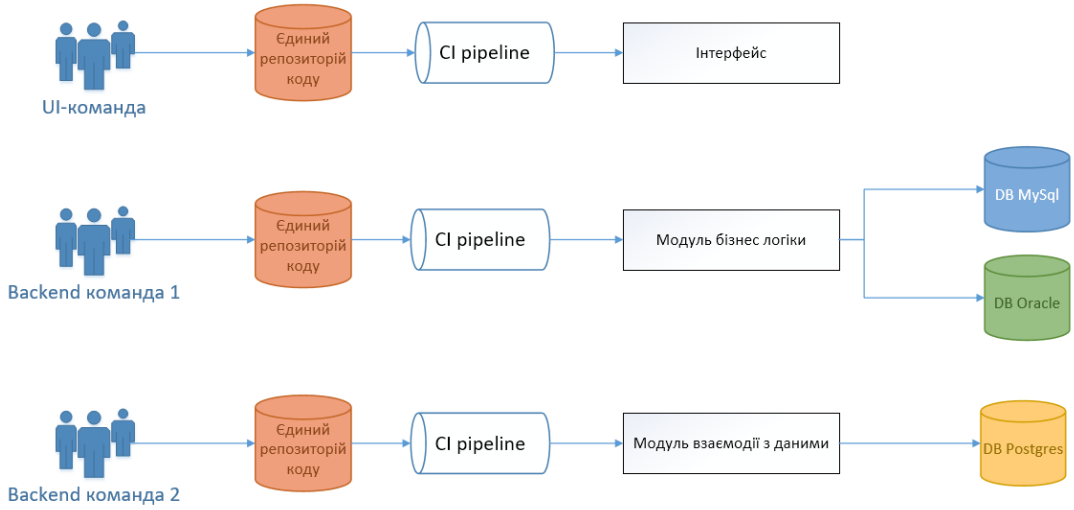


Рисунок 3.2 – Розробка мікро-сервісного програмного забезпечення

З рисунку 3.2 видно, що розробляючи мікро-сервіси кожна команда володіє та відповідає за окремий сервіс, може незалежно одна від одної збирати, розгортати та тестувати свою частину системи, бо має окремий незалежний репозиторій та інфраструктуру (сервер та базу даних).

Мікро-сервісна архітектура має такі характеристики:

— логіка застосування розбита на невеликі частини з чітко визначеними границями, що координуються разом для отримання цілісного рішення;
— кожен компонент розгортається повністю незалежно один від одного, окрім цього компонент може бути скопійований та використаний в іншому застосуванні;
— кожен компонент може бути реалізований за допомогою будь-яких технологій, адже мікро-сервіси спілкуються між собою завжди використовуючи юніфікований формат, як, наприклад, JSON;
— мікро-сервіси завдяки своїй природі маленьких та незалежних розподілених застосувань дозволяють компаніям мати декілька невеликих команд, що сфокусовані над розробкою певного функціоналу і їх знання можуть обмежуватися лише їх доменом або частиною.

Порівняльний аналіз. Порівнюючи монолітну та мікро-сервісну архітектуру можна зробити висновок, що монолітний підхід краще підходить для побудови невеликих застосувань, щодо яких відомо, що вони в майбутньому не будуть ускладнюватися та потребуватимуть лише незначних змін. Також така архітектура влучна на перших стадіях розробки проекту, в умовах, коли ще не відома кінцева форма системи, а лише невелика частина функціоналу, що може бути реалізована в одному застосуванні. Якщо грамотно реалізувати застосування, застосовуючи модульний підхід, то з нарощення функціоналу, його буде легко рознести на декілька незалежних сервісів.

Враховуючи поставлені до розроблюваної інтелектуальної системи підбору клієнтського контенту вимоги щодо можливості швидкого нарощення нового функціоналу та можливості швидкого масштабування, краще буде одразу обрати мікро-сервісну архітектуру. Хоча зазвичай системи прийнято розробляти невеликі продукти починаючи з монолітного підходу, а потім переходити по мірі ускладнення системи до мікро-сервісного підходу, знаючи те, що потрібно одразу побудувати комплексну систему та бути готовими до швидкого масштабування функціоналу та системи в цілому, вибір падає на мікро-сервісну архітектуру.

3.4 Розробка структури системи

Визначившись у попередньому підрозділі з підходом до побудови архітектури системи, тепер детальніше розберемо компоненти системи та опишемо кожен з них. Також у додатку Ж приведена структурна схема системи.

Щоб підібрати контент для користувачів потрібно спочатку натренувати математичну модель, а потім нею користуватися. Також для надання персоналізованих послуг з підбору контенту, кожен користувач повинен пройти авторизацію, для чого його дані мають зберігатися в базі. Дані про користувачів, та їх поведінку з точки зору взаємодії з продуктами/елементами компанії-клієнта надходитимуть зі сторонніх клієнтських систем, тому повинен бути механізм інтеграції.

Окрім UI веб-застосування та мобільного додатку система включає декілька окремих відносно незалежних слабо зв’язаних компонентів – сервісів. Кожен сервіс являє собою самостійну частину програмного забезпечення, що поставляється та розгортається окремо.

Запити від користувачів системи потрапляють до сервісу, обробки запитів, де відповідно о обробляються. Щоб обробити та відповісти на запити цей сервіс спілкується із базою веб-сервісів для виконання таких дія як, наприклад, авторизація, реєстрація, додавання елемента в збережене чи видалення. Окрім цього відбувається комунікація із сервісом генерації рекомендацій для отримання переліку елементів з найкращими прогнозами. До цього сервісу можуть також звертатися сторонні системи для отримання переліку підібраного контенту.

Сервіс генерації рекомендацій отримує запити від сервісу обробки запитів користувачів та сторонніх систем, що потребують прогнозів. Цей сервіс з допомогою математичної моделі, генерую перелік елементів з найкращими показниками вірогідності взаємодії для користувачів. Цей перелік відрізняється від переліку сервісу обробки запитів тим, що він ще не збагачений елементами зі збереженого та найменш популярними елементами. Сервіс генерації рекомендацій спілкується з базою даних аналітики та прогнозувань, звідки

витягує параметри для ініціалізації своєї моделі. Також сервіс підготовки та тренування моделей сповіщає цей сервіс, коли потрібно забирати нові параметри з бази для актуалізації моделі.

Сервіс підготовки та тренування моделей прогнозування відповідає за оффлайн процес тренування математичної моделі прогнозування використовуючи дані про попередні взаємодії користувача та елементи, що він отримує з бази даних аналітики та прогнозувань. Цей сервіс отримує з бази усі дані про взаємодії користувачів та пропускає крізь модель зберігаючи історичну послідовність. Сервіс з певною періодичністю зберігає стан моделі в базу та сповіщує сервіс генерації рекомендацій, що потрібно оновити модель. Також з час від часу із більшою періодичністю, щоб охватити більший обсяг даних цей сервіс донавчує модель новими даними та також сповіщає про необхідність актуалізувати модель. Сповіщення відбувається з допомогою брокера повідомлень, шляхом відправлення повідомлення до теми (topic), на яку підписані всі підняті одиниці сервісу генерації рекомендацій.

Сервіс постачання даних від клієнта відповідає за витяг з систем та баз даних клієнта необхідних даних, що стосуються взаємодії користувачів з елементами, а також даних ідентифікації користувачів, щоб вони могли авторизуватися у системі. Отримавши дані, сервіс перевіряє їх на валідність, підчищає за необхідності, видаляє збиткові, обертає в потрібний формат та передає до сервісу обробки та підготування даних, з допомогою брокера повідомлень.

Сервіс обробки та підготування даних отримавши дані від сервісу постачання даних вже робить маніпуляції з даними, щоб привести їх до повної готовності для використання сервісом підготовки та тренування моделей прогнозування. Він обраховує додаткові властивості об’єктів даних, зводить все в єдине ціле та зберігає в базі даних для аналітики та прогнозувань. Це, що стосується даних для прогнозування. Також цей сервіс обробляє дані необхідні для веб-сервісів та зберігає їх в базі веб-сервісів.

Для якісної побудови системи на базі мікро-сервісів застосовано такі шаблони як сервіс-виявлення (service-discovery) та API-шлюз (API-Gateway). Клієнтське застосування могло б робити запити до кожного із сервісів. Але такий підхід зразу натикається на масу обмежень – необхідність знати адресу кожної кінцевої точки, робити запит за кожною частиною інформації окремо і самостійно збирати результат. Для рішення такого роду проблем застосовуємо – єдину точку входу. Їх використовують для прийому зовнішніх запитів і маршрутизації в необхідні сервіси внутрішньої інфраструктури, віддачі статичного контенту, аутентифікації, стрес-тестування, міграції сервісів, динамічного управління трафіком[9]. Сервіс-виявлення (service-discovery) дозволяє автоматично визначати мережеві адреси для доступних розгорнутих одиниць сервісів, які можуть динамічно змінюватися з причин масштабування, падінь і оновлень. Ключовою ланкою тут є реєстр сервісів.

3.5 Вибір та обґрунтування інтсрументів та технологій

Для розробки рішення задачі прогнозування для інтелектуальної системи підбору клієнтського контенту було обрано мову програмування Python. Це рішення ґрунтується на тому, що Python – це найпопулярніша мова для розробки ML/AI рішень, тому має дуже широку спільноту, та багато інформації в мережі про вирішення тих чи інших проблем, значну кількість відшліфованих бібліотек, а також проста у використані та має гарний функціонал для програмування математичних розрахунків та алгоритмів.

Під час розробки ML/AI рішення дуже важливо обрати правильний фреймворк глибинного навчання для роботи. Обраний фреймворк має задовільнити всім вимогам, таким як: сумісність з уже існуючими системами, використання знайомих команді мов програмування, цінова доступність розробки рішення, широка спільнота, рівень готовності фреймворку до впровадження в промисловість, продуктивність, швидкість розробки рішення тощо. Тому перед вибором фреймворку було проаналізовано декілька основних та найпопулярніших.

TensorFlow – фреймворк розробленій компанією Google на мовах програмування C++ та Python. TensorFlow вважається однією з найкращих бібліотек з відкритим кодом для чисельного обчислення. TensorFlow гарний для дуже складних та незвичайних проектів. Він використовується для побудови систем розпізнавання голосу/зображень та текстових програм (наприклад, Google Translate).

До переваг можна віднести:

— багато документації та інструкцій;

— пропонуються засоби для моніторингу навчальних процесів моделей та візуалізації (Tensorboard);

— підтримується великою спільнотою розробників та технічних компаній;

— він підтримує розподілене навчання;

— Tensorflow Lite дозволяє працювати з низькою затримкою на мобільних пристроях.

Проте є і недоліки:

— Низька продуктивність у порівнянні з CNTK чи MXNet.

— Tensorflow є достатньо низькорівневим та вимагає багато шаблонного коду.

— Складний процес відладки.

Keras – це мінімалістична бібліотека на базі Python, що запускається поверху TensorFlow, Theano або CNTK. Вона була розроблений інженером Google для полегшення експериментів. Підтримує широкий спектр шарів нейронної мережі, таких як згорткові шари, рекуррентні шари або щільні шари. Гарний у використанні в областях перекладу, розпізнавання образів, розпізнавання мовлення, прогнозування тощо.

Переваги Keras:

— Прототипування виконується дійсно швидко і легко;

— Легкий з точки зору побудови моделей DL з великою кількістю шарів;

— Модулі повністю налаштовуються;

— Спрощений та інтуїтивно зрозумілий інтерфейс;

— Вбудована підтримка для навчання на кількох GPU;

— Може працювати на Spark;

— Підтримує NVIDIA GPU, Google TPU і GPU з підтримкою Open-CL, такими як AMD.

Недоліки:

— Може виявитися занадто високорівневим;

— Обмежується Tensorflow, CNTK та Theano backends.

PyTorch – це Python-спадкоємець бібліотеки Torch, написаний в Lua і є великим конкурентом для TensorFlow. Він був розроблений Facebook і використовується Twitter, Salesforce та ще багато ким. PyTorch в основному використовується для швидкої та ефективної підготовки глибоких моделей навчання, тому є вибором великої кількості дослідників.

PyTorch має ряд істотних переваг:

— Процес моделювання простий та прозорий завдяки архітектурному стилю;

— Типовий режим за замовчуванням більше схожий на традиційне програмування, і можна використовувати загальні інструменти налагодження, такі як відладчик pdb, ipdb або PyCharm;

— Оснащений безліччю попередньо вбудованих моделей та модульних частин, що є готовими та зручними для комбінування;

— Розподілене навчання підтримується з версії 0.4.

Недоліки PyTorch:

— Не вистачає можливості model serving;

— Рішення ще не готове до виробництва;

— Не має інтерфейсів для моніторингу та візуалізації, як наприклад, Tensorboard.

У підсумку, було обрано бібліотеку Keras, бо вона здатна забезпечити швидкий процес розробки та проведення експериментів, найпростіша у використанні, має можливість підключення до засобів моніторингу та візуалізації, має виробничий рівень готовності та широку спільноту, а також дозволяє реалізувати розроблені математичні моделі та в повному обсязі провести заплановані експерименти.

Для розробки інших частин системи, таких як сервіс отримання та підготовки даних, сервіс взаємодії з клієнтом, сервіс обробки запитів та інших було обрано мову програмування Java версії 8. Java характеризується досить високою швидкодією, однією з найкращих серед високорівневих мов програмування; має дуже велику спільноту та бібліотеки для вирішення майже будь-яких завдань. Мова програмування Java з самого початку розроблялася як легка у застосування та для побудови програмного коду, який буде легко підтримувати та легко розібрати стороннім командам. Java добре підходить для розробки великих комерційних проектів, є стабільною та перевіреною часом, постійно дороблюється та підтримується багатьма компаніями та спільнотами по всьому світі. Мова має багато платформ для полегшення та прискорення розробки часто зустрічних задач. Все це робить Java гарним вибором для розробки.

Для швидкої розробки мікро-сервісів було обрано комбінацію фреймворків та інструментів: Spring Framework, Spring Boot, Spring JDBC Template, Spring MVC. Spring Framework серед іншого забезпечує гарний механізм інверсії контролю та ін’єкції залежностей, Spring Boot прибирає необхідність у більшості конфігурацій застосування, Spring MVC робить комунікацію через HTTP протокол простою та легкою в імплементації, Spring JDBC Template позбавляє великої кількості шаблонного коду, виграє по швидкодії у порівнянні із Spring Data та ORM та дозволяє писати нестандартні запити. Ці інструменти у поєднанні дозволяють досить швидко розробляти невеликі

помірної складності програмні артефакти та ідеально підходять для мікро-сервісів.

Взаємодія між сервісами всередині системи відбуватиметься на базі принципів REST, здебільшого з використання формату JSON.

Для розгортання мікро-сервісів було обрано рішення на базі Docker, Openshift, Kubernetes. З допомогою Docker можна створити образи віртуальних машин, описавши та зібравши все необхідне для запуску на машині застосування. Маючи образ такого контейнеру можна легко робити горизонтальне масштабування завдяки платформам Openshift та Kubernetes. При необхідності на базі такого образу будуть підійматися нові одиниці цього сервісу та автоматично запускатися. Можна налаштувати автоматичне масштабування при зростанні навантаження. В якості сервіс-виявлення (service-discovery) гарним вибором буде Eureka, а в якості API-шлюзу (api-gateway) – Zuul. Ribbon – це інструмент балансування на стороні клієнта. У порівнянні з традиційним, тут запити проходять безпосередньо за потрібною адресою, що виключає зайвий вузол при виклику. З коробки він інтегрований з механізмом сервіс-виявлення, який надає динамічний список доступних одиниць сервісів для балансування між ними.

У додатку И наведена діаграма розгортки системи. З діаграми видно, що сервіси розгортаються за допомогою віртуальних контейнерів Docker. Дані зберігаються в базі даних MySql. Для забезпечення асинхроної комунікації між сервісами по специфікації JMS використаний брокер повідомлень Apache MQ (AMQ). Система розгортається в хмарі з використанням сторонніх IaaS рішень. Взаємодія між структурними компонентами ведеться на базі протоколу TCP/IP та більш високого рівня HTTP.

3.6 Висновки

В розділі було визначений та обґрунтований підхід до побудови архітектури системи, а потім спроектовано та розроблено архітектуру інтелектуальної системи підбору клієнтського контенту.

Було розібрано та детально описано роботу системи. Розібрані та опрацьовані сценарії використання системи. Також в рамках розділу було розроблено структуру системи, з детальним описом та призначення компонентів системи. Описаний механізм інтеграції системи зі сторонніми системами та внутрішня взаємодія компонентів системи.

Для реалізації технічного рішення був проведений аналіз та зроблений обґрунтований вибір інструментів та технологій.

Інтелектуальна система підбору клієнтського контенту є комплексною системою, що складається з декількох окремих незалежний застосувань, що у поєднанні являють собою цілісну структуру. Кожен компонент – мікро-сервіс забезпечує та відповідає за чітко визначену частину логіки і функціоналу. Система спроектована за принципами мікро-сервісної архітектури та слідує усім кращим практикам. Впроваджені такі шаблони, як сервіс-виявлення та API-шлюз. Взаємодія між сервісами відбувається за принципами REST через веб-запити, а також з використанням брокерів повідомлень.

4 ДОСЛІДЖЕННЯ ТА ТЕСТУВАННЯ

4.1 Дослідження ефективності моделей

Для аналізу ефективності моделей та порівняння було використано такі метрики, як коефіцієнт влучень або влучність (Hit Ratio, далі HR), а також трохи видозмінений нормалізований дисконтований кумулятивний приріст – NDCG (Normalized Discounted Cumulative Gain) по K елементах. За значення K було прийнято 15, бо це значення за замовченням кількості елементів підібраного контенту, а також трохи більше середнього значення кількості підкатегорій у замовленні.

HR метрику можна розглядати як просту та базову метрику для оцінки якості ранжування. Припустимо у наступному замовленні користувача будуть присутні товари чотирьох підкатегорій, тоді метрика подивиться чи є кожна з цих категорій у списку з K елементів, та підрахує середнє значення. Тобто якщо у списку будуть лише 3 підкатегорії з чотирьох, то оцінка буде $\frac{3}{4} = 0.75$.

HR метрика проста у розумінні та реалізації, проте має істотний недолік – вона не враховує положення елементів у списку, тому для того зоб врахувати також положення у списку використано також метрику NDCG. NDCG – одна з найчастіше уживаних метрик якості ранжування. Так само як і DCG вона враховує порядок розташування елементів у списку, за рахунок ділення на логарифм номеру позиції у списку $k+1$ (тобто вага кожної позиції рівна зворотньому логарифму). Проте окрім цього, це є нормалізована версія DCG метрики, завдяки діленню на максимальне можливе (ідеальне) значення DCG – IDCG. NDCG можна виразити формулами 4.1 та 4.2:

$$\text{DCG@}K = \sum_{k=1}^{K} \frac{r^{true}{}_{(p(k))}}{\log_2(k+1)}, \tag{4.1}$$

$$\text{NDCG@}K = \frac{DCG@K}{IDCG@N}, \tag{4.2}$$

де r^{true} – дійсна релевантність елементу, тобто чи буде присутня підкатегорія у наступному замовлені, що приймає значення 0 або 1, $p(k)$ – прогнозована ймовірність появи категорії у замовленні, під номером *k* у списку із *K* елементів з найвищою ймовірністю, а $IDCG@N = \sum_{n=1}^{N} \frac{1}{\log_2(n+1)}$, бо r^{true} може приймати значення 0 або 1, а *N<=K* – це кількість підкатегорій у замовленні (якщо підкатегорій більше 15, то *N=15*). Модифікація метрики NDCG, полягає саме у тому, що IDCG береться по кількості підкатегорій у замовленні, що в даному випадку і буде ідеальним значенням. Таким чином метрика NDCG наслідує від DCG можливість врахування позиції в списку, при цьому приймаючи значення в діапазоні від 0 до 1 включно.

Дослідження проводилися на реальних даних торгової мережі. Щоб оцінити якість прогнозувань було адаптовано, так звану, «leave-one-out» оцінку, що широко зустрічається у літературі [10, 11, 12], коли для кожного користувача для тестування залишається його останнє замовлення, а решта використовується для тренувань. Проте замість одного замовлення для кожного користувача було залишено 20% від його останніх замовлень, при чому якщо, 20% замовлень становить не ціле значення, то відбувається округлення завжди у більшу сторону. Тестова частина становить ≈22% всього набору даних. Варто зауважити, що якщо кількість попередніх замовлень менша п’яти, то дані доповнюються нулями. Прогнозування відбувалося для кожної з категорій (близько 200).

Далі наведені результати досліджень по кожній з трьох моделей описаних у другому розділі. По осі x відкладена розмірність латентного простору, у який відображається користувач та підкатегорія. Варто зазначити, що досліджувана модель використовувала дані, в яких категоріальними ознаками користувача, та елемента були лише їх ідентифікатори, а реальних ознак не було, інакше потрібно було б підбирати кількість латентних ознак, для користувача та підкатегорії таку, щоб при об’єднанні з реальними ознаками, розмірності векторів співпадали та можна було б виконати по-елементне множення.

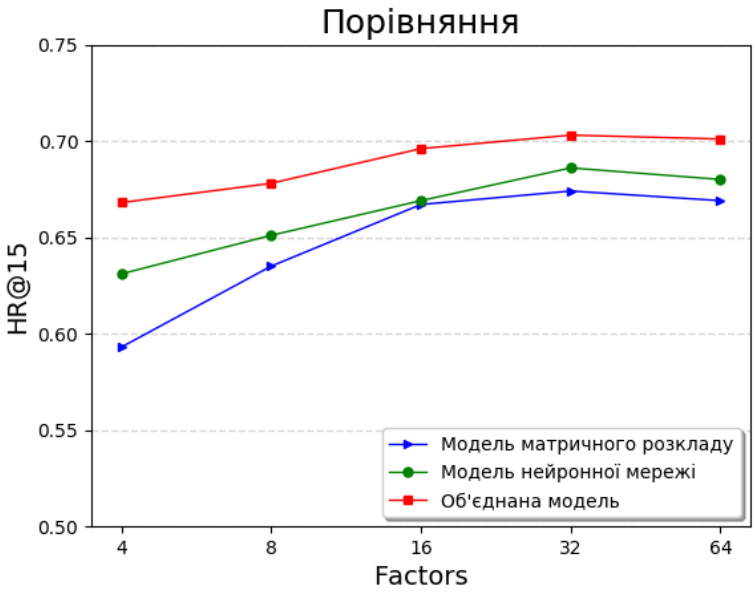


Рисунок 4.1 – Значення метрики HR@15

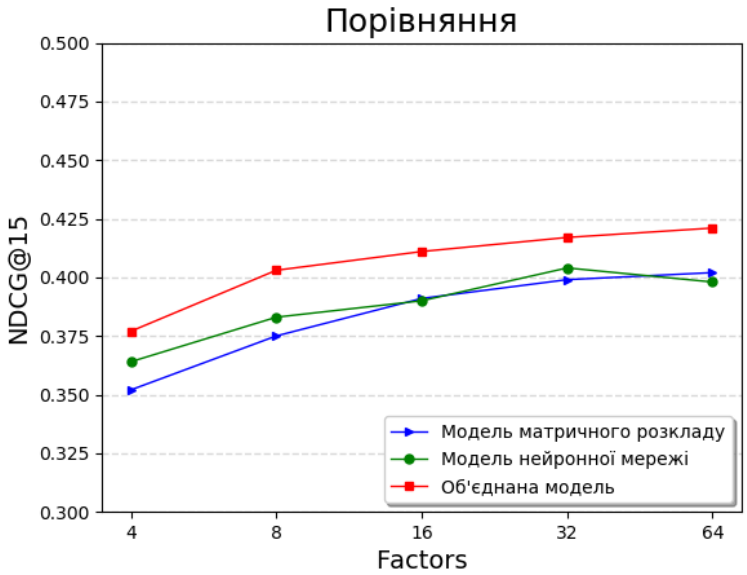


Рисунок 4.2 – Значення метрики NDCG@15

З рисунків 4.1 та 4.2 видно, що всі три моделі показали задовільні результати. Проте при порівнянні видно, що модель на базі нейронної мережі відпрацьовує трохи краще за модель матричного розкладу, а об’єднана модель показує найкращі результати.

За результатами досліджень можна зробити висновок, що об’єднання моделей на базі нейронної мережі та матричного розкладу виправдало себе, а кожна з моделей показала прийнятні результати. Показники метрики HR свідчать про те, що у списку підібраного контенту буде більше половини актуального для

нього. А показники метрики NDCG вказують на те, що приблизно половина актуальних для користувача підкатегорій будуть у списку підібраного контенту та в середньому будуть знаходитись в межах середини, або рівномірно розподілені по всьому списку.

Також при проведені досліджень було експериментально досліджено вплив окремого завчасного тренування моделей перед об'єднанням. Тобто значення ваг з співпадаючих рівнів окремо натренованих моделей нейронних мереж були використані для початкової ініціалізації параметрів об'єднаної мережі. У таблиці 4.1 наведено результати експерименту для об'єднаної моделі:

Таблиця 4.1 – Вплив завчасного тренування моделей

	З завчасним тренуванням		Без завчасного тренування	
Розмірність латентного простору	HR@15	NDCG@15	HR@15	NDCG@15
4	0.668	0.377	**0.675**	**0.387**
8	0.678	0.403	**0.682**	**0.406**
16	**0.693**	**0.411**	0.687	0.408
32	**0.703**	**0.417**	0.693	0.410
64	**0.701**	**0.421**	0.689	0.407

Спираючись на результати приведені у таблиці 4.1 можна зробити висновок, що завчасне тренування моделі позитивно впливає на результати, при збільшенні розмірності латентного простору, ця закономірність може бути пояснена тим, що при збільшеній розмірності латентного простору було збільшено кількість параметрів, що були ініціалізацовані за рахунок завчасно натренованих моделей.

4.2 Тестування системи

Для забезпечення гарного та надійного покриття тестами системи, використано 3 рівні тестів: юніт-тести, компонентні тесті та інтеграційні тести.

Юніт-тести – це тести, що пишуться в межах застосування та націлені на точкове покриття методів та функцій.

Компонентні тесті – тести більш високого рівня, що покривають функціонал цілого компонента, сервісу, з використання заглушок та моків.

Інтеграційні тести – тести найвищого рівня, що пишуться в окремому застосуванні для покриття функціоналу, що забезпечуються декількома компонентами системи, які мають при цьому інтегруватися та взаємодіяти.

Для написання юніт-тестів та компонент тестів використовуються бібліотеки Junit, , Spring Test , Spring MVC Test. Для реалізації інтеграційних тестів задіяно фреймворк Cucumber та мову специфікацій Gherkin. Цей фреймворк та мова дозволяють у сприятливому для читання людиною форматі, звичайною англійською мовою описувати потрібні сценарії та підв’язувати до них реалізацію на мові Java.

Нижче у таблиці 4.1 приведено матрицю відповідності вимог та інтеграційних тестів. За допомогою анотацій, що підтримуються фреймворком Cucumber, кожному тестовому сценарію присвоєно ідентифікаційний номер, який також зазначений у таблиці. Опис вимог наведений нижче у таблиці 4.3.

Таблиця 4.2 – Матриця відповідності вимог

№ тесту / № вимоги	1	2	3	4	5	6	7	8	9
1	✓		✓	✓	✓				✓
2		✓				✓	✓	✓	
3			✓	✓					✓
4				✓	✓				
5					✓				
6						✓		✓	
7							✓		

Продовження таблиці 4.2

№ тесту / № вимоги	1	2	3	4	5	6	7	8	9
8								✓	
9									✓

Таблиця 4.3 – Опис вимог до сценаріїв тестування

№ вимоги	Опис
1.	Успішна авторизація користувача
2.	Успішна авторизація представника клієнта
3.	Авторизований користувач отримує підібраний контент
4.	Авторизований користувач може додати елемент контенту у збережене
5.	Авторизований користувач може видалити елемент контенту із списку збереженого
6.	Представник клієнта може переглянути контент, що найчастіше потрапляє до рекомендованого
7.	Представник клієнта може змінити кількість рекомендованого контенту
8.	Представник клієнта може фільтрувати контент по категорії
9.	Користувач може фільтрувати контент по категорії

4.3 Висновки

В цьому розділі були приведені результати досліджень проведених із кожною з моделей описаних у розділі 2 та приведених у додатках А, Б та В. Були також описані метрики, за допомогою яких була проведена оцінка моделей, та пояснено, чому саме ці метрики були обрані.

Кожна із моделей показала прийнятні результати. Найкращі результати показала об’єднана модель матричного розкладу та нейронної мережі, тому саме вона була обрана для реалізації у системі інтелектуального підбору клієнтського контенту. Побудована модель враховує історичні дані послідовності минулих взаємодій користувача, що було одним із завдань при побудові системи. Дослідження також показали, що об’єднану модель краще ініціалізувати з використанням параметрів завчасно натренованих моделей.

Також у розділі було описано тестування системи. Кожен сценарій використання було покрито інтеграційним тестом. Все тести проходять успішно. Були також описані технології та інструменти, що були використані для тестування трьох рівнів: юніт, компонент та інтеграційного.

5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

В даному розділі викладено маркетинговий аналіз перспектив реалізації інтелектуальної системи підбору клієнтського контенту, як окремого продукту, а також оцінено можливості її ринкового впровадження.

5.1 Опис ідеї проекту

Проект націлений на надання послуг з реалізації та впровадження системи підбору клієнтського контенту для офлайн та онлайн роздрібних торгівців та постачальників різного роду контенту. Така система зможе підвищити обсяги продажів та підвищити рівень лояльності користувачів клієнтських систем.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система впроваджується в бізнес, для аналізу поведінки користувачів та генерування рекомендацій покупок або дій із сервісом, що в свою чергу збільшить обсяг продажів, лояльність існуючих клієнтів та допоможе привернути нових, а також оптимізувати закупки.	Супермаркети/ гіпермаркети, магазини оффлайн роздрібної торгівлі	Оптимізація закупок, збільшення обсягу продажів.
	Онлайн роздрібна торгівля.	Покращення показників метрик «час на сайті» та «глибина перегляду», збільшення обсягів продажів, оптимізація закупок.

Продовження таблиці 5.1

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	Онлайн відео прокат.	Зростання лояльності користувачів до ресурсу; покращення показників метрик «час на сайті» та «глибина перегляду»; збільшення обсягів продажів.

На ринку немає можливих затверджених конкурентів, що надають послуги конкретно з розробки та впровадження систем підбору контенту, такі рішення на разі розробляються або в середині компаній або на індивідуальне замовлення IT-компаніями широкого профілю. Також можна додати, що існуючі системи будують свої рекомендації на аналізі взаємодії користувача з окремими елементами, а не групами, і прогнозують таку ж взаємодію.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка	N (нейтральна	S (сильна
		Мій проект	Luxoft	Epam			
1.	Вартість	Середня вартість	Висока вартість	Висока вартість			+

Продовження таблиці 5.2

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна)	S (сильна сторона)
		Мій проект	Luxoft	Epam			
2.	Вартість обслуговування	Середня вартість	Висока вартість	Висока вартість			+
3.	Прогноз наступної покупки	Так	Ні	Ні			+
4.	Зручність інтерфейсу	Зручний інтерфейс	Незручний інтерфейс	Зручний інтерфейс		+	
5.	Швидкість впровадження	Швидка	Тривала	Тривала			+

Знов ж таки закріплених прямих конкурентів немає на ринку немає. IT-компанії широкого профілю також здатні розробляти системи підбору контенту, проте будуть програвати вузькоспеціалізованій компанії в компетенції та досвіді; швидкість впровадження та вартість за рахунок цього буде більш конкурентною.

5.2 Технологічний аудит ідеї проекту

Щоб здійснити технологічний аудит ідеї проекту потрібно проаналізувати можливі технології для реалізації продукту.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Back-end частина інтеграції з існуючими системами бізнесу	Java 8, ActiveMQ, Spring Integration, JDBC template,	Наявні, дороблювати непотрібно.	Доступні, відкритий доступ.
		Java 8, Spring Integration, RabbitMQ	Наявні, дороблювати непотрібно.	Доступні, відкритий доступ.
2.	Аналітичний модуль рекомендацій	Python 3, Java 8, Keras	Наявні, можна доробити\модифікувати та покращити	Доступні, відкритий доступ.
3.	Back-end частина підтримки інтерфейсу та графічної аналітики	Java 8, Spring MVC, Spring Data, JDBC template.	Наявні, дороблювати непотрібно.	Доступні, відкритий доступ.
		Php	Наявні, можливо доведеться дороблювати для деяких задач інтеграції.	Доступні, відкритий доступ.
4.	База даних	Oracle	Наявні, дороблювати непотрібно.	Доступна, за оплату.

Продовження таблиці 5.3

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
4.	База даних	PostgreSQL	Наявні, дороблювати непотрібно.	Доступна безкоштовна та платна версії.
		MySql	Наявні, дороблювати непотрібно.	Доступна безкоштовна та платна версії.
5.	Клієнтський веб інтерфейс	Angular5	Наявні, дороблювати непотрібно.	Доступні, відкритий доступ.
		ReactJS	Наявні, дороблювати непотрібно.	Доступні, відкритий доступ.
6.	Клієнтський мобільний інтерфейс	Java/Kotlin, Swift	Наявні, дороблювати непотрібно.	Доступні, відкритий доступ, для розробки необхідні комп’ютери MAC.
		JavaScript	Наявні, дороблювати непотрібно.	Доступні, відкритий доступ.
Обрані технології реалізації ідеї проекту: Java 8, Spring Boot, String Data, Spring JDBC template, Java/Swift, Angular4+, MySql, Keras				

Для реалізації проекту наявні та доступні усі необхідні технології, що свідчить про технологічну здійсненність проекту.

5.3 Аналіз ринкових можливостей запуску

При проведені дослідження ринкових можливостей запуску був проведений аналіз основних показників ринку. Аналіз наведений у таблиці 5.4.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	Зараз на ринку України відсутні інтелектуальні системи підбору клієнтського контенту як окремий продукт, є лише послуги з розробки персональних рішень
2.	Загальний обсяг продаж	Український ринок CRM систем за 2017 рік оцінений у 30 млн доларів, що є лише 0.13 частиною світового обсягу [13]. Глобальний ринок CRM показав $36,8 млрд прибутків за 2017 рік [14]
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Економія на масштабах, доступ до ресурсів.

Продовження таблиці 5.4

№	Показники стану ринку (найменування)	Характеристика
5.	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	8-30

Компанії завжди зацікавленні у збільшенні обсягів продажів та притягнення до себе нових користувачів своїх сервісів та покупців. На сьогоднішній день бізнес зацікавлений у впровадженні ML/AI рішень для різної оптимізації та максимізації прибутку.

В сучасній Україні сформовані всі умови для входження на ринок, оскільки розвинута IT-індустрія та можна легко набрати фахівців до команди для швидкої реалізації та впровадження.

Має місце технологічне відставання більшості компаній та їх бажання надолужити та впроваждувати новітні технології.

Єдиним чинником проти може стати відсутність коштів у компаній клієнтів, проте на ринку можна знайти достатню кількість великих гравців з необхідними фінансовими ресурсами.

Враховуючи позитивну динаміку ринку він є привабливим для входження.

У таблиці 5.5 визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.Оптимізація закупок; 2. Збільшення продажу; 3. Утримання клієнта	1. Супермаркети/ гіпермаркети, оффлайн магазини роздрібної торгівлі. 2. Онлайн платформи роздрібної торгівлі.	1. Ціна 2. Досвід успішних впроваджень 3. Репутація постачальника 4. Функціональність 5. Відповідність стратегії замовника 6. Вартість впровадження 7. Вартість підтримки 8. Послуги консалтингу 9. Кількість партнерів	1. Якість 2. Захищеність 3. Зручність інтерфейсу 4. Простота та зручність 5. Масштабованість 6. Комплексність рішення 7. Наявність методик роботи з системою 8. Динаміка розвитку продукту 9. Надійність ПО 10. Можливість інтеграції 11. Налаштування своїми силами

Для проведення аналізу ринкового середовища проведений аналіз факторів, що сприяють ринковому впровадженню (таблиця 5.7) і факторів, що йому перешкоджають (таблиця 5.6).

Негативними факторами при виході на ринок стане несприятливе правове та економічне становище в державі, а також можливість появи нових конкурентів в умовах швидкого розвинення IT-індустрії.

Таблиця 5.6 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1.	Поява нових конкурентів	Можлива поява нових конкурентів через динамічний розвиток ринку CRM систем та подібних рішень.	Цінова та якісна конкуренція.
2.	Економічний	Економічна нестабільність середовища, загальна зниження платоспроможності	Оптимізація бізнес процесів, режим економії вихід на нові зарубіжні ринки.
3.	Правова безпека	Правова незахищеність бізнесу в Україні	Патентування, переведення бізнесу закордон.

До факторів, що позитивно впливатимуть на можливості виходу на ринок можна віднести новизну ринку, розвинуту IT-індустрію, що відіграватиме роль сприятливого середовища, а також популярність нових IT-рішень на базі ML/AI технологій.

Таблиця 5.7 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1.	Відносно новий ринок для Українського бізнесу	Масове впровадження CRM систем почалося в Україні лише в 2010.	Інтенсивна маркетингова компанія.
2.	Легка доступність людський ресурсів.	Багато IT фахівців на ринку	Економія витрат на пошук персоналу.

Для проведення аналізу пропозиції визначаються загальні риси конкуренції (таблиця 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Конкурентне середовище ще не до кінця сформоване	Відсутня чітка сегментація ринку CRM та подібних рішень	Розвивати нішовий сегмент, щоб захопити найбільшу частку ринку
Національний	Аналізується ринок України	Продаж по всіх регіонах країни
Внутрішньогалузева	Конкуренція лише межах систем CRM та рекомендаційних систем.	Глибокий аналіз ринку та постійний моніторинг конкурентів своєї галузі

Продовження таблиці 5.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Характер конкурентних переваг: цінова і нецінова	Цінова. Універсальний характер рішення.	Пошук шляхів зниження собівартості.

Більш детальний аналіз умов конкуренції в обраній галузі економіки проведений за моделлю 5 сил М. Портера. Дані наведені у таблиці 5.9.

Проаналізувавши таблицю 5.9, можна зробити висновок, що можливість виходу на ринок з огляду на конкурентну ситуацію є високою. Для виходу на ринок товар в першу чергу повинен пропонувати унікальні характеристики, які відсутні у продуктах конкурентів. Продукт буде конкурувати за рахунок новаторського рішення, а також своєю універсальністю, швидкістю впровадження та можливостями розширення та масштабування.

На основі аналізу конкуренції, проведеного в таблиці 5.9, а також із урахуванням характеристик ідеї проекту (таблиця 5.2), вимог споживачів до товару (таблиця 5.5) та факторів маркетингового середовища (таблиці 5.6 та 5.7), визначається та обґрунтовується перелік факторів конкурентоспроможності, що надається у таблиці 5.10.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Прямі конкуренти відсутні. Непрямі конкуренти: провайдери CRM систем, що мають модулі рекомендацій, персональні розробки від IT компаній.	Бар'єри для входження на ринок відсутні.	Ринкова сила поста-чальників дуже слабка	Обмеж-ення платоспро можністю	Товароз амінник и відсутні
Висновки	Конкурентне середовище ще не до кінця сформоване	Потенційни ми конкурента ми є провайлери CRM систем	Постачальники не диктують умови роботи	Покупця м легше відмовит ися від рішення, ніж платити великі гроші.	Обмеже ння для роботи на ринку через товари замінни ки відсутні.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Універсальний характер застосування	На ринку відсутній готовий продукт універсального характеру застосування.
2.	Ціна	Захоплення ринку за рахунок низької маржинальності продажів та пошуку шляхів зменшення собівартості.
3.	Зручність інтерфейсу	Один з ключових факторів ефективності користування системою.
4.	Масштабованість системи	Один з ключових факторів можливості адаптації системи під зростання бізнесу клієнта.
5.	Швидкість розробки та впровадження	Для бізнесу завжди важлива швидкість, щоб бути попереду конкурентів

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін «Інтелектуальна система підбору клієнтського контенту»

№	Фактор конкурентоспроможності	Рейтинг товарів-конкурентів у порівнянні						
		-3	*-2*	*-1*	*0*	*1*	*2*	*3*
1.	Універсальний характер застосування	✓						
2.	Ціна			✓				

Продовження таблиці 5.11

№	Фактор конкурентоспроможності	Рейтинг товарів-конкурентів у порівнянні						
		-3	*-2*	*-1*	*0*	*1*	*2*	*3*
3.	Зручність інтерфейсу				✓			
4.	Масштабованість системи				✓			
5.	Швидкість розробки та впровадження	✓						

Фінальним етапом ринкового аналізу можливостей впровадження проекту є проведення SWOT-аналізу (складання матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 5.12).

Ринкові загрози та ринкові можливості виступають як наслідки (прогнозовані результати) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (таблиця 5.8, 5.9 аналіз потенційних конкурентів).

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів у таблиці 5.13.

Таблиця 5.12 – SWOT- аналіз стартап-проекту

Сильні сторони: 1. Цінова перевага. 2. Передова технологія, новаторське рішення.	Слабкі сторони: 1. Висока вартість IT фахівців. 2. Відсутність досвіду ведення бізнесу.
Можливості: 1. Відносно новий ринок для Українського бізнесу – несформоване конкурентне середовище. 2. Багато IT фахівців на ринку – економія витрат на пошук персоналу. 3. Зростання попиту на рекомендаційні системи.	Загрози: 1. Можлива поява нових конкурентів через динамічний розвиток ринку CRM систем та подібних рішень. 2. Економічна нестабільність середовища, загальна зниження платоспроможності. 3. Правова незахищеність бізнесу в Україні

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	**Кооперація Максимізація спільного виграшу**	**Дуже висока (за рахунок кооперації з існуючими великими гравцями IT ринку)**	**Час реалізації прискорений (1 рік)**

Продовження таблиці 5.13

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
2.	Індивідуалізм Максимізація власного виграшу	Середня (адже немає досвіду та репутації для отримання фінансування)	Стандартний час реалізації (1,5-2 роки)
3.	Суперництво Максимізація відносного виграшу	Середня (адже немає досвіду та репутації для отримання фінансування)	Стандартний час реалізації (1,5-2 роки)

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 5.14).

Після визначення та аналізу цільових груп потенційних клієнтів робиться вибір на яких групах потрібно робити фокус, та можна буде будувати стратегію охоплення ринку.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Супермаркети/ гіпермаркети, магазини оффлайн роздрібної торгівлі	Готові, але впровадження трудоємке	Зростаючий попит (~3 млн долларів)	Помірна. Конкурентне середовище ще не до кінця сформоване, але потенційними конкурентами є провайлери CRM систем	Складно-щі незначні
2.	Онлайн роздрібна торгівля.	Готові	Зростаючий попит (~1 млн долларів)		
3.	Онлайн відео прокат.	Готові	Зростаючий попит (~1 млн долларів)		
Які цільові групи обрано: Усі, але з пріоритетом на першу групу.					

Базові стратегії в обраних сегментах ринку представлені у таблиці 5.15.

Таблиця 5.15 – Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1.	Кооперація	Стратегія концентрованого маркетингу	Цінова перевага. Передова технологія, новаторське рішення.	Стратегія спеціалізації

Залежно від міри сформованості галузевого ринку, характеру конкурентної боротьби, необхідно обрати одну з трьох стратегій конкурентної поведінки: розширення первинного попиту, оборонну або наступальну стратегію або ж застосувати демаркетинг або диверсифікацію (таблиця 4.16).

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№	Чи є проект «першо-прохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1.	Так	Шукати нових	Буде копіювати, але при цьому удосконалювати їх	Стратегія заняття конкурентної ніші

Таблиця 5.17 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1. Якість 2. Захищеність 3. Зручність інтерфейсу 4. Простота та зручність 5. Масштабованість 6. Комплексність рішення 7. Наявність методик роботи з системою 8. Динаміка розвитку продукту 9. Надійність ПО 10. Можливість інтеграції 11.Налаштування своїми силами 12. Ціна	Стратегія спеціалізації	1. Цінова перевага. 2. Передова технологія, новаторське рішення. 3. Зручність інтерфейсу. 4.Масштабованість системи. 5. Можливість інтеграції.	1. Підвищення обсягів продажів. 2. Оптимізація закупок. 3. Задоволений клієнт.

5.5 Розроблення маркетингової програми

Маркетингова програма – це намічений для планомірного здійснення, об'єднаний єдиною метою та залежний від певних строків комплекс взаємопов'язаних завдань і адресних заходів соціального, економічного, науково-технічного, виробничого, організаційного характеру з визначенням ресурсів, що використовуються, а також джерел одержання цих ресурсів [15].

Маркетингова програма, спрямована на вирішення окремих комплексних проблем, наприклад на організацію виробництва нового продукту, на завоювання нового сегмента або ринку в цілому [15].

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 5.18 підведені підсумки результатів попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1.	Збільшення обсягів продажів товарів	Аналіз споживчої поведінки для ефективного рекламування	Цінова перевага, зручний інтерфейс, вузька спеціалізація лише на рекомендаційних системах
2.	Оптимізація закупок	Прогнозування споживчого попиту на товари	

Продовження таблиці 5.18

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
3.	Покращення показників метрик «час на сайті» та «глибина перегляду»	Затримка та зацікавлення споживачів за рахунок аналізу споживчої поведінки	Цінова перевага, зручний інтерфейс, вузька спеціалізація лише на рекомендаційних системах
4.	Зростання лояльності користувачів до Інтернет-ресурсу	Пропонування персоналізованого контенту	

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та послуги, його фізичні складові, особливості процесу його надання (таблиця 5.19).

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
І. Товар за задумом	Товар забезпечує клієнтів механізмом індивідуального підбору контенту для користувачів їх системами, послугами, товарами, що збільшує обсяги продажів та лояльність кінцевих користувачів

Продовження таблиці 5.19

Рівні товару	Сутність та складові
II. Товар у реальному виконанні	Товар представляє собою програмний комплекс з декількох програмних артефактів, що інтегруються між собою, клієнтськими системами та базами даних
	Програмні артефакти поставляються запакованими в архіви .jar та як .py скрипти
III. Товар із підкріпленням	До продажу: відбувається інсталяція та конфігурування системи, проводяться тренінги для клієнта
	Після продажу: відбувається підтримка програмного забезпечення та його допрацювання за потреби клієнта
Програмні артефакти розповсюджуються з вбудованою ліцензією, також можливе патентування структури нейронних моделей	

Таблиця 5.20 – Визначення меж встановлення ціни

№	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	Від 10000$	Від 25000$	> 1млн $/рік	Нижня межа: 20000$ за впровадження Верхня межа: 40000$ за впровадження

Аналіз системи збуту передбачає визначення ефективності кожного елемента цієї системи, оцінювання діяльності апарату працівників збуту. Аналіз витрат обігу передбачає зіставлення фактичних збутових витрат за кожним каналом збуту і видом витрат із запланованими показниками для того, щоб виявити необґрунтовані витрати, ліквідувати затрати, що виникають у процесі руху товарів і підвищити рентабельність наявної системи збуту.

Дані щодо визначення системи збуту надаються в таблиці 4.21.

Таблиця 5.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Лояльність до відомих назв та компаній з високою репутацією, а також тяжіння до низьких цін	Продаж, впровадження і підтримка	Канал нульового рівня	Прямий маркетинг

У якості концепції маркетингових комунікацій були обрані інтегровані маркетингові комунікації. Важливо повністю обдумувати синхронізувати та координувати дії по всіх напрямках каналів комунікацій. Опір робиться на прямий канал збуту та репутацію. Розрахунок на сарафанне радіо.

Таблиця 5.22 – Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Поведінка B2B (особисте знайомство, побудова довготривалих стосунків)	Медіа Теплі та холодні дзвінки Конференції та тематичні заходи	1.Підвищення обсягів продажів. 2.Оптимізація закупок. 3.Задоволений клієнт.	Збільшення продажів, швидке та недороге впровадження	Висвітлити ефективність рішення та його значущість для розвитку бізнесу та встановлення переваги над конкурентами

Важливо звернути увагу, що цільовий сегмент проекту це B2B, що впливає на характер комунікацій та побудову маркетингової компанії. Більш ефективними будуть особисті комунікації з представниками компаній, виступи на тематичних конференціях та заходах. Велику роль відіграватиме репутація компанії, тому з перших замовлень потрібно буде підтримувати високу якість товару та бути клієнт-орієнтованими.

5.6 Висновки

В цьому розділі було проведено маркетинговий аналіз з метою визначення можливості та доцільності ринкової комерціалізації проекту інтелектуальної системи підбору клієнтського контенту.

Результати дослідження свідчать про можливість ринкової комерціалізації, що обґрунтовується позитивною динамікою нового, ще не до кінця сформованого, ринку, потенціал якого досить значний, судячи з західних більш розвинутих регіонів світу.

При комерційній реалізації проекту можуть стати на заваді економічне та правове становище в країні, проте проект досить легко буде поширювати і за межами України.

При побудові маркетингової компанії варто спиратися на прямий канал збуту нульового рівня, та висвітлювати ефективність рішень такого роду, унікальність рішення та значущість впровадження для встановлення конкурентної переваги, а також на швидке та легке впровадження та інтеграцію.

Підсумок маркетингового аналізу вказує на доцільність подальшої реалізації проекту.

ВИСНОВКИ

У роботі для побудови програмного комплексу інтелектуальної системи підбору клієнтського контенту було проаналізовано існуючі рішення у сфері рекомендаційних систем, існуючі підходи до вирішення задачі прогнозування, розглянуто їх недоліки та проблеми з якими стикаються при розробці подібних систем, та можливі способи вирішення, а також було розроблено комбіновану математичну модель для прогнозування майбутніх взаємодій користувача.

Були виконані наступні завдання:

- досліджено існуючі алгоритми прогнозування клієнтських взаємодій;
- досліджено існуючі підходи фільтрування контенту в рекомендаційних системах;
- розроблено комбіновану математичну модель для прогнозування користувацької взаємодії, що враховує історичні дані;
- розроблено систему підбору клієнтського контенту на базі запропонованої моделі;
- досліджено ефективність запропонованого рішення.

За результатами досліджень проведених у роботі було для реалізації системи було обрано комбіновану модель алгоритму прогнозування на базі нейронних мереж довготривалої короткочасної пам’яті зі звичайними мережами прямого розповсюдження та моделлю матричного розкладу, бо модель показала кращі результати, це також свідчить про виправданість такого способу об’єднання.

Особливостями реалізованої системи є те, що вона оцінює користувацькі взаємодії із групами об’єктів одночасно, тобто обробляє одразу данні взаємодії із кожним входження до замовлення – групові взаємодії. Також система враховує історичні дані послідовності замовлень, як групових взаємодій.

В результаті виконання завдань роботи було побудовано програмний комплекс системи інтелектуального підбору клієнтського контенту, що може легко масштабуватися та нарощувати функціонал. Система розроблена таким

чином, що здатна інтегруватися з клієнтськими системами. Архітектура системи розроблена так, що рішення є більш універсальним у порівнянні з існуючими системами та може бути легко адаптованим під інтеграції з різними системами клієнтів. Систему була протестована на трьох рівнях, а також описано механізм роботи системи, її структура та архітектура.

Результати розробок та досліджень роботи були використані при розробці реальної системи, що впроваджена в експлуатацію, що підтверджує практичне значення одержаних результатів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Buder J. Learning with personalized recommender systems: A psychological view / J. Buder, C. Schwind. // Computers in Human Behavior. – 2012. – №28. – С. 207–216.

2. Funk S. FunkSVD proposal [Електронний ресурс] / Simon Funk. – 2006. – Режим доступу до ресурсу: http://sifter.org/~simon/journal/20061211.html.

3. Nielsen M. Neural networks and deeplearning [Електронний ресурс] / Michael Nielsen. – 2018. – Режим доступу до ресурсу: http://neuralnetworksanddeeplearning.com/chap1.html.

4. Sigmoid function [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Sigmoid_function.

5. Understanding LSTM Networks [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

6. Begio Y. Learning Long-term dependencies with gradient descent is difficult [Електронний ресурс] / Y. Begio, P. Simard, P. Frasconi – Режим доступу до ресурсу: http://ai.dinfo.unifi.it/paolo/ps/tnn-94-gradient.pdf

7. Hochreiter S. Long-short term memory [Електронний ресурс] / Sepp Hochreiter. – 1997. – Режим доступу до ресурсу: http://www.bioinf.jku.at/publications/older/2604.pdf.

8. Cho K. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation [Електронний ресурс] / K. Cho, D. Bahdanau – Режим доступу до ресурсу: https://arxiv.org/pdf/1406.1078v3.pdf.

9. Микросервисная архитектура, Spring Cloud и Docker [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://habr.com/post/280786/.

10. A generic coordinate descent framework for learning from implicit feedback [Електронний ресурс] / I.Bayer, X. He, B. Kanagal, S. Rendle. – 2017. – Режим доступу до ресурсу: https://arxiv.org/abs/1611.04666.

11. Fast matrix factorization for online recommendation with implicit feedback [Електронний ресурс] / X.He, H. Zhang, M. Kan, T. Chua – Режим доступу до ресурсу: https://arxiv.org/abs/1708.05024.

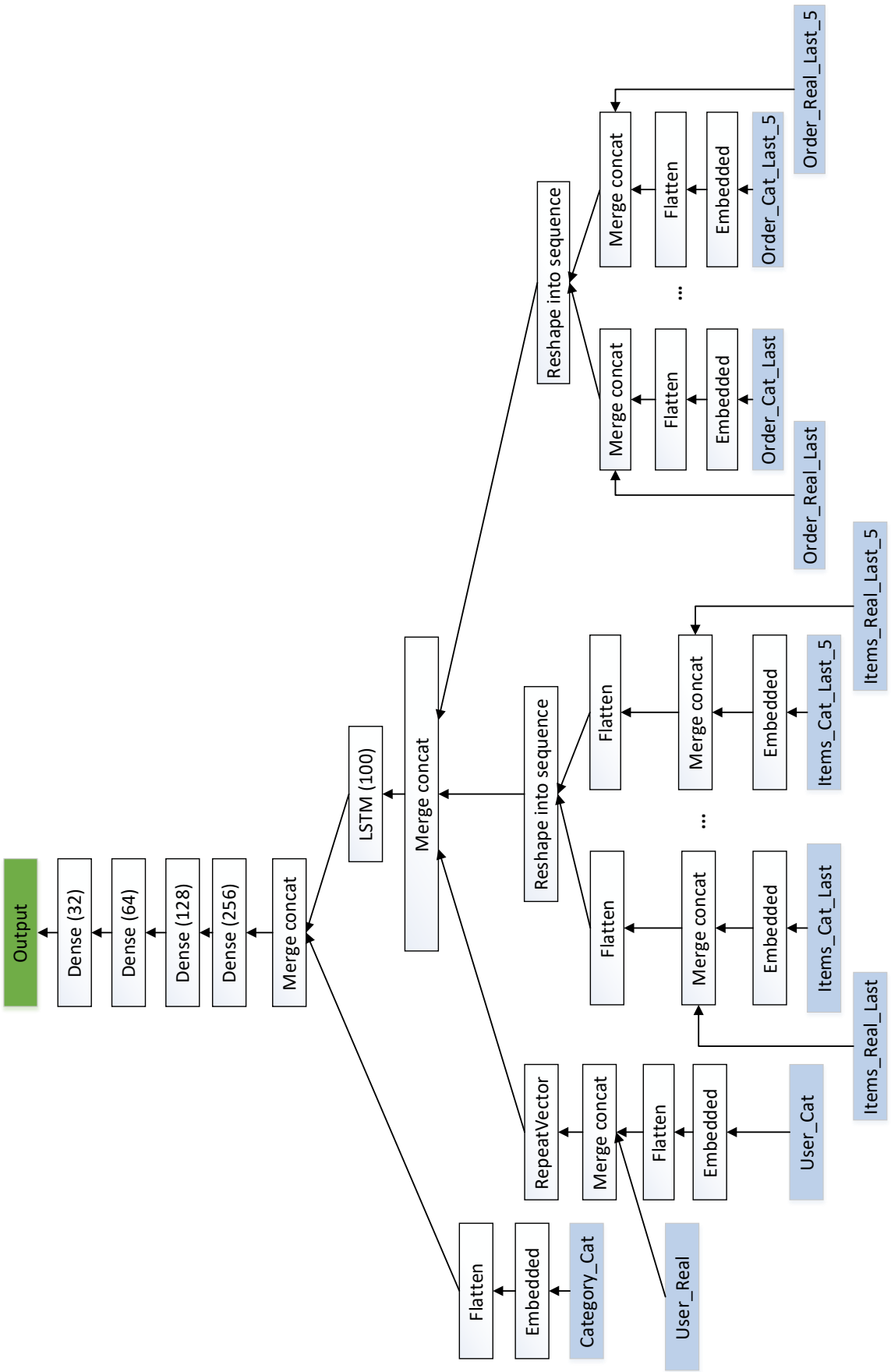
12. Bayesian personalized ranking from implicit feedback [Електронний ресурс] / S.Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme. – 2009. – Режим доступу до ресурсу: https://arxiv.org/abs/1205.2618.

13. Главные тенденции украинского рынка CRM-систем [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://crosssellguide.com/glavnye-tendentsii-ukrainskogo-rynka-crm-sistem/.

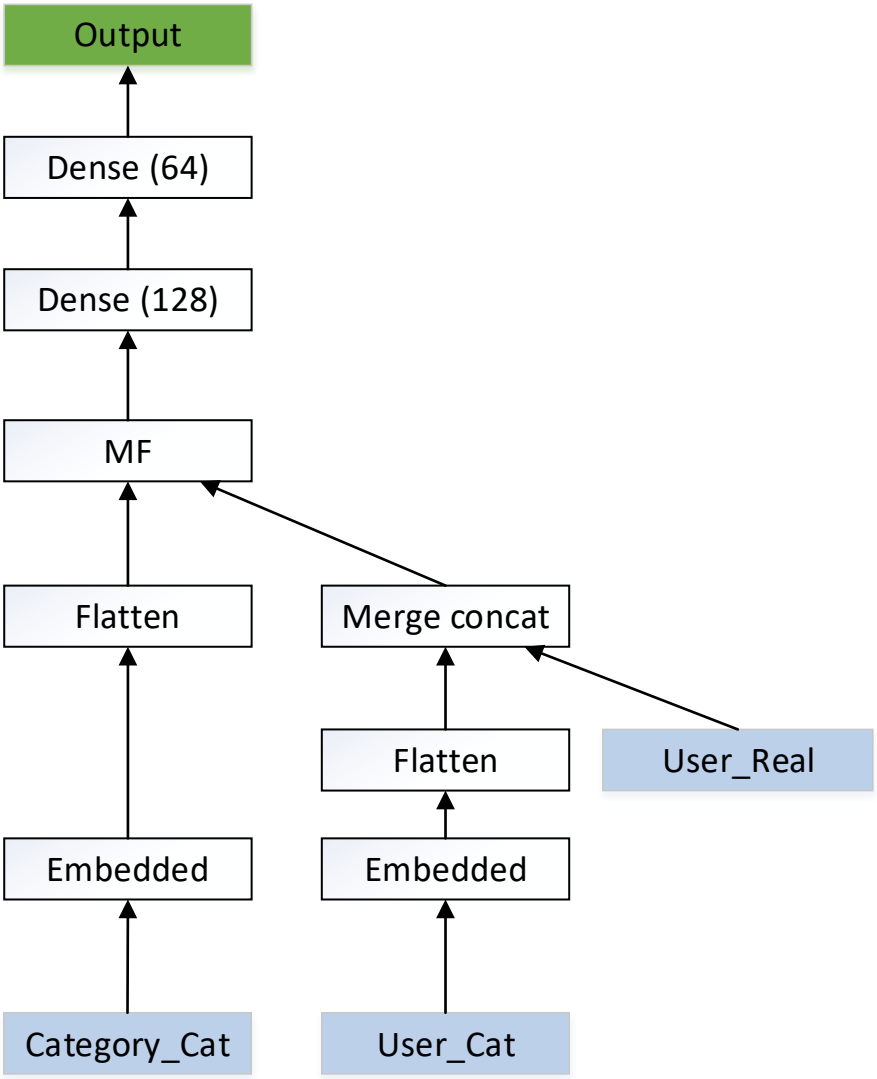
14. CRM-системы стали крупнейшим сегментом рынка ПО в 2017 году [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://news.finance.ua/ru/news/-/424278/crm-sistemy-stali-krupnejshim-segmentom-rynka-po-v-2017-godu.

15. Маркетингові програми [Електронний ресурс] – Режим доступу до ресурсу: https://pidruchniki.com/1263111349596/marketing/marketingovi_programi.

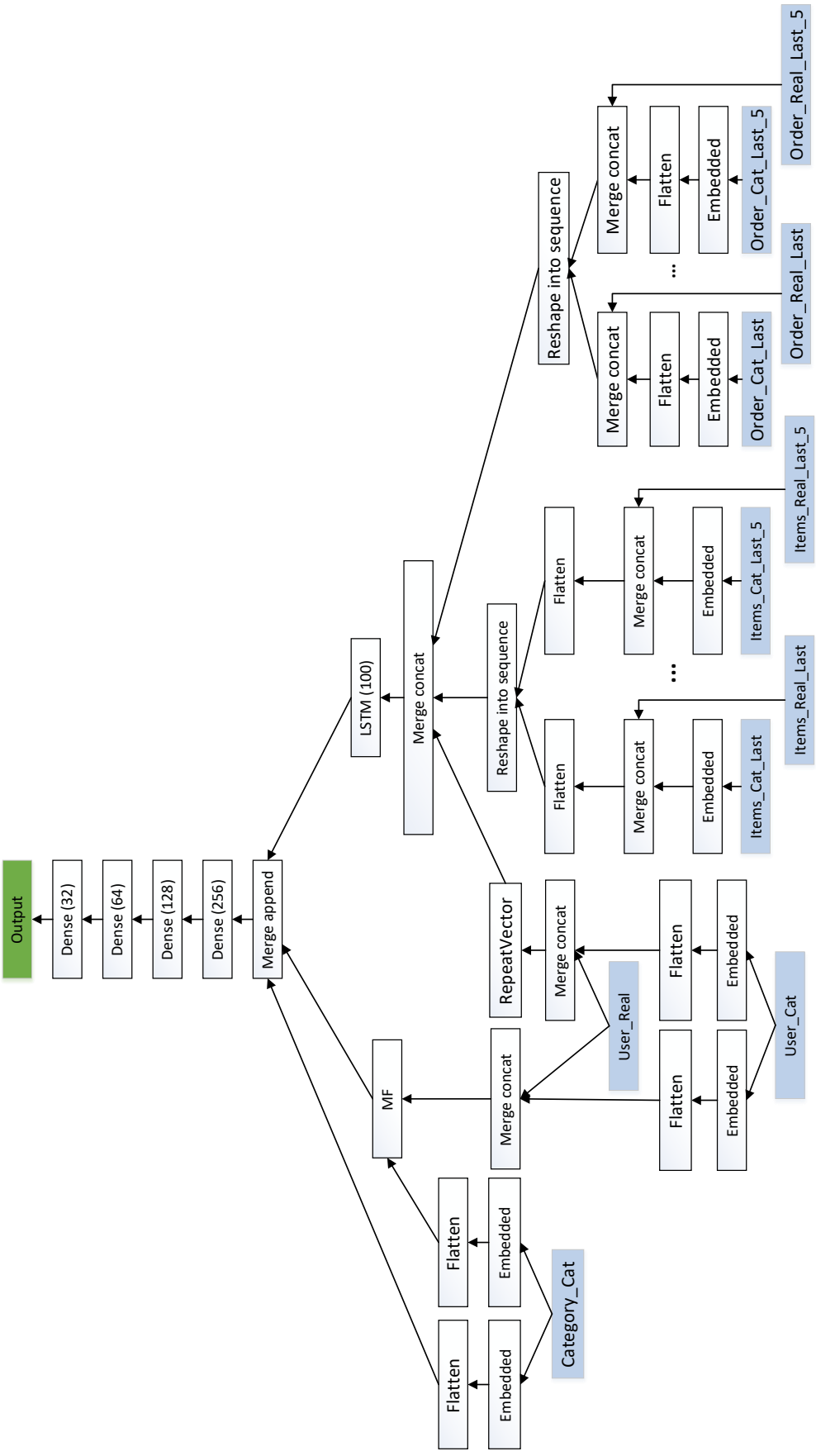
ДОДАТОК А


Output
Dense (32)
Dense (64)
Dense (128)
Dense (256)
Merge concat
LSTM (100)
Merge concat
Reshape into sequence
Merge concat
Flatten
Embedded
Order_Cat_Last_5
Order_Real_Last_5
Flatten
Embedded
Order_Cat_Last
Order_Real_Last
Flatten
Merge concat
Embedded
Items_Cat_Last_5
Items_Real_Last_5
Flatten
Merge concat
Embedded
Items_Cat_Last
Items_Real_Last
RepeatVector
Merge concat
Flatten
Embedded
User_Cat
Flatten
Embedded
Category_Cat
User_Real

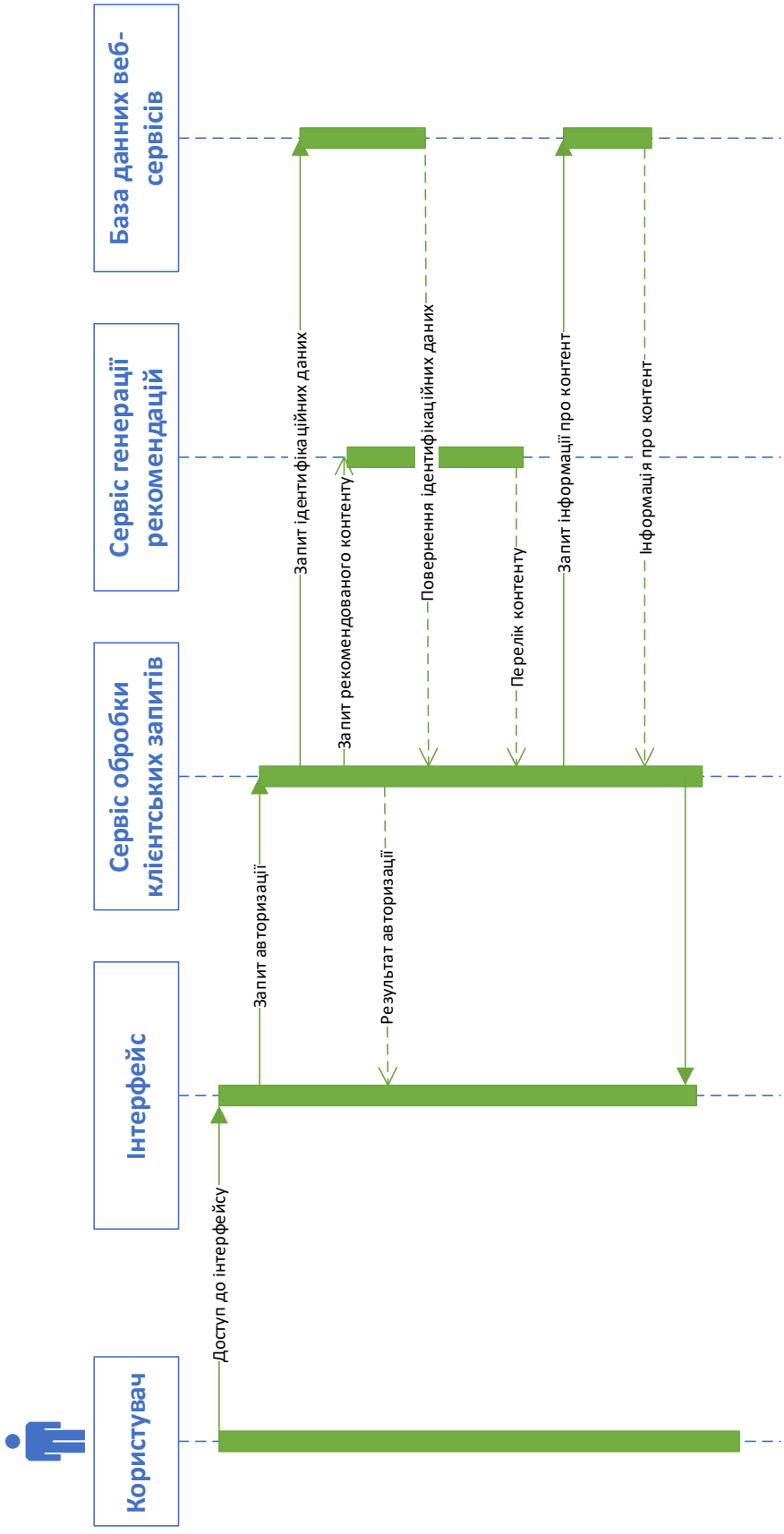
ДОДАТОК Б


Output
Dense (64)
Dense (128)
MF
Flatten
Merge concat
Flatten
User_Real
Embedded
Embedded
Category_Cat
User_Cat

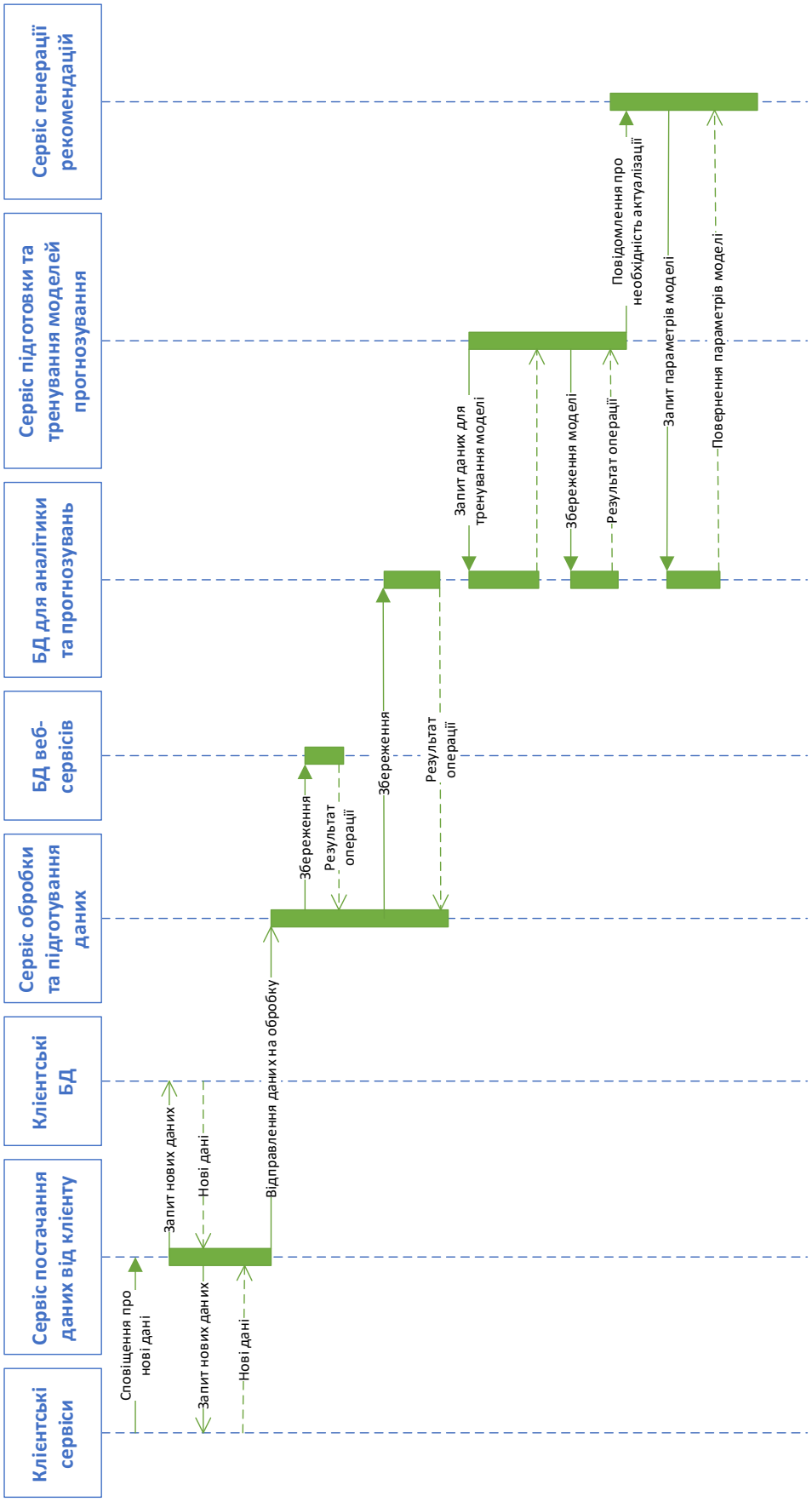
ДОДАТОК В


Output
Dense (32)
Dense (64)
Dense (128)
Dense (256)
Merge append
LSTM (100)
Merge concat
Reshape into sequence
Flatten
Merge concat
Embedded
Items_Cat_Last_5
Items_Real_Last_5
Items_Cat_Last
Items_Real_Last
Reshape into sequence
Flatten
Merge concat
Embedded
Order_Cat_Last_5
Order_Real_Last_5
Order_Cat_Last
Order_Real_Last
RepeatVector
Merge concat
User_Real
Flatten
Embedded
User_Cat
MF
Merge concat
Flatten
Embedded
Flatten
Embedded
Category_Cat

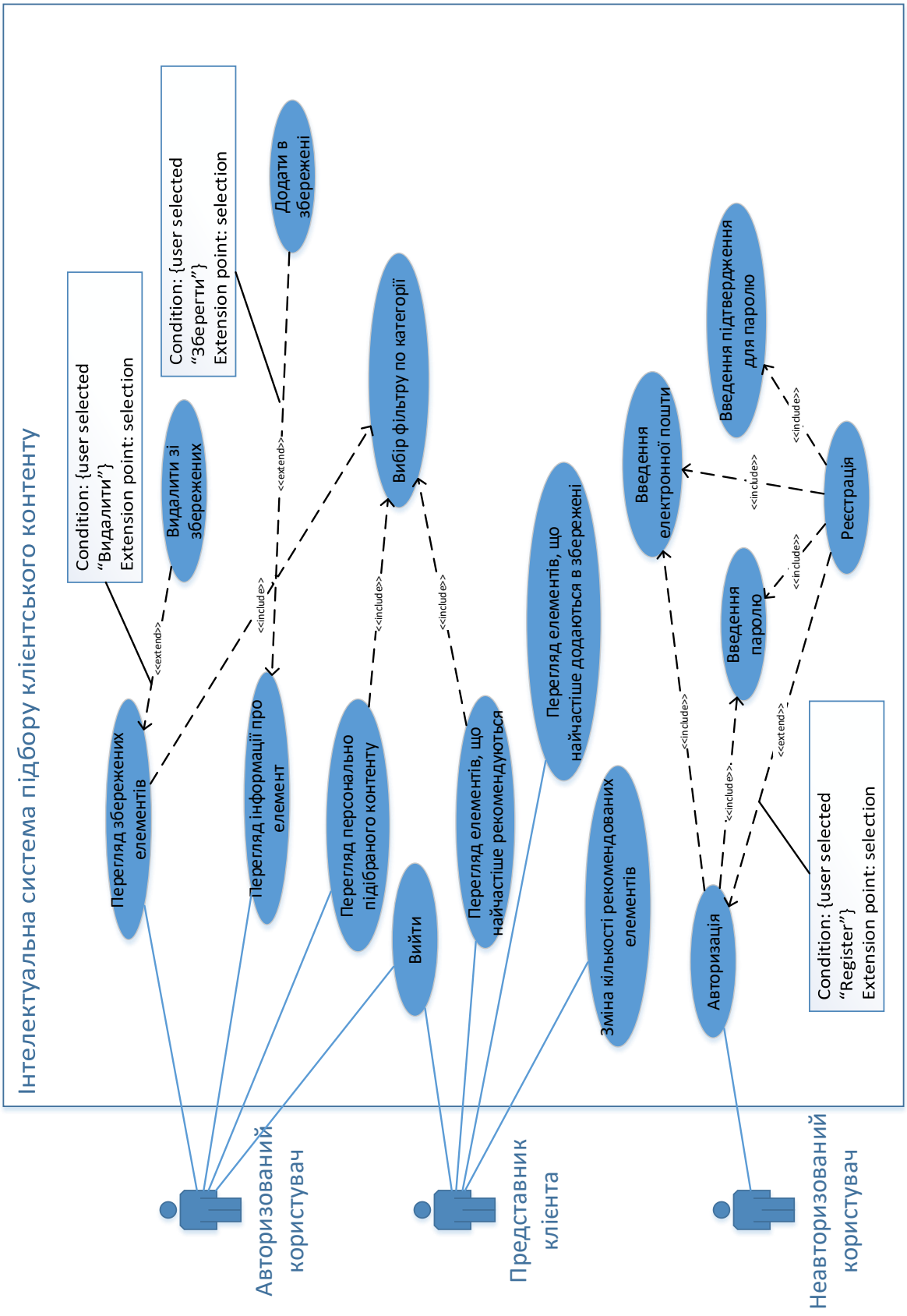
ДОДАТОК Г


Користувач
Інтерфейс
Сервіс обробки клієнтських запитів
Сервіс генерації рекомендацій
База данних веб-сервісів
Доступ до інтерфейсу
Запит авторизації
Результат авторизації
Запит ідентифікаційних даних
Запит рекомендованого контенту
Повернення ідентифікаційних даних
Перелік контенту
Запит інформації про контент
Інформація про контент

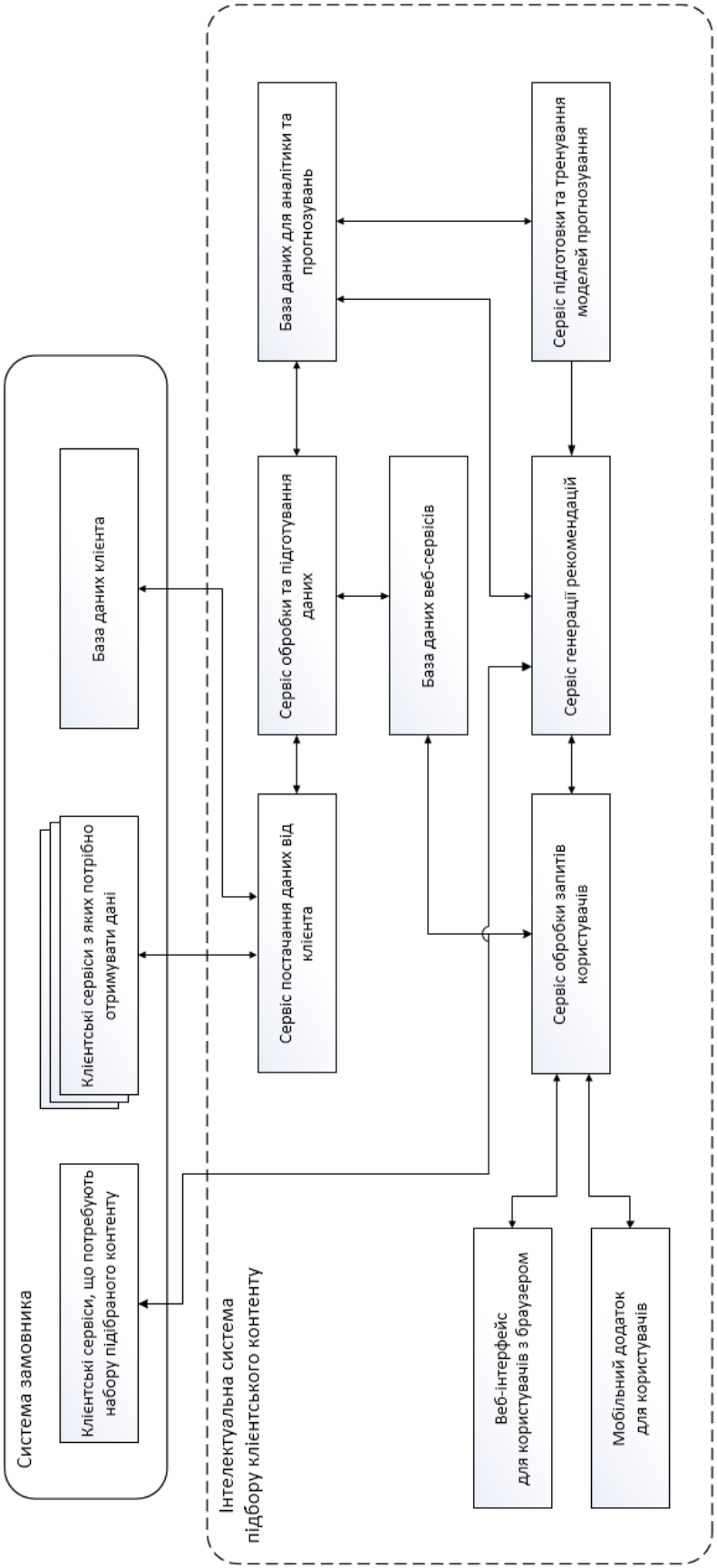
ДОДАТОК Д


Клієнтські сервіси
Сервіс постачання даних від клієнту
Клієнтські БД
Сервіс обробки та підготування даних
БД веб-сервісів
БД для аналітики та прогнозувань
Сервіс підготовки та тренування моделей прогнозування
Сервіс генерації рекомендацій
Сповіщення про нові дані
Запит нових даних
Нові дані
Відправлення даних на обробку
Збереження
Результат операції
Збереження
Результат операції
Запит даних для тренування моделі
Збереження моделі
Результат операції
Повідомлення про необхідність актуалізації
Запит параметрів моделі
Повернення параметрів моделі

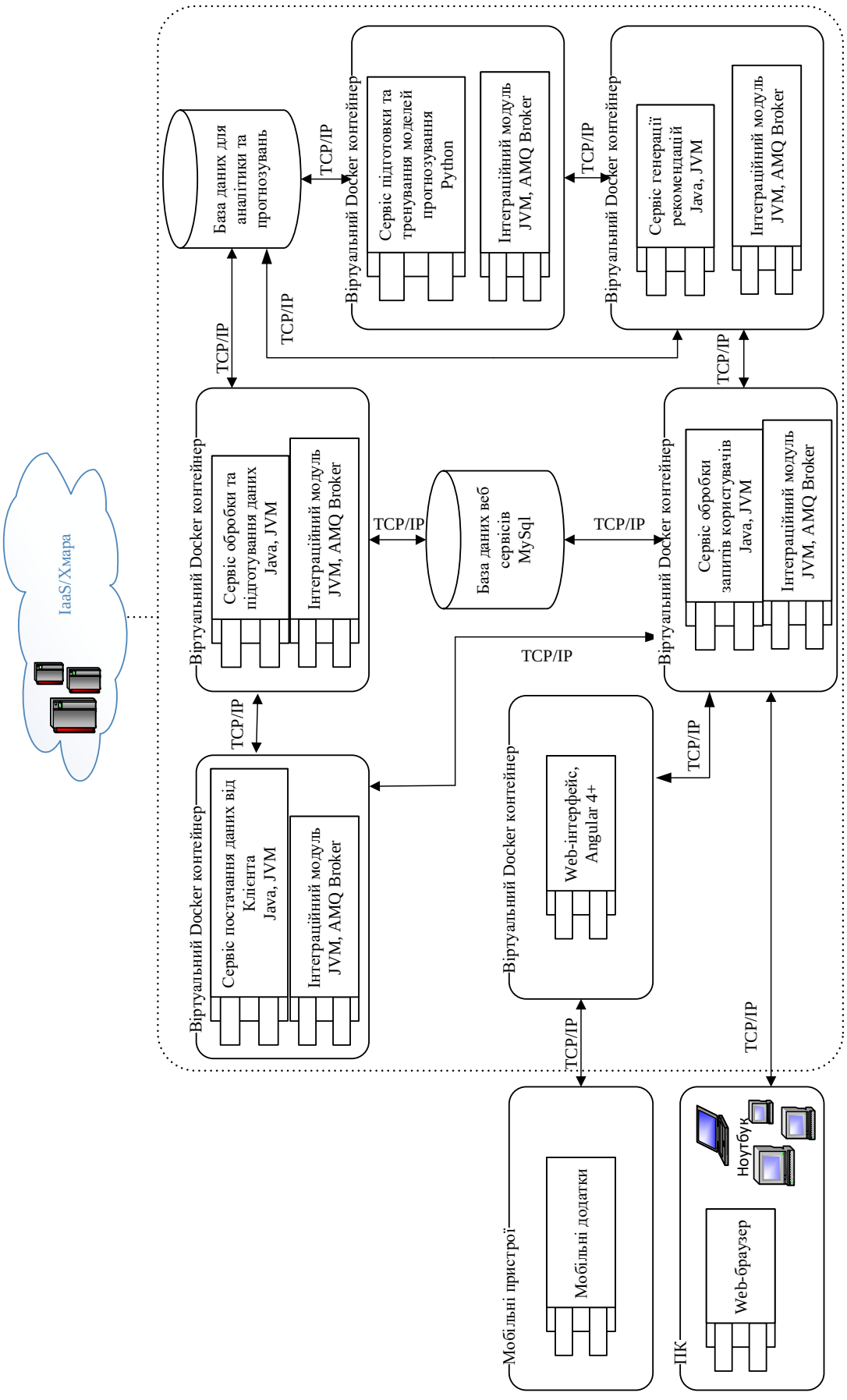
ДОДАТОК Е


Інтелектуальна система підбору клієнтського контенту
Condition: {user selected
"Видалити"}
Extension point: selection
Condition: {user selected
"Зберегти"}
Extension point: selection
Додати в збережені
Видалити зі збережених
Вибір фільтру по категорії
Введення підтвердження для паролю
Введення електронної пошти
Перегляд збережених елементів
Перегляд інформації про елемент
Перегляд персонально підібраного контенту
Перегляд елементів, що найчастіше рекомендуються
Перегляд елементів, що найчастіше додаються в збережені
Введення паролю
Реєстрація
Вийти
Зміна кількості рекомендованих елементів
Авторизація
Condition: {user selected
"Register"}
Extension point: selection
<<extend>>
<<include>>
Авторизований користувач
Представник клієнта
Неавторизований користувач

ДОДАТОК Ж


Система замовника
Клієнтські сервіси, що потребують набору підібраного контенту
Клієнтські сервіси з яких потрібно отримувати дані
База даних клієнта
Інтелектуальна система підбору клієнтського контенту
Сервіс постачання даних від клієнта
Сервіс обробки та підготування даних
База даних для аналітики та прогнозувань
База даних веб-сервісів
Сервіс підготовки та тренування моделей прогнозування
Сервіс генерації рекомендацій
Сервіс обробки запитів користувачів
Веб-інтерфейс для користувачів з браузером
Мобільний додаток для користувачів

ДОДАТОК И


IaaS/Хмара
Віртуальний Docker контейнер
Сервіс обробки та підготування даних Java, JVM
Інтеграційний модуль JVM, AMQ Broker
Віртуальний Docker контейнер
Сервіс підготовки та тренування моделей прогнозування Python
Інтеграційний модуль JVM, AMQ Broker
Віртуальний Docker контейнер
Сервіс генерації рекомендацій Java, JVM
Інтеграційний модуль JVM, AMQ Broker
Віртуальний Docker контейнер
Сервіс обробки запитів користувачів Java, JVM
Інтеграційний модуль JVM, AMQ Broker
Віртуальний Docker контейнер
Сервіс постачання даних від Клієнта Java, JVM
Інтеграційний модуль JVM, AMQ Broker
Віртуальний Docker контейнер
Web-інтерфейс, Angular 4+
База даних для аналітики та прогнозувань
База даних веб сервісів MySql
Мобільні пристрої
Мобільні додатки
ПК
Ноутбук
Web-браузер
TCP/IP

ДОДАТОК К

Category_Cat	Категоріальні ознаки підкатегорії, в даному випадку ідентифікатор категорії
User_Real	Реальні ознаки користувача
User_Cat	Категоріальні ознаки користувача
Items_Cat_Last	Категоріальні ознаки елементів, що входили до останнього замовлення
Items_Real_Last	Реальні ознаки
Items_Cat_Last_5	Категоріальні ознаки елементів, що входили до 5-го по черзі останнього замовлення
Items_Real_Last_5	Реальні ознаки ознаки елементів, що входили до 5-го по черзі останнього замовлення
Order_Cat_Last	Категоріальні ознаки останнього замовлення
Order_Real_Last	Реальні ознаки останнього замовлення
Order_Cat_Last_5	Категоріальні ознаки 5-го по черзі останнього замовлення
Order_Real_Last_5	Реальні ознаки 5-го по черзі останнього замовлення