Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____*Комп'ютерних наук*_____
(повна назва)

Кафедра _____*Системотехніки*_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

_____Дослідження методів проєктування систем_____
_____рекомендації товарів_____
(тема)

Виконав:
студент _2_ курсу, групи _*СПРм-22-1*_____
_____*Новіков М.В.*_____
(прізвище, ініціали)

Спеціальність __*122 Комп'ютерні науки*_____
_____
(код і повна назва спеціальності)
Тип програми _____*освітньо-наукова*_____
Освітня програма _*Системне проєктування*__
_____
(повна назва освітньої програми)

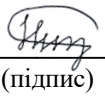
Керівник _____*проф. Міщеряков Ю.В.*_____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ ___*Гребеннік І.В.*___
(підпис) (прізвище, ініціали)

20*24* р.

Я як студент(ка) ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.



_____ *Новіков М.В.*
(підпис) (прізвище, ініціали)

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено 21 червня 2024 р.

Керівник кваліфікаційної роботи _____ проф. Міщеряков Ю.В.
(підпис) (посада, прізвище, ініціали)

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*

Кафедра *Системотехніки*

Рівень вищої освіти *другий (магістерський)*

Спеціальність *122 Комп'ютерні науки*
(код і повна назва)

Тип програми *освітньо-наукова*

Освітня програма *Системне проєктування*
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«____»_____ 20___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові *Новікову Микиті Валерійовичу*
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів проєктування систем рекомендації товарів

затверджена наказом університету від *01 квітня* 2024 р. № *259Ст*

2. Термін подання студентом роботи до екзаменаційної комісії *21 червня* 2024 р.

3. Вихідні дані до роботи *Об'єкт дослідження – методи рекомендації товарів. Предмет дослідження – властивості рекомендаційних методів. Технічне забезпечення: ІВМ-сумісний персональний комп'ютер.*

4. Перелік питань, що потрібно опрацювати в роботі *Аналіз предметної області. Дослідження методів формування рекомендацій. Дослідження гібридних методів формування рекомендацій.*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій *Схеми алгоритмів рекомендаційних систем на основі*

пам'яті, схеми рекомендаційних систем на основі моделі, схеми схеми гібридних рекомендаційних систем, графіки із результатами експериментів.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	*Отримання завдання на виконання роботи*	*01.04.2024*	
2	*Аналіз предметної області рекомендацій товарів*	*02-6.04.2024*	
3	*Аналіз задачі створення гібридних рекомендаторів*	*6-10.04.2024*	
4	*Постановка задачі*	*11-13.04.2024*	
5	*Визначення умов дослідження*	*14-18.04.2024*	
6	*Визначення стандартизованих рішень для специфічних задач*	*19-23.04.2024*	
7	*Проєктування рекомендаційних моделей*	*24.04-31.04.2024*	
8	*Реалізація моделей та проведення тестування*	*01-11.05.2024*	
9	*Аналіз результатів експерименту над моделями*	*12-14.05.2024*	
10	*Визначення цілей та умов дослідження гібридних рекомендаційних систем*	*15-20.05.2024*	
11	*Проєктування гібридних рекомендаційних систем*	*21-26.05.2024*	
12	*Реалізація та тестування гібридних моделей*	*26.05-04.05.2024*	
13	*Аналіз результатів експерименту над гібридами*	*05-07.06.2022*	
14	*Оформлення пояснювальної записки*	*08-15.06.2022*	
15	*Представлення на рецензування*	*22.06.2022*	

Дата видачі завдання *24 березня* 2024 р.

Студент _____
(підпис)

Керівник роботи _____ *проф. Міщеряков Ю.В.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Робота містить: 77 сторінок, 17 рисунків, 1 таблицю, 2 додатки, 30 джерел.

МЕТОДИ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ, ПРОДАЖ ТОВАРІВ, ДОСЛІДЖЕННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ, ГІБРИДНІ РЕКОМЕНДАЦІЙНІ СИСТЕМИ, ГЛИБОКЕ НАВЧАННЯ.

Об’єктом дослідження є методи рекомендації товарів.

Предметом дослідження є властивості рекомендаційних методів.

Метою дослідження є проєктування рекомендаційної системи, що має найбільш оптимальні показники ефективності.

Методи дослідження – методи системного проєктування, методи формування рекомендацій, методи машинного навчання, методи глибокого навчання, методи статистичного аналізу, методи математичного моделювання.

В роботі проводиться аналіз предметної області рекомендацій товарів, визначаються умови дослідження, визначаються вимоги до рекомендаційних систем, розглядається специфіка формування рекомендацій для продажу товарів, проводиться проєктування, тестування та аналіз підходів до реалізації рекомендаційних систем, проводиться проєктування, дослідження та оцінка варіантів реалізації гібридних рекомендаційних систем.

Сфера застосування – продаж товарів через мережу Інтернет.

ABSTRACT

Thesis contains: 77 pages, 17 images, 1 table, 2 appendices, 30 references.

METHODS OF RECOMMENDATION GENERATION, SALE OF GOODS, RESEARCH OF RECOMMENDATION SYSTEMS, HYBRID RECOMMENDATION SYSTEMS, DEEP LEARNING.

The object of research is the methods of product recommendation.

The subject of research is the properties of recommendation methods.

The aim of the research is to design a recommendation system with the most optimal performance indicators.

Research methods include systems design methods, recommendation generation methods, machine learning methods, deep learning methods, statistical analysis methods, and mathematical modeling methods.

The work analyzes the subject area of product recommendations, defines the research conditions, establishes the requirements for recommendation systems, discusses the specifics of forming recommendations for the sale of goods, design, testing, and analysis of approaches to implementing recommendation systems, and explores the design, research, and evaluation of hybrid recommendation system implementations.

Scope of application – online sales of goods.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Датасет – набір даних;

Ембединг – представлення об’єкту у вигляді багатовимірного вектору;

Рекомендатор – рекомендаційна система;

Факторизація – розкладання;

CB – системи на основі контенту;

CF – колаборативна фільтрація;

FM – машина факторизації;

GMF – узагальнена матрична факторизація;

MF – матрична факторизація;

РС – рекомендаційна система.

ВСТУП

На сьогодні, в умовах постійного росту інформаційних технологій та впровадження їх в щоденний побут людей, особливо велике значення відводиться електронній комерції та продажу товарів з використанням відповідних Інтернет-ресурсів. З урахуванням обсягів продукції, що реалізується, обсягів фінансових ресурсів, що виділяються на дану індустрію, та кількість даних, котру потенційному клієнту необхідно обробити в процесі взаємодії з відповідними сервісами, одним з найголовніших аспектів електронної комерції є рекомендаційні засоби.

За останнє десятиліття відбувся стрімкий розвиток рекомендаційних технологій, було опубліковано велику кількість робіт, що демонструють нові підходи до реалізації рекомендаційних систем. Зокрема, розвиток глибокого навчання дав змогу розвивати рекомендаційні алгоритми в напрямку орієнтації на нейромережеві моделі, та було запропоновано значний перелік підходів до проєктування. Саме тому, дослідження властивостей нових методів та практик проєктування з урахуванням традиційних підходів, дослідження можливості їх інтеграції є в значній мірі актуальним.

Дана робота присвячена дослідженню методів проєктування систем рекомендації товарів, як окремо, так й у якості компонентів комплексних гібридний систем.

Метою дослідження є проєктування рекомендаційної системи, що має найбільш оптимальні показники ефективності.

Об’єктом дослідження є методи рекомендації товарів.

Предметом дослідження є властивості рекомендаційних методів.

Методи дослідження – методи системного проєктування, методи формування рекомендацій, методи машинного навчання, методи глибокого навчання, методи статистичного аналізу, методи математичного моделювання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз поняття рекомендаційної системи

В контексті електронної комерції рекомендаційна система – інструмент, можливості якої полягають в аналізі великих масивів даних з метою надання клієнтам персональних рекомендацій. З точки зору користувача сервісу дана система допомагає в прийнятті рішень з покупки того чи іншого товару, дозволяє зменшити час, що відводиться на пошук, звернути увагу на нові позиції, знайти котрі в каталозі самостійно є малоймовірним. Для власника ж платформи функція даної системи полягає в збільшенні фінансових показників роботи підприємства, що є наслідком покращення користувацького досвіду, підвищення лояльності та середнього чеку, більш того, такі рекомендації стимулюють демократизацію популярності товарів.

Рекомендаційна система дозволяє на основі інформації про об’єкти, суб’єкти та результати діяльності підприємства робити певні прогнози з вподобань для конкретних клієнтів. Система аналізує взаємодію між клієнтом та товаром, та на основі цього прогнозує взаємодію в майбутньому [1]. Дані системи можуть бути реалізовані за різними принципами та алгоритмами, або ж поєднувати в собі більш ніж один з метою досягнення більш високих показників. У даній області основними з них є точність та швидкодія. Швидкість генерації рекомендацій зменшується зі зростанням кількості користувачів й обсягу даних, та з урахуванням того, що рекомендації надаються в переважній більшості прецедентів бізнес-процесу, у разі високого навантаження на комерційну систему, час відгуку серверу може зрости до неприпустимих значень. Точність у свою чергу є більш комплексною проблемою, що має в собі велику кількість підпроблем, однією з котрих є холодний старт.

Формування ж пропозицій у даних умовах у свою чергу може дуже негативно вплинути на ефективність системи при експлуатації. Більш того,

проблема холодного старту є особливо характерною для наявної предметної області – на відміну від рекомендацій, наприклад, на платформах пов’язаних з переглядом кіно або відвідуванням локацій, переважна більшість рекомендацій товарів відбувається в умовах повного знаходження або близькості до холодного старту для користувача.

В області РС можна виділити декілька різних за принципом класів простих систем, кожна має різні властивості та по-різному демонструє себе у різних умовах таких як холодний старт користувача, вкрай великі обсяги даних, постійне оновлення переліку товарів, тощо. Нижче наведені основні зі згаданих класів систем.

– Колаборативна фільтрація (Collaborative filtering) – формування рекомендацій на основі історії уподобань «схожих» користувачів, іншими словами аналіз взаємодії користувачів з товарами;

– Контентно-орієнтовані рекомендації (Content-based systems) – використання інформації про зміст товару, тобто певний набір його характеристик, з метою відбору прийнятних для профілю користувача з можливістю урахування певного характеристико-орієнтованого контексту;

– Методи на основі знань (Knowledge-based systems) – використання експертних знань з предметної області з метою побудови з їх використанням рекомендацій на основі даних про покупців та товари.

Реалізація кожного з даних підходів може відрізнятись в залежності від інтерпретації, котра обрана при проєктуванні. Так, наприклад, система може бути представлена у вигляді довільного покрокового алгоритму, операцій над матрицями, класифікацією з використанням машинного навчання, тощо. Реалізація може також різнитися в залежності від задачі, що виконує система – що саме вона прогнозує, в якій кількості, які базові та додаткові дані приходять на вхід, тощо. Проте, незалежно від тих чи інших алгоритмів чи методів, що використовуються в системі, вони всі працюють на основі одного з базових підходів, що визначає принцип побудови рекомендацій.

1.2 Аналіз підходів до створення рекомендацій

1.2.1 Аналіз методу колаборативної фільтрації

Серед методів формування рекомендацій колаборативна фільтрація, як вже було зазначено, використовує інформацію про подібність об’єктів, що ґрунтується на взаємодії між користувачами та товарами. В основі даного класу алгоритмів – припущення про існування «подібності», тобто припущення, що користувачі, подібні за переліком позицій та характером взаємодії з ними, схильні в майбутньому обирати схожі товари [2].

У колаборативній фільтрації можна виділити два підходи, що розрізняються за об’єктом аналізу подібності:

a) Користувацька фільтрація – визначення найбільш подібних до клієнта користувачів та побудова рекомендацій на основі їх вподобань. Даний варіант передбачає більшу персоналізацію, проте має низький рівень масштабованості: зі зростанням кількості користувачів в системі стрімко зростає й складність відповідних обчислень. Тому така система є найбільш ефективною у разі, якщо кількість товарів більше кількості користувачів. Крім того, задача пошуку схожих за інтересами користувачів є складнішою, що значно посилюється в умовах холодного старту користувача;

b) Об’єктна фільтрація – визначення на основі товарів, з котрими взаємодіяв клієнт, найбільш подібних з точки зору рейтингу та переглядів. Даний підхід доцільно використовувати у середовищі, де кількість об’єктів значно нижче за кількість користувачів в системі, та при вдалій реалізації такий рекомендатор буде мати в середньому меншу похибку. Об’єктний підхід передбачає більшу масштабованість та стабільність – постійно зростаюча кількість клієнтів має посередній вплив на роботу алгоритму, модель на основі подібності об’єктів є менш динамічною, що є перевагою у разі постійного зростання компанії. Більш того матриця подібності об’єктів є в рази меншою, ніж в альтернативному підході, тому вимагає менше місця для зберігання, а

відповідні обчислення накладають на систему набагато менше навантаження. Проте очевидний недолік такої фільтрації – рекомендації, що в кінці отримуються клієнтом, є в значній мірі менш персоніфікованими [3].

Таким чином, у загальному вигляді принцип колаборативної фільтрації для прогнозування уподобань спирається виключно на статистику, абстрагуючись від змісту об’єкту. Частіше за все, правильні рекомендації знаходяться за межами простої подібності товарів та є більш комплексним питанням, котре обумовлюється великою кількістю відповідних тенденцій та потребує глибокого аналізу поведінки користувачів. Колаборативна фільтрація надає досить простий проте ефективний спосіб такого аналізу, котрий не потребує ні побудови алгоритмів подібності товарів, ні попередньої роботи експертів, та має задовільну точність, якщо розглядати підходи окремо – вищу ніж на основі контенту. Проте така РС є вкрай вразливою для холодного старту – при доданні нового товару або реєстрації нового користувача вона не буде мати достатньо інформації про них для визначення «схожих» позицій та подальшого формування, у даній ситуації можливості системи обмежені в порівнянні з альтернативними підходами.

1.2.2 Аналіз контентно-орієнтованого методу

Метод формування рекомендацій на основі контенту передбачає аналіз атрибутів чи опису товарів: категорії, ціна, текст опису, перелік характеристик, числових, категоріальних, ключових слів, тощо, з метою визначення найбільш потенційно задовільних з них. Для формування переліку товарів останні порівнюються на відповідність «профілю» або «моделі вподобань» користувача, що являє собою структурований опис вподобань клієнта на основі його попередньої діяльності в системі. Даний профіль частіше за все представлений в наступному вигляді: вектор атрибутів – представлення профілю у вигляді вектору, де кожна позиція визначає ступінь інтересу

користувача до того чи іншого атрибуту, та являє собою відповідний ваговий коефіцієнт.

Виходячи з усього вищеописаного, можна зробити висновок, що content-based фільтрація є виключно персоніфікованим підходом, що при роботі звертає увагу не на загальні тенденції по взаємодіям користувачів з товарами, а виключно на взаємодію з ними цільового користувача. Зворотній бік полягає в тому, що результат є вкрай передбачуваним, у переліку товарів, що надаються потенційному клієнту, відсутні будь-які неочікувані випадкові та нові для нього елементи. При цьому з технічної точки зору метод потребує забезпечення механізмів для оновлення профілю при кожній новій взаємодії користувача з товарами.

У цілому, даний метод в більшості залежить від повноти та якості опису різних атрибутів товарів. Тому не зважаючи на простоту даного методу, при реалізації він іноді може потребувати додаткової підготовки даних. У деяких випадках цінні з точки зору аналізу атрибути можуть бути представлені у формі, що не дозволяє проведення жодних операцій: у тексті опису, на зображенні, тощо. З цієї причини в даних випадках доцільним є використання відповідних модулів для попереднього аналізу, ідентифікації та вилучення необхідної інформації. Крім того, деякі системи оперують об’єктами, що мають довільну структуру атрибутів – такі об’єкти потребують особливого підходу до представлення та порівняння, що важливо враховувати при проєктуванні рекомендаційної системи.

Джерело даних, на якому ґрунтується даний підхід робить його вкрай стійким до умов постійної зміни та росту даних. Так по-перше, при доданні нового товару він вже стає придатним до аналізу, алгоритм не потребує даних про те, хто його придбав та як оцінив, уся необхідна інформація – додається на старті адміністратором інформаційної системи. Більш того, при появі в базі системи нового споживача та придбанні одного чи двох товарів алгоритм вже дозволяє скласти мінімальний профіль та надавати в рекомендаціях релевантні позиції, тоді як для колаборативної фільтрації тої самої інформації було б

замало. Для створення початкового профілю у парі з content-based системами використовують механізми зворотного зв’язку: аналіз часу перегляду товарів, демографічні дані, експрес-опитування, інтерактивні рекомендації, тощо. Усі вищеописані особливості підходу роблять його в певній мірі стійким до холодного старту. Цією ж властивістю обумовлюється розповсюдженість практики його використання у якості доповнення до колаборативної фільтрації, тоді як через недостатньо високий показник точності його рідко використовують як самодостатню систему [4].

1.2.3 Аналіз методу на основі знань

Для формування рекомендацій Knowledge-based підхід використовує стратегію принципово відмінну у порівнянні з попередніми типами проєктування. На відміну від повного або майже повного приховування системи та важелів впливу на неї від користувача, що є характерним до систем «взаємодії user-item», система на основі знань надає йому можливості з контролю рекомендацій та керується чітко сформованими самим клієнтом набором вимог до товару.

Незалежно від способу реалізації, основу роботи даного алгоритму складають операції перевірки та зіставлення клієнтських вимог зі предметними знаннями (Domain knowledge) – схемою спеціальних логічних конструкцій, що умовно можна представити у вигляді тверджень «if-this-then-than». На практиці вони являють собою знання експертів з предметної області, з котрою пов’язана система, та виступають у якості фундаменту для побудови рекомендацій. Сукупність цих правил складає базу знань (Knowledge base), що зберігає необхідні для роботи алгоритму правила та періодично поповнюється особами, відповідальними за це. Коли система має закінчений перелік вимог, рекомендаційна система пропускає його через систему знань з метою ідентифікації в базі взаємозв’язків, що підпадають під відповідні обмеження.

У якості об’єкту аналізу алгоритми на основі знань можуть приймати різну інформацію. В залежності від того, з якою саме інформацією працює система, прийнято поділяти рекомендатори, що працюють за даним підходом, на дві категорії:

a) На основі обмежень – системі надається перелік обмежень щодо атрибутів об’єкту, що можуть являти собою відповідні границі для чисельних доменів, перелік допустимих значень для категоріальних доменів, тощо [5]. При цьому можливе накладання обмежень, що не відповідають напряму структурі атрибутів об’єкту, а пов’язані зі специфікою предметної області.

b) На основі прецедентів – системі надаються контрольні точки у вигляді набору товарів, та на основі подібності атрибутів цих точок ідентифікуються конкретні умови й подібні товари. В даному випадку знання предметної області містять певні метрики подібності, що й дозволяють оцінити схожість між товарами. Коли користувач формує нову задачу, система шукає подібні випадки, порівнюючи параметри нового запиту з параметрами збережених, адаптуючи знайдені рішення до нової задачі з деякими модифікаціями.

Найбільш позитивний аспект концепції розглянутого методу полягає в тому, що вона вкрай ефективно вирішує проблему холодного старту в найбільш нетривіальний спосіб. Алгоритм не виставляє жорстких вимог до історії взаємодії клієнта чи товару, може почати роботу як тільки буде наданий мінімальний перелік даних у вигляді вимог [6]. При цьому комплексний набір правил дозволяє зберегти при формуванні достатній рівень глибини та випадковості, сама ж база знань забезпечує гнучкість системи та можливість постійної адаптації до динамічних бізнес-умов та вимог компанії. База знань, проте, при всіх можливостях, що вона надає, виступає в якості слабкого місця системи – для продуктивної її роботи потрібні знання компетентних фахівців з предметної області, без наповненої бази алгоритм не буде мати жодних інструкцій до формування рекомендацій. Також для деяких предметних

областей необхідність до прямого опитування клієнта може бути проблемою, вирішення котрої є окремою задачею.

Системи на основі знань використовуються при особливій складності предметної області та наявності значної кількості комплексних аспектів для аналізу – для формування відповідних рекомендацій потрібно мати велику кількість стартових вимог та врахувати великий перелік тенденцій. Тому такі системи є найбільш ефективними при роботі з предметними областями, специфіка котрих передбачає вкрай високі ціни на товари та низьку частоту придбання, іншими словами в обставинах, коли операція покупки товару є зваженим та вкрай важливим для клієнта рішенням. Рекомендації в таких випадках мають особливо високі вимоги до точності, котрим knowledge-based системи в значній мірі відповідають. Ефект від стійкості до холодного старту максимізується в середовищі, де майже усі користувачі знаходяться у відповідних умовах, при цьому процес уточнення умов клієнтом, котрий в цілому негативно впливає на користувацький досвід, є типовим та обов'язковим для предметної області.

1.3 Підходи до реалізації рекомендацій

Описані вище підходи відносяться до принципу, за яким інформація використовується для формування рекомендацій. Проте кожен з них може бути реалізований у більш ніж один спосіб. В залежності від алгоритму реалізації рекомендаційні системи також можна поділити на дві категорії:

a) На основі пам'яті;
b) На основі моделі.

Підхід на основі пам'яті передбачає використовування під час роботи чіткого алгоритму, що залежить від обраного принципу побудови рекомендацій та оснований на аналізі взаємодій між клієнтом та товаром. Таким чином під час довільного сеансу роботи системи, при формуванні рекомендацій для певного користувача система в реальному часі аналізує необхідний масив

інформації з використанням однієї чи більше метрик схожості за певною чітко визначеною послідовністю дій та математичних перетворень [7]. З одного боку, з такого підходу до реалізації рекомендатора випливає ряд переваг. Перш за все, точність таких рекомендацій в більшості є на високому рівні, задача системи – обчислити точні значення з використанням усіх наявних в системі на цей момент даних, що має однозначно позитивний вплив на якість прогнозу. Використання повного об'єму інформації також добре впливає на одну зі сторін масштабованості – при доданні нового товару, в порівнянні з підходом на основі моделі, не потрібно виконувати жодних додаткових дій.

Проте підхід на основі пам'яті має ряд недоліків. По-перше, необхідність обробки великих масивів даних призводить до непомірно великих часових витрат при кожному прогнозі зі зростанням системи та, відповідно, загальних обсягів корисних даних. Окрім проблеми масштабованості підходи на основі пам'яті мають також потребу в даних про об'єкт котрий піддається аналізу, тобто щоб клієнт або товар міг бути повноцінно інтегрований в рекомендаційне середовище, система має мати достатню кількість відповідних даних типу «клієнт-товар», що залишає проблему холодного старту.

У свою чергу, підходи на основі моделі пропонують вирішення вищеописаних проблем. Такі системи ґрунтуються на глибинному навчанні та задіють можливості нейронних мереж, такі як стійкість до великих обсягів даних та можливість вирішення комплексних проблем, та поєднують їх з класичними алгоритмами, завдяки чому система може ідентифікувати в значній мірі комплексні взаємозв'язки у взаємодії користувача з товаром [8], при цьому значно зменшується складність системи [9]. Описані властивості даних алгоритмів у результаті дають змогу досягнути відносно високих показників точності, дозволяють знизити вплив холодного старту на ефективність рекомендаційної мережі та підвищують масштабованість системи.

1.4 Аналіз проблеми проєктування гібридних рекомендацій

Кожен з описаних методів має свої особливості реалізації, власні недоліки та переваги при задіянні у різних умовах. У свою чергу довільні системи, що поєднують більш ніж один алгоритм, відносять до класу гібридних. Використання комбінації підходів передбачає обробку інформації одразу з декількох джерел[10], тоді як гнучкість гібридної моделі дозволяє керувати впливом кожного компоненту, нівелювати їх слабкі місця, більш того, сприяючи доповненню їх взаємних переваг [11]. Задача проєктування гібридної системи полягає в обґрунтованому та ефективному поєднанні методів та практик в єдину модель з метою досягнення підвищення її загальних характеристик, досягнувши при цьому сприйнятливих показників оптимізації. У порівнянні з іншими простішими методами, її ефективність значно вище, та залежить від обраної в процесі проєктування архітектури. При створенні моделі потрібно приймати такі проєктувальні рішення, при котрих гібридна система буде максимально задовольняти поставленим вимогам та цілям. При цьому під час проєктування гібридної системи передбачається розв’язання ряду проблем, котрі описані нижче.

Першим питанням, котре вирішується, є вибір алгоритмів. Необхідно визначити, які алгоритми слід комбінувати та як їх найкраще інтегрувати для досягнення оптимальних результатів. В контексті даної задачі не існує універсального рішення або методів, оскільки кожен набір даних та кожен контекст вимагають індивідуального підходу, а спрогнозувати ефективність роботи окремих алгоритмів як єдиної взаємопов’язаної системи неможливо без проведення відповідних експериментів. Окрім цього, при збільшенні кількості користувачів та елементів в базі, система може зіткнутися з проблемами масштабованості, зокрема з підвищенням часу відгуку та зниженням продуктивності генерації рекомендацій. Тому під час визначення математичної моделі прийнято звертати увагу ще й на складність того чи іншого компоненту схеми, на ступінь складності обчислень порівняльно до результату, що система

отримує від застосування, та на специфіку предметної області з точки зору швидкості зростання тих чи інших типів даних.

Коли визначена архітектура системи, та відомий перелік алгоритмів, що використовуються в ній, окрім вже зазначених питань треба з'ясувати, у якій формі будуть зберігатися дані та як вони будуть оновлюватися. Оскільки гібридна система використовує великий перелік різних за характером даних, дане питання потребує особливої уваги. Також на початкових етапах проєктування варто урахувати принцип, за яким система буде збирати дані про потреби користувачів, та в якому вигляді буде представлений зворотній зв'язок.

Після вирішення усіх задач пов'язаних з моделлю постає питання реалізації окремих її елементів. На даному етапі одним з найбільш пріоритетних факторів залишається оптимізація, та враховуючи той факт, що більшість підходів використовує принцип взаємодії користувача та товару, неминучим є питання роботи з великою кількістю розріджених даних. Для даної проблеми існує великий перелік методів та, відповідно, рішень, що їх реалізують. У свою чергу, головним критерієм при виборі методу оптимізації розріджених даних є приріст швидкодії.

В області проблем практичного характеру також виникає потреба в адаптації рекомендацій до індивідуальних потреб користувачів та контексту, в якому вони перебувають, при цьому зберігаючи певний рівень випадковості в товарах. Врахування часу доби, різного роду даних про користувача та інших контекстних факторів може значно покращити якість рекомендацій [12]. У той ж час ключовим аспектом залишається забезпечення різноманітності в рекомендаціях та сприяння відкриттю нового контенту, який може бути цікавим користувачам, але виходить за рамки їх звичайних переваг. Гібридні системи можуть балансувати між точністю та різноманітністю, але знаходження правильного співвідношення є в значній мірі комплексним питанням, що потребує проведення відповідних досліджень.

Також одним з найпріоритетніших аспектів проєктування рекомендаційної системи є холодний старт. Найбільшою перевагою гібридних

моделей та головною причиною їх використання на комерційних підприємствах є її стійкість до даних умов, та значна частина приросту точності обумовлена саме цією властивістю [13]. Тому під час проєктування схеми рекомендатору вкрай важливим є орієнтація на мінімізацію впливу при різних типах холодного старту.

Для оцінки ефективності системи в процесі створення та порівняння архітектурних рішень між собою варто мати відповідні метрики, за котрими можна було б чітко визначити, наскільки точним є отриманий результат. Для рекомендаційних систем метрики залежать від задачі чи переліку задач, що вирішуються. Можна виділити наступний перелік таких задач:

– Задача прогнозування рейтингу – система прогнозує певний числовий показник – оцінка або рейтинг, котрий описує потенційний ступінь вподобання клієнтом товару;

– Задача класифікації – прогнозування того, чи сподобається клієнту товар, чи не сподобається, або чи купить він його, тощо. Іншими словами кінцевий варіант представляє прецедент у вигляді класу, що належить до відповідного бінарного чи множинного простору;

– Задача ранжування – передбачає впорядкування списку рекомендацій в порядку релевантності. Завдання ранжирування особливо важливе у ситуаціях, коли необхідно спрогнозувати обмежену кількість позицій з чисельного за змістом каталогу.

В залежності від того, які задачі обрані для вирішення, використовуються відповідні метрики. Наприклад, для оцінки задачі класифікації можна використати точність (Precision) чи повноту (Recall), для задачі прогнозування рейтингу – середню абсолютну похибку (MAE), середньоквадратичну помилку (RMSE) чи середню абсолютну відсоткову помилку (MAPE). Задача ранжування у свою чергу може бути оцінена за такими метриками як, наприклад, нормалізований знижений накопичувальний виграш (NDCG). При оцінюванні гібридної рекомендаційної системи часто вирішуються одразу декілька задач, наприклад прогнозування оцінки разом з позицією к кошику чи

показником регулярності придбання. Тому проєктування такої системи вимагає зваженого вибору значної кількості метрик для аналізу системи, що розробляється.

1.5 Постановка задачі

Мета дослідження – проєктування рекомендаційної системи, що має найбільш оптимальні показники ефективності, а саме:

- має вищу точність в порівнянні зі базовими методами;
- має прийнятну швидкість – приріст часу роботи має бути не більше за встановлене значення;
- ефективно вирішує проблему розрідженості даних;
- має високий рівень точності в умовах холодного старту.

Задля досягнення поставленої мети необхідно виконати наступні задачі:

- дослідити існуючі архітектури рекомендаційних систем;
- дослідити підходи до вирішення розрідженості даних
- на основі результатів дослідження побудувати декілька варіантів архітектури системи;
- реалізувати та порівняти системи за показниками точності та швидкодії в різних умовах.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ

2.1 Визначення умов дослідження

У ході дослідження методів побудови рекомендаційних систем варто визначити певні умови, за якими будуть проводитися дослідження. По-перше треба зазначити, що само рекомендаційні системи будуть аналізувати. Оскільки робота з рекомендаційними системи є в значній мірі вимогливою до даних задачею, проведення досліджень потребує великих обсягів комерційних даних. У цей ж час сама предметна область продажу товарів передбачає порівняльно великі обсяги даних, тому вимоги до величини датасету також зростають, а з урахуванням наявності значного рівня конфіденційності в даній сфері, доступ до такої інформації є в значній мірі обмеженим. Тому за відсутності можливості аналізу неявних показників взаємодії користувачів з товарами, таких як час перегляду сторінок, придбання тих чи інших позицій, у дослідженні буде проводитися аналіз показників явного зворотного зв’язку – оцінок товарів. Дана інформація є більш доступною, при цьому не набагато менш показовою.

По-друге слід зазначити, які само показники роботи рекомендаційних систем будуть фіксуватися. У рамках даної роботи прийнято рішення досліджувати ефективність системи, яка традиційно складається з наступних показників системи:

a) Точність або ефективність;
b) Швидкодія.

Для визначення швидкодії доцільно використати середній час, котрий система витрачає на генерацію рекомендацій клієнту. Для цього фіксується повна довжина проміжку часу від початку роботи рекомендатора до завершення роботи на надання результатів прогнозу за усією тестовою вибіркою даних у кінцевому вигляді, прийнятному для подальшої роботи. Після

цього обчислюється частка від ділення отриманого значення на загальну кількість клієнтів, що фігурують у вибірці.

$$T = \frac{t_{\text{кін}} - t_{\text{поч}}}{n}, \tag{2.1}$$

де T – середня кількість часу, що система витрачає на рекомендації клієнта;

$t_{\text{поч}}$ – час початку роботи рекомендаційного алгоритму;

$t_{\text{кін}}$ – час отримання результатів роботи рекомендаційного алгоритму;

n – загальна кількість клієнтів, що фігурують у вибірці.

У свою чергу, для оцінки ефективності систем передбачається використання двох наступних метрик – точності (precision) та повноти (recall). З точності можна зробити висновок щодо здатності моделі ідентифікувати релевантні дані, тоді як повнота демонструє яку частину з загальної кількості релевантних позицій змогла ідентифікувати система. Аналіз та зіставлення цих двох показників може дати уявлення про ефективність моделі та її недоліки. Точність та повнота визначаються формулами 2.2 та 2.3 відповідно [14].

$$Precision = \frac{TP}{TP + FP}, \tag{2.2}$$

$$Recall = \frac{TP}{TP + FN}, \tag{2.3}$$

де TP – кількість істино позитивних результатів;

FP – кількість хибно позитивних результатів;

FN – кількість хибно негативних результатів.

Проте вищеописані метрики використовуються для оцінки бінарної класифікації, коли прогноз у предметній області продажу товарів, ведеться в умовах мультикласової класифікації. Кожен клас в даній задачі являє собою числовий показник, тобто оцінку, котру клієнт може надати товару. Оскільки

оцінки традиційно надаються користувачами в межах шкали від одного до п’яти, у рамках задачі, наприклад, оцінити точність роботи рекомендатора, необхідно оцінити його роботу розпізнавати надання клієнтом усіх п’яти оцінок, тобто розпізнавати усі п’ять класів.

Оскільки таким чином ми отримаємо п’ять окремих показників точності, потрібно об’єднати результати для того, щоб оцінити загальний показник точності РС. З метою об’єднувати отримані значення варто використати схему макро-усереднення, котре дає змогу отримати середнє значення для множини оцінок, обчислених для класифікатора в контексті певної метрики. Вибір саме макро-усереднення обумовлений тим, що класи оцінок за мірою представленості в більшості дорівнюють один одному у розмірі. Принцип макро-усереднення справедливий для усіх метрик, як для точності, так й для повноти, та описаний у формулі 2.4. У даній формулі у якості *B* може виступати як точність, так і повнота, в залежності від того, до якої метрики потрібно застосувати принцип макро-усереднення.

$$B_{\text{макро}} = \frac{1}{n}\sum_{i=1}^{n} B_i, \tag{2.4}$$

де $B_{\text{макро}}$ – макро-усереднення для певної метрики – точності або повноти;

B_i – значення метрики для класу і;

n – загальна кількість класів.

Оскільки крім систем на основі пам’яті під час дослідження передбачається задіяння систем на основі моделі, варто визначити додаткові умови, пов’язані з їх специфікою. Таким чином, для запобігання проблеми «витоку даних» при формуванні тренувальної та валідаційної вибірок було прийнято рішення щодо використання підходу до поділу на основі послідовності у часі. Вищезазначена проблема являє собою використання в процесі навчання даних, що не передбачаються для використання на даному етапі. Нехтування часовою динамікою може мати такі наслідки, як неправильна

інтерпретація даних під час навчання через упущення глобального часового контексту [15], переоцінка продуктивності моделі внаслідок неявного використання вже при тренуванні інформації з більш пізнього часового періоду, що в певній мірі знижує коректність умов при тестуванні, штучно підвищуючи показники ефективності системи.

З цієї причини, враховуючи специфіку дослідження, використання випадкового поділу за визначеним коефіцієнтом не є доцільним. Тому була запропонована вже зазначена альтернативна схема за принципом послідовності у часі. Такий підхід виключає випадковий перетин у часі в контексті довільного користувача двох вибірок, використовуючи для виділення тестового набору даних принцип «n-останніх», де n – довільно обрана, з урахуванням загальної, кількість записів, що відносяться до певного клієнта. Таким чином загальний набір даних підлягає валідації, після чого верифіковані записи підлягають занесенню в тестувальну вибірку. Такий метод є потенційно більш ефективним при навчанні та більш показовим при оцінці ефективності моделі.

2.2 Вимоги до систем

2.2.1 Визначення часткового інтерфейсу

Після визначення умов дослідження, варто також визначити вимоги й до самих систем. По-перше, перед проєктуванням схем систем та проведення самих досліджень треба прийняти певні рішення щодо питань, котрі до цього етапу залишалися відкритими. Таким чином, якщо усі системи будуть використовувати спільні рішення та відповідати спільним вимогам, то їх порівняльна оцінка буде більш показовою.

Перш за все, потрібно визначити функцію, котру мають виконувати усі системи, та відповідно частковий інтерфейс. Наявність частково загального інтерфейсу дозволяє проводити більш показове порівняння різних за принципом рекомендаційних систем. Це допомагає виявити сильні та слабкі

сторони кожної системи, та робить умови прогнозування більш придатними для задачі порівняльного аналізу, бо якщо одна система ітераційно надає інформацію оцінки клієнтом товару, а інша ітеративно прогнозує десять товарів, котрі він скоріш за все купить, то порівняння таких систем стає складною задачею, при тому, що цінність таких порівнянь в значній мірі знижується.

Таким чином, функцією систем має бути прогнозування оцінки клієнтом одного товару. Іншими словами, вхід кожної системи має містити область, що приймає дані товару, в залежності від того, які з них потребує конкретна система, а на виході – певний числовий показник, що відображає ступінь, у котрій конкретний товар відповідає вподобанням клієнта. При цьому, в залежності від принципу системи та її особливостей, вхід окрім зазначених даних приймає також в тій або іншій формі характеристику клієнта та інші специфічні для реалізації дані. Так само й вихід – система, окрім оцінки може надавати, наприклад, ймовірність присутності товару у якості першої позиції у кошику, проте однозначно має надавати вищеописаний показник, що корелює з вподобаннями клієнта відносно одного конкретного товару, а не ряду товарів, тощо. Вищеописані вимоги варто брати до уваги при проєктуванні схем кожної з систем, що досліджуються. Виключенням можуть стати лише системи, базові принципи котрих не дозволяють накласти такі обмеження та дотримуватися відповідних умов.

2.2.2 Забезпечення обробки послідовностей товарів

Для рекомендаторів, особливо на основі моделей, існує багато проблем, пов’язаних з проєктуванням конкретної системи, котрі потребують вибір та реалізацію того чи іншого алгоритму у якості компоненти системи. Беручи до уваги велику кількість систем, багато з вищеописаних проблем є спільними для деяких з систем, що досліджуються. Тому має сенс заздалегідь визначити усі з цих проблем, як окремі задачі, та вирішити їх, після чого використати для усіх

систем, що не лише спрощує проєктування, а й за аналогією з інтерфейсом надає більше об’єктивності дослідженням, бо системи крім ключових для підходу компонент, використовують спільні рішення.

Говорячи про колаборативно-специфічні задачі, перш за все виділяється задача надання системі послідовності товарів, що були оцінені клієнтом. Треба визначити спосіб у який ці дані надаються системі. Якщо для колаборативної фільтрації на основі пам’яті ця задача не викликає додаткових питань через те, що реалізація являє собою або довільний алгоритм, або операції над матрицями, то моделі на основі пам’яті мають специфіку детермінованих розмірностей даних при тому, що на вхід вони мають бути надані в нормалізованому вигляді.

Таким чином, для цієї задачі було запропоновано використати наступний підхід. Нехай існує послідовність товарів, при цьому кожен елемент послідовності містить інформацію про товар, обов’язково ідентифікатор товару, та оцінку котру клієнт надав товару. Оскільки кінцева довжина має мати чітко визначене значення, існує необхідність з певних міркувань обмежити потенційно нескінченну послідовність товарів фіксованим числом. У цьому випадку доцільно сформувати вибірку з N останніх товарів, котрі придбав клієнт. Експериментальним шляхом встановлено, що максимально оптимальним числом для N буде п’ятнадцять, усі ж позиції котрі не зайняті, наприклад, якщо клієнт придбав менше п’ятнадцяти товарів, усі відповідні позиції заповнюються токенами, котрі сигналізують про відсутність товару в послідовності, потрібні для того, щоб скорегувати розмірність під систему, котра приймає рівно п’ятнадцять позицій. При обробці послідовності ці токени перетворюються у відповідне числове представлення, що відповідає пустому місцю в послідовності. Обробку ж послідовності можна реалізувати за допомогою ембедінгу – алгоритму представлення об'єктів у вигляді векторів у тому чи іншому просторі ознак [16]. Кожен елемент, тобто кожен товар можна розкласти на дані про товар, оцінку, котра надана товару клієнтом, та позицію товару в послідовності. Після цього кожен з цих можна пропустити через шар

ембедингу, після чого з отриманими представленнями у вигляді ембедингу даних товару, ембедингу оцінки та позиційного кодування, можна робити подальші операції. Експериментальним шляхом визначено, що найбільш ефективним є складання отриманих послідовностей, у порівнянні з, наприклад, добутком чи конкатенацією. Таким чином схема отримання послідовності товарів представлена формулою 2.5.

$$GD = E\left(data_{\text{товар}}\right) + E(rate) + E(pos), \tag{2.5}$$

де GD – числовий показник закодованих даних одного з останніх товарів клієнта;

E – функція ембедінгу;

$data_{\text{товар}}$ – вектор даних товару;

$rate$ – вектор оцінок, що клієнт надав товару;

pos – позиційний ідентифікатор товару в послідовності.

Після кодування даних отримане представлення подається на вхідний шар рекомендаційної системи разом з іншими вхідними даними.

2.2.3 Забезпечення обробки характеристик товарів

Предметна область продажу товарів передбачає наявність у різних товарів кардинально різного набору характеристик, кількість котрих у товару є довільною. Оскільки типів товарів можуть бути десятки тисяч, кількість відповідних характеристик досягає мільйонів. З цієї причини при передачі системі характеристик товару виникає проблема, сутність котрої полягає в тому, що при вхід такої системи має передбачати усі можливі характеристики, котрі може мати товар, тобто вхід повинен мати дуже великий розмір, що може бути вкрай незручним та складним при реалізації. Дана проблема ускладнюється за умови наявності категоріальних даних – оскільки традиційний спосіб обробки даного типу даних у машинному навчанні, one-hot

encoding, передбачає розкладення атрибуту в окремі стовбці, кількість котрих досягає кількості категорій в домені, кожен такий атрибут значно підвищує ширину вхідного вектору. Вищеописана проблема в повній мірі відповідає предметній області продажу товарів – багато товарів мають категоріальні дані, наприклад, матеріали для корпусу годинників, тип механізму в велосипедах, тощо.

З цієї причини метод, що був запропонований до кодування послідовностей товарів, можна з деякими змінами застосувати й для вирішення даної задачі. По-перше, кількість атрибутів для обробки у такій кодувальній підсистемі буде також мати кінцевий розмір. Оскільки статистично кількість атрибутів у товару вище ніж кількість придбаних товарів клієнта, має сенс встановити розмір вектору приблизно двадцять п’ять. Кожен атрибут товару має власний ідентифікатор атрибуту, та відповідно значення, котре атрибут приймає для даного товару. Ці значення перетворюються за допомогою відповідних шарів ембедінгу, після чого, за аналогією з послідовністю товарів, визначається сума векторів.

$$AD = E(attr) + E(val_{attr}), \tag{2.6}$$

де AD – вектор, що представляє закодовані дані атрибутів товару;

E – функція ембедінгу;

$attr$ – вектор ідентифікаторів атрибутів товару;

val_{attr} – вектор значень атрибутів товару;

Результуючий вектор розмірності 25 буде в повній мірі репрезентувати набір атрибутів товару у формі, що придатна для використання в машинному навчанні.

2.2.4 Обробка матриць

Деякі з алгоритмів, котрі будуть досліджуватися, у схемі потребують роботи з матрицями. Ця задача, враховуючи предметну область електронної комерції, ускладняється вкрай великими розмірностями даних, при наявності вкрай високого рівня розрідженості. Традиційно для рекомендаційних систем для рішення цих питань в схемі рекомендатору використовується компонент матричної факторизації.

Алгоритми з класу матричної факторизації працюють шляхом розкладання матриці на добуток двох матриць, що мають нижчу розмірність [17]. Позначимо матрицю взаємодії користувачів з товарами як $A \in R^{m \times n}$, де m – кількість клієнтів, n – кількість товарів. Тоді загальний вигляд розкладу буде наступним: $UV^T = A$, при тому, що матриця ембедінгу користувача є $U \in R^{m \times k}$ та матриця ембедінгу товару є $V \in R^{n \times k}$, а k – кількість компонент. Матриці U та V містять компактне представлення всіх характеристик клієнтів та товарів відповідно, тому модель матричної факторизації в процесі навчання намагається знайти оптимальний вигляд матриць.

Для використання в складі нейромережевої моделі алгоритм матричної факторизації доцільно реалізувати як GMF (Generalized Matrix Factorization) [18]. Даний алгоритм є нейромережевим підходом до даної задачі, що поєднує класичну матричну факторизацію з додатковими функціями та нелінійними перетвореннями для врахування більш складних взаємодій між користувачами та предметами. Такий підхід дозволяє ефективно інтегрувати алгоритм в нейромережеве середовище, що у свою чергу дозволить досягти більшої продуктивності, тоді як використання нелінійних функцій може дати можливість ідентифікувати більш складні взаємозв’язки та таким чином дати змогу підвищити рівень точності та адаптивності. Одночасно з цим матрична факторизація є також частковим вирішенням проблеми розрідженості даних, бо в самій рекомендаційній системі обробка даних виконується з представленням,

елементи котрого у порівнянні з початковою матрицею мають значно менші розмірності.

2.3 Проєктування окремих систем рекомендації товарів

2.3.1 Визначення переліку архітектур рекомендаційних систем

Перед безпосередньо проєктуванням слід визначити перелік архітектур, що будуть порівнюватися, щоб таким чином повністю репрезентувати кожну з груп та підходів до генерації рекомендацій. При виконанні цієї умови результати досліджень могли би бути максимально показовими.

По-перше, для обраних загальних підходів, а саме колаборативна фільтрація та системи на основі контенту, необхідно реалізувати системи на основі пам’яті. Таким чином, можна буде порівняти системи на основі моделі з їх більш традиційним аналогом, та зробити вже на основі практичних результатів висновки щодо переваг та недоліків кожного з цих варіантів реалізації. Вибірка ж архітектур на основі моделі у свою чергу має найбільш повно відображати варіативність підходів до систем у даному класі.

Таким чином обраний наступний перелік моделей:

– CF на основі пам’яті – традиційний підхід до реалізації колаборативної фільтрації;

– CB на основі пам’яті – традиційний підхід до реалізації фільтрації на основі контенту;

– Wide and Deep – поєднання традиційного та нейромережевого підходів;

– DeepFM – нейромережева архітектура на основі матричної факторизації;

– NeuMF – нейромережева реалізація колаборативної фільтрації;

– Deep and Cross – архітектура, що фокусується на взаємодії між ознаками;

– AutoRec – реалізація колаборативної фільтрації на основі автоенкодерів.

2.3.2 Проєктування моделі CF на основі пам’яті

Класична реалізація рекомендаційної системи на основі принципу колаборативної фільтрації передбачає побудову алгоритму, що за рахунок проведення певних математичних обчислень дозволяє вивести клієнтську оцінку для певного товару. Для даної архітектури ключовими етапами буде побудова матриці взаємодії, обчислення подібності між клієнтами, та прогнозування оцінок.

Матриця взаємодії є основним компонентом колаборативного підходу, котрий є необхідним для проведення усіх подальших перетворень. Вона являє собою двовимірну матрицю, де вертикальна вісь є представленням клієнтів, а горизонтальна – товарів, тобто окремо узятий рядок є представленням усіх показників взаємодії, що користувач має відносно кожного товару, а окремий стовпець дає представлення про результати взаємодії з усіма клієнтами, що можна побачити на рисунку 2.1. Таким чином, значення в комірках такої матриці відображають взаємодію між користувачами та елементами, котра у даному випадку, як вже було зазначено, являє собою оцінки, котрі користувач надає товарам.

У результаті алгоритм даної реалізації рекомендаційної системи наведений на рисунку 2.2.

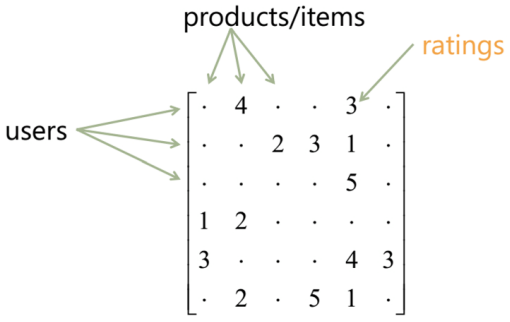


Рисунок 2.1 – Схема матриці взаємодії

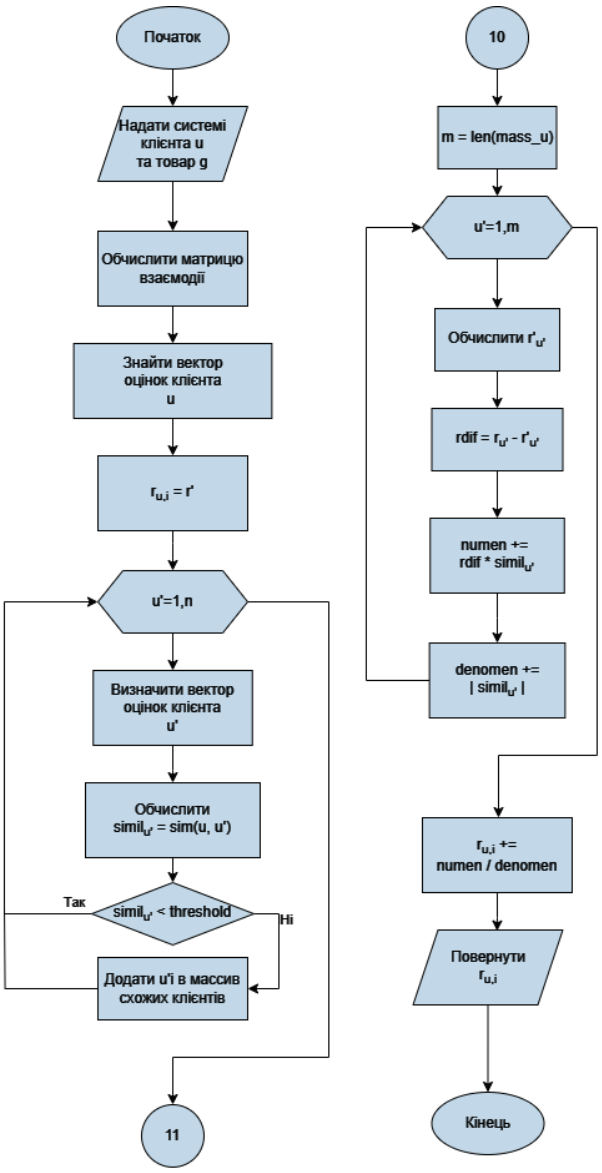


Рисунок 2.2 – Схема алгоритму CF на основі пам’яті

Під час етапу визначення подібності між клієнтами визначаються певні коефіцієнти, що відображають дану міру подібності, для чого треба обрати підходящу метрику. З урахуванням специфіки, тобто великої розмірності матриці, у якості функції подібності була обрана косинусна відстань. Дана міра є порівняльно простою в обчисленні, при тому в повній мірі відображає схожість вподобань клієнтів. Таким чином, для обчислення подібності між цільовим клієнтом A та довільним клієнтом B, визначаються відповідно два вектори розмірністю N, де N – непостійна величина, що являє собою загальну кількість товарів в системі на момент генерації рекомендацій. Таким чином задача визначення міри подібності між користувачами зводиться до визначення

косинусної відстані між векторами в n-мірному просторі, та приймає наступний вигляд [19]:

$$sim(A,B) = \frac{\sum_{i=1}^{n}(A_i * B_i)}{\sum_{i=1}^{n}(A_i)^2 + \sum_{i=1}^{n}(B_i)^2}, \quad (2.7)$$

де $sim(A,B)$ – показник подібності між користувачем А та В;

n – кількість товарів в системі на момент обчислень;

A_i – оцінка, надана клієнтом А товару і;

B_i – оцінка, надана клієнтом В товару і.

На останньому етапі за отриманими показниками виділяється певна кількість найбільш подібних клієнтів, котра для даного дослідження була встановлена в десять осіб. Тоді прогнозування оцінки клієнта певному товару має виглядає наступним чином:

$$r_{u,i} = \bar{r} + \frac{\sum_{u' \epsilon U}^{n} sim(u,u')(r_{u',i} - \bar{r}_{u'})}{\sum_{u' \epsilon U}^{n} |sim(u,u')|}, \quad (2.8)$$

де $r_{u,i}$ – прогнозована оцінка клієнта u товару і;

$\bar{r}$ – середня оцінка клієнта за товарами;

u' – користувач з множини U, що складається з n найбільш схожих на u;

$sim(u,u')$ – функція подібності двох клієнтів;

$r_{u',i}$ – оцінка, що надана подібним клієнтом u' товару і;

$\bar{r}_{u'}$ – середня оцінка подібного користувача.

Дана схема обчислення дозволяє під час прогнозування врахувати загальний рівень схожості клієнта відносно більшості клієнтів у базі даних, а також тенденцію оцінок клієнта, щоб прогнозування найбільш персоніфікованої оцінки.

2.3.3 Проєктування моделі СВ на основі пам’яті

Класична реалізація рекомендаційної системи на основі контенту передбачає досить простий алгоритм, котрий полягає в порівнянні та зіставленні двох списків. При цьому першим списком є перелік значень характеристик товару, для котрого прогнозується оцінка, а другий список являє собою профіль клієнта. Як вже було зазначено, профіль клієнта – це вектор, де кожна позиція являє собою ваговий коефіцієнт, що характеризує ступінь зацікавленості клієнта в тому чи іншому атрибуті товару. Таким чином, якщо при покупці телевізору для клієнта є вкрай важливою ціна та роздільна здатність екрану, то при прогнозі дані характеристики моделі будуть мати значний вплив на кінцеву оцінку.

Варто зазначити, що форма, у котрій має бути представлений профіль, це саме вектор, а не масив, бо при урахуванні кількості характеристик, робота з розрідженими структурами не є доцільним. При цьому, базовий підхід до порівняння переліку атрибутів товару з моделлю вподобання потребує деяких додаткових кроків зважаючи на особливості предметної області. Оскільки не має жодних обмежень на домен товару, окремі позиції в базі даних можуть мати кардинально різний перелік характеристик. З чого можна зробити наступний висновок: у профілі користувача перелік характеристик того чи іншого типу товару, котрий порівнюється, буде займати незначний відсоток від загального його розміру. Тому має сенс відокремлювати від профілю лише ту частку, котра в повній мірі описує перелік атрибутів домену, до котрого включений товар, що підлягає прогнозу.

Усі вищезазначені моменти передбачають додаткову умову використання в якості кожного елементу вектору пари «ідентифікатор атрибуту – значення атрибуту», через те що позиція атрибуту у векторі не відображає його позицію в загальному списку атрибутів. Також варто зазначити, що атрибути можуть бути й категоріальними, що в даному випадку вирішується шляхом використання методу one-hot encoding [20]. Присутні також й «атрибути

витрат», тобто ті, котрі тим кращі для клієнта, чим їх значення менше, та оскільки підхід, що розглядається є лінійним, то цей випадок потрібно врахувати під час реалізації функції нормалізації, котра буде виглядати наступним чином:

$$inter(k, a_i) = \begin{cases} \dfrac{k - a_{i,min}}{a_{i,max} - a_{i,min}}, & a_i \to max \\ \dfrac{k - a_{i,max}}{a_{i,min} - a_{i,max}}, & a_i \to min \end{cases}, \tag{2.9}$$

де $inter(k, a_i)$ – нормалізоване значення k атрибуту a_i;

k – значення, що нормалізується;

a_i – атрибут i, до котрого відноситься значення k;

$a_{i,max}$ – максимальне значення атрибуту i;

$a_{i,min}$ – мінімальне значення атрибуту i.

Тоді при формуванні профілю клієнта, кожен окремий коефіцієнт зацікавленості атрибуту у векторі обчислюється шляхом нормалізації усереднення усіх його взаємодій с товарами, котрі відносяться до атрибуту, тобто за формулою 2.10.

$$w_a(u, i) = inter\left(\frac{\sum_{j=1}^{A} a_{u,i,j}}{A}, a_i\right), \tag{2.10}$$

де $w_a(u, i)$ – коефіцієнт зацікавленості клієнта u в атрибуті i;

u – клієнт;

i – ідентифікатор атрибуту, для котрого обчислюється коефіцієнт;

A – кількість елементів множини оцінок клієнтом u атрибуту i;

$a_{u,i,j}$ – значення j-ої оцінки клієнтом u атрибуту i.

За умови наявності в системи клієнтської моделі вподобань, можна виконати прогнозування кінцевої оцінки для товару за формулою 2.11.

$$prate(u,i) = \sum_{j=1}^{n} \left(w_{u,j} * inter(k_j, a_i) * 5 \right), \tag{2.11}$$

де $prate(u,i)$ – прогнозована оцінка клієнта u для товару i;

n – кількість атрибутів в домені, до котрого включений товар i;

$w_{u,i}$ – коефіцієнт зацікавленості клієнта u в товарі i;

k_i – значення атрибуту k_j у товарі i.

Таким чином, загальний алгоритм для прогнозування оцінки товару має схему, зображену на рисунку 2.3.

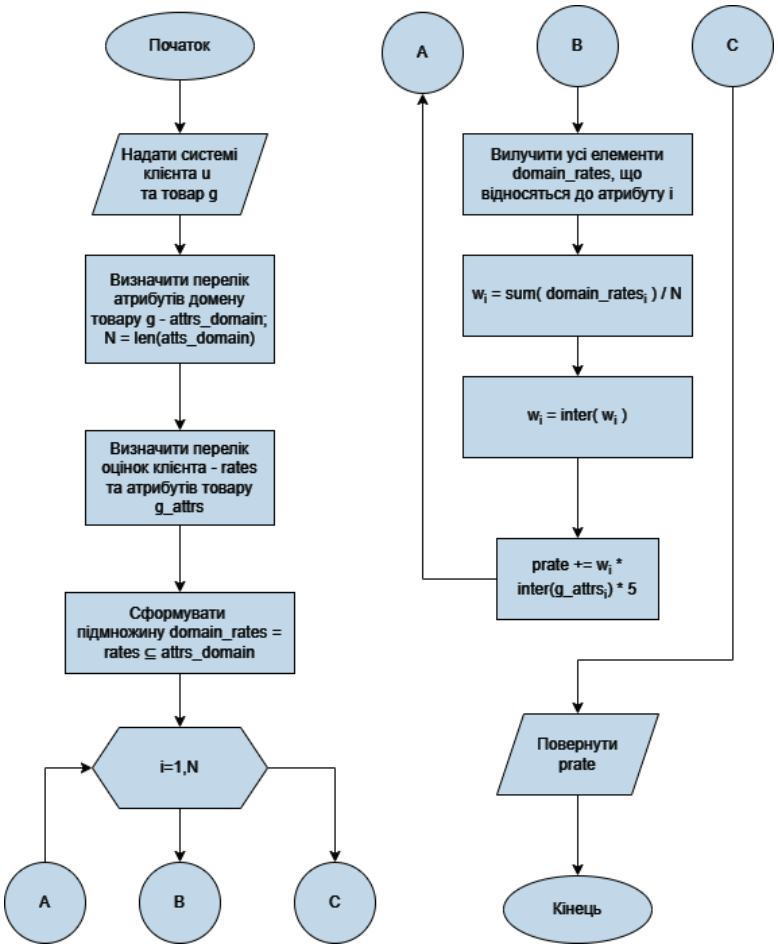


Рисунок 2.3 – Схема алгоритму СВ на основі пам’яті

2.3.4 Проєктування моделі Wide and Deep

Модель Wide and Deep є комбінованим підходом до вирішення задачі генерації рекомендацій, що поєднує в собі як рішення на основі моделі, так й на основі пам’яті. Її особливість полягає в поєднанні двох типів моделей: широкої лінійної моделі та глибокої нейронної мережі [21]. Даний варіант реалізації дозволяє одночасно використовувати переваги обох підходів: використання пам’яті та аналіз складних взаємозв’язків в великих обсягах даних.

У якості широкої частини моделі пропонується використати модуль, котрий являє собою вже визначений алгоритм контентно-орієнтованої моделі на основі пам’яті. Таким чином при проведенні досліджень ми будемо мати можливість відстежити приріст точності чи швидкодії порівняльно з базовою реалізацією рекомендатора на даному підході.

На рисунку 2.3 можна побачити визначену схему реалізації Wide and Deep. Як видні зі схеми, глибока частина моделі являє собою нейронну мережу зі входом, та трьома шарами з функцією активації – ReLU, шириною в 1024, 512 та 256 нейронів відповідно, кількість встановлено експериментальним шляхом. Оскільки типовою проблемою для функції ReLU є «вибух нейронів» [22], на кожному відповідному шарі після використання функції потрібно парно використовувати Dropout для зменшення ваг та запобігання проблеми перенавчання. На вході текстові дані товару проходять через кодування на відповідних шарах ембедингу, у цей час атрибути товару та профіль користувача як окремі послідовності проходять через схему ембедингу, визначену в підрозділі 2.2.3. Після цього усі три потоки поступають на вхідний шар конкатенації, а далі – на приховані шари мережі.

Вихід являє собою шар, що об’єднує результати обчислень глибокого та широкого компонентів системи, на даному ж етапі результат проходить фазу перетворення та нормалізується згідно прийнятій у предметній області шкали – від одного до п’яти, з наявністю дробової частини показника для можливості порівняння – необхідний аспект задачі формування рекомендацій.

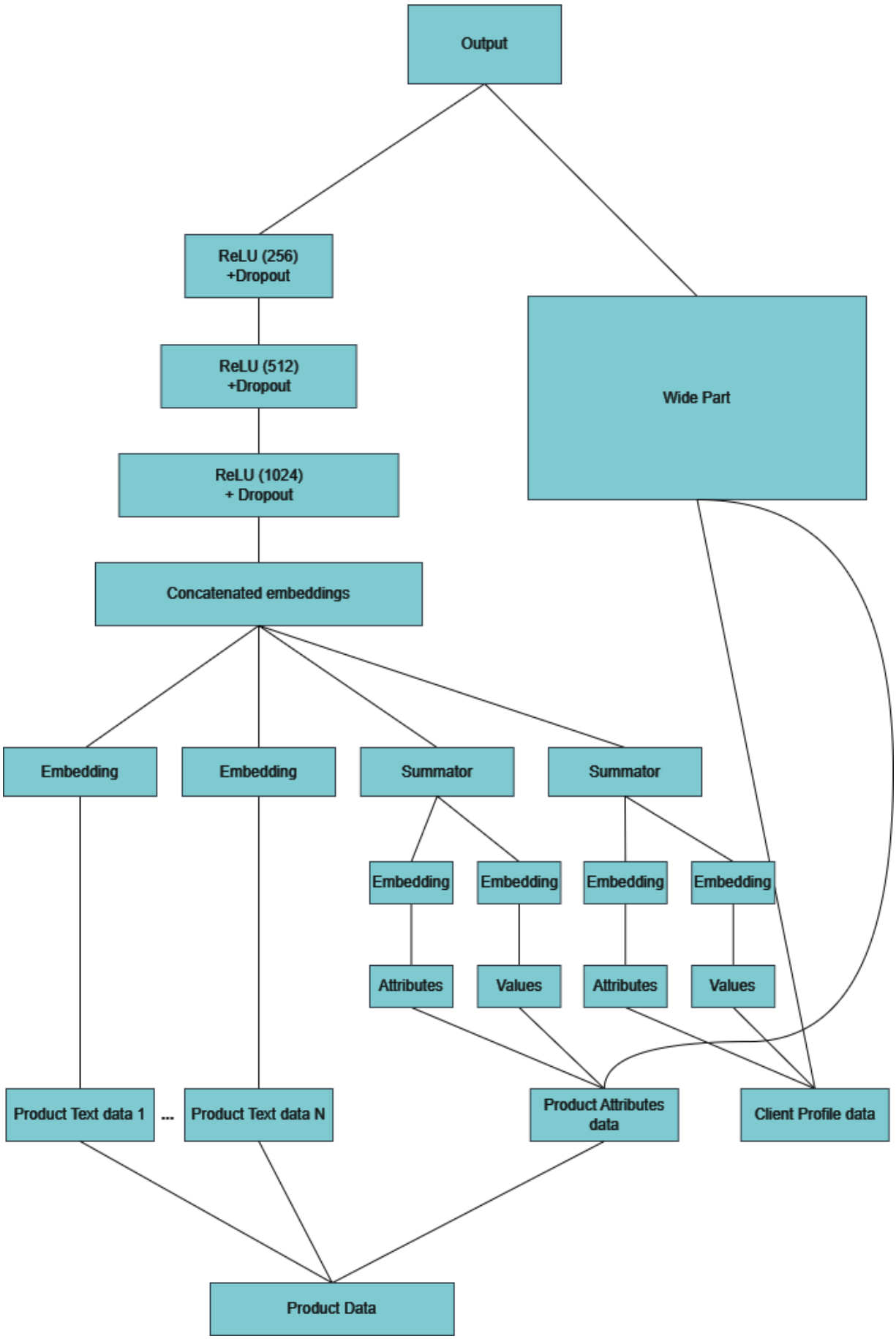


Рисунок 2.4 – Схема моделі Wide and Deep

2.3.5 Проєктування моделі DeepFM

Модель DeepFM, на відміну від багатьох інших представлених в дослідженні, являє собою, незважаючи на простоту схеми, гібридний підхід до формування рекомендацій. Модель заснована на ідеї використання машин факторизації (FM). Даний алгоритм дозволяє поєднати матричну факторизацію з лінійною регресією, та передбачає взаємозв'язки другого порядку між ознаками, що в сукупності потенційно може зробити прогнозування моделі більш ефективним [23]. Поєднання в собі можливостей машин факторизації та нейронних мереж, дозволяє моделі ефективно працювати з високорозмірними та розрідженими даними через особливість організації даних алгоритмів.

Схема системи, що буде досліджуватися, наведена на рисунку 2.5.

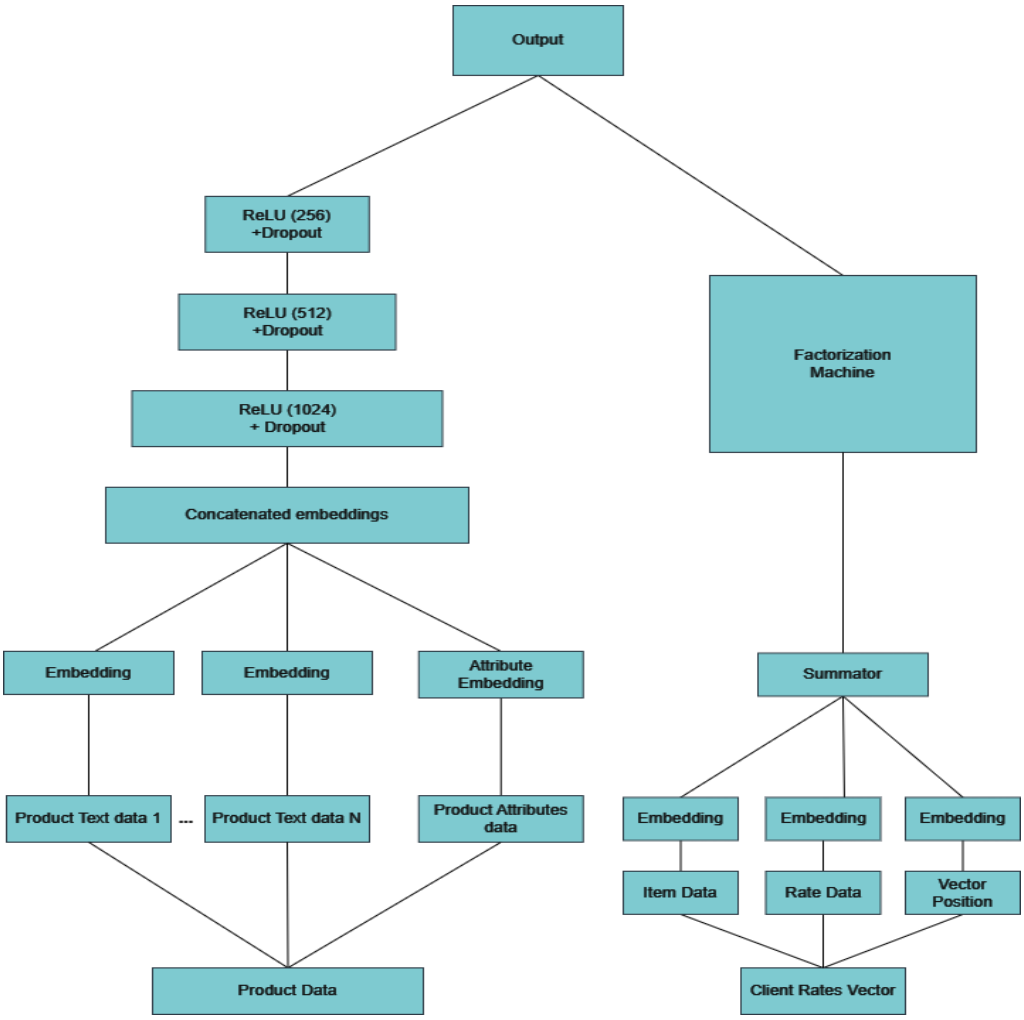


Рисунок 2.5 – Схема моделі DeepFM

Як можна побачити зі схеми, контентно-орієнтована глибока частина системи є аналогічною тій, що присутня в попередній схемі – багатошаровий перцептрон, з нелінійною функцією активації ReLU. Починаючи з даної схеми та в усіх наступних принцип кодування атрибутів, що описаний в підрозділі 2.2.3 буде представлений у вигляді загальної мережевої одиниці «Attribute Embedding». У якості ж широкої частини представлений блок, що являє собою машину факторизації для обробки колаборативних даних. Перед подачею даних клієнта на вхід блоку використовується схема ембедінгу, що описана в підрозділі 2.2.2.

2.3.6 Проєктування моделі NeuMF

NeuMF – це нейромережева модель РС, яка поєднує матричну факторизацію та методи глибокого навчання із колаборативно-орієнтованим підходом для підвищення точності рекомендацій. У певній мірі підхід, що використовує модель NeuMF, можна назвати повноцінним нейромережевим аналогом традиційної колаборативної фільтрації. На рисунку 2.6 наведена схема моделі NeuFM.

Загальна схема мережі передбачає подачу в якості вхідних даних двох векторів – вектор клієнтів та вектор товарів. Кожен елемент цих векторів являє собою перший та другий елементи довільного члену умовного вектору «клієнт-товар-оцінка», що являє собою структуровану подачу усіх взаємодій клієнту з товаром. Довжина цих векторів це є параметр моделі, від котрого залежить й довжина вихідного шару, тому даний параметр з урахуванням вимог до систем можна встановити в один. У такому випадку на вході отримуємо пару «клієнт-товар», на виході прогноз оцінки клієнта u відносно товару i.

У якості елемента матричної факторизації, як вже було зазначено, було прийнято рішення використовувати алгоритм GMF для більшої сумісності двох паралельних елементів мережі. У той же час у складі багатошарового перцептрону використовується спадаюча за шириною схема шарів лінійних

перетворень з функцією активації ReLU та регуляцією нейронів. На входах GMF та MLP, як традиційно прийнято в мережах даного типу, використовуються поелементне множення та конкатенація відповідно. Виходи двох систем подаються на шар конкатенації, після чого потрапляють на вихідний шар, що являє собою сигмоїду, масштабовану під п’ятибальну шкалу оцінювання.

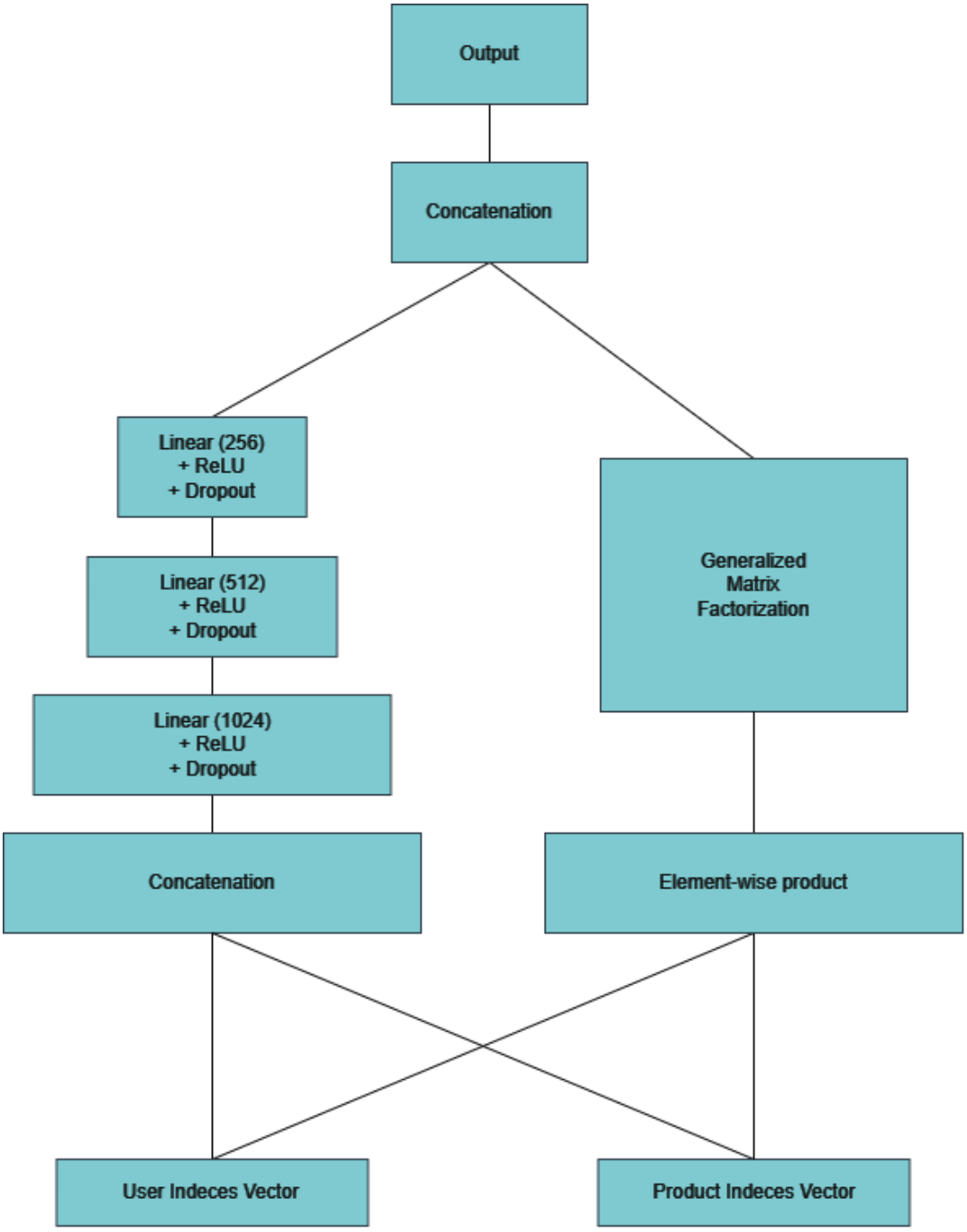


Рисунок 2.6 – Схема моделі NeuMF

2.3.7 Проєктування моделі Deep and Cross

Wide and Cross – архітектура, котра як вже було згадано, спрямована на підвищення продуктивності системи за рахунок встановлення взаємозв'язків між окремими ознаками. Моделювання взаємодій між ознаками, як вхідними, так й тими неявними, що знаходяться в прихованих шарах системи, дає нам найкращі рекомендації порівняно з використанням окремих ознак.

Ключовий компонентом для досягнення цієї задачі в системах даного типу є специфічний структура – перехресний тип шару (Cross Layer), котрий виконує фіксацію вищеописаних взаємодій за допомогою формування нових ознак на кожному наступному рівні шляхом перехресного множення вже наявних [24].

Таким чином, на рисунку 2.7 можна побачити схему моделі Deep and Cross. Оскільки ціллю перехресної частини системи є визначення лінійних взаємозв'язків, а глибокої – нелінійних, було прийнято рішення щодо передачі однієї й тої ж інформації з метою дослідження даних з різних перспектив. даної причини кількість шарів в Cross-компоненті буде аналогічною значенню цього параметру у нейромережевому компоненті, при тому що цінність встановлених взаємозв'язків зростає на кожному новому рівні.

Оскільки Cross-мережа використовує механізм перехресного множення, який не змінює розмірність ознак, а лише додає нові перехресні терміни до поточних ознак, традиційна пірамідальна схема не є застосовною до даного компоненту через потенційну можливість вкрай стрімкого росту кількості взаємозв'язків на кожному наступному рівні, котрі при цьому втрачають цінність для прогнозування. Тому усі шари в мережі мають однаковий розмір з вхідним шаром. Експериментально встановлено, що оптимальною кількістю шарів є чотири.

Глибока ж мережа виглядає аналогічно даним компонентам в попередніх моделях та заснована на використанні функції ReLU. Після прогнозування

результати обох підсистем можна ефективно поєднати за схемою «конкатенація + сигмоїда».

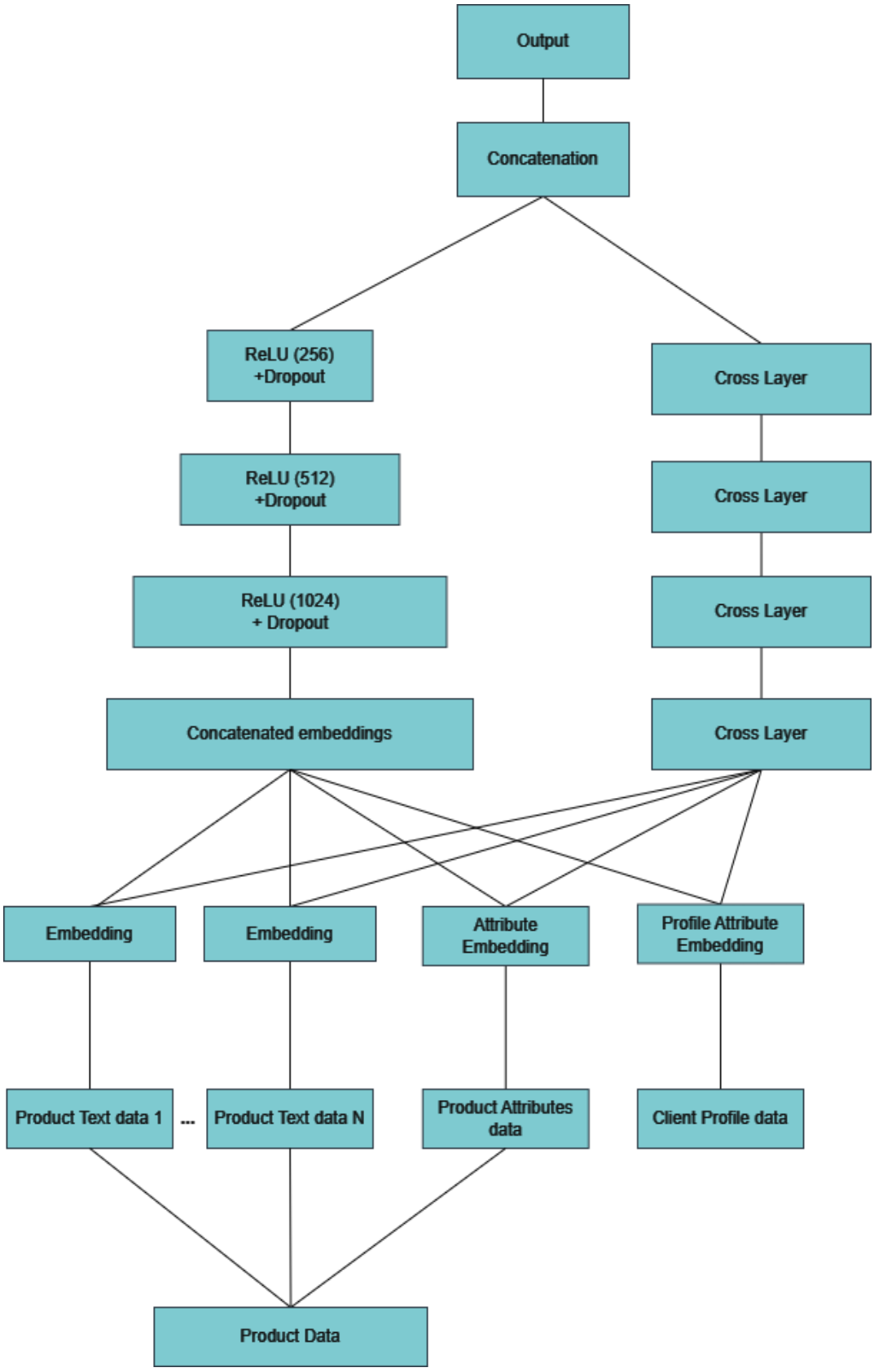


Рисунок 2.7 – Схема моделі Deep and Cross

2.3.8 Проєктування моделі AutoRec

Рекомендаційна модель AutoRec передбачає підхід, в значній мірі відмінний від інших наявних в дослідженні типів архітектур. Дана модель заснована на парадигмі колаборативної фільтрації, та в цілому вся система може представити як великорозмірний нейромережевий автоенкодер. Мета такої системи – навчитися операції зменшення та подальшого відновлення розмірності вектору вхідних даних із мінімальними втратами інформації. Оскільки мова йде про рекомендації на основі колаборативної фільтрації, то в якості такого вектору виступає вектор взаємодії клієнта з усіма товарами. Таким чином в процесі навчання на основі поведінки усіх клієнтів рекомендатор намагається заповнити інформаційні пробіли, що логічно виділяються в рамках встановлених глибинних взаємозв'язків – пропущенні або іншими словами ще неоцінені клієнтом товари [25].

На рисунку 2.8 можна побачити схему системи, що буде використана в дослідженні. На схемі можна побачити, що на вхід подається вектор взаємодії клієнта, що представлений переліком усіх оцінок, що клієнт надав кожному товару. Оскільки автоенкодер не потребує кодування даних та є відносно простою мережею, то вектор не проходять через ембедінг, більш того містить елементи для кожного товару в системі. Пропущені значення, тобто неоцінені товари, позначаються відповідним токеном. Таким чином вплив великої розмірності вектору не має критичного впливу на швидкодію системи.

Приховані шари представлені енкодером, декодером та простором латентних ознак. Енкодер представлений нелінійними шарами ReLU розмірністю, котра варіюється в залежності від кількості товарів у базі даних. Так для одного датасету розміри шарів у моделі становлять 20000 та 8000, латентний простір – 2000, а вхідні та вихідні шари – 60000. Для другого ці величини становлять для входу та виходу 1100, для першого прихованого шару – 700, для другого – 400, латентний простір складає 200 нейронів. Декодер за кількістю та розмірністю за принципом автоенкодерів аналогічний енкодеру.

Параметри моделі встановлено експериментальним шляхом. Для стабілізації ваг, як і в попередніх моделях, в парі з ReLU використано компоненти гасіння нейронів Dropout.

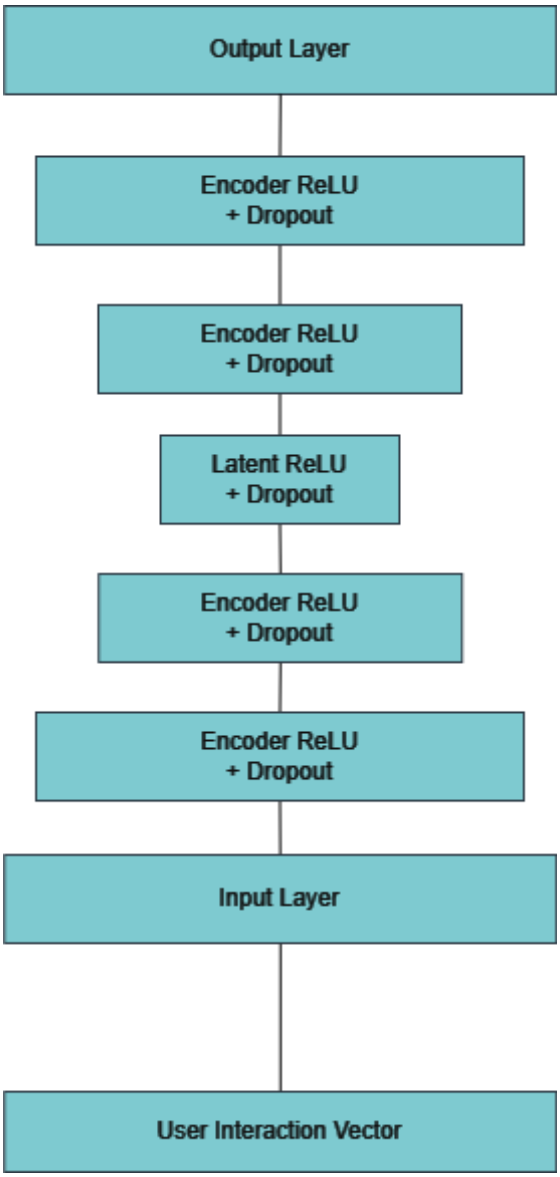


Рисунок 2.8 – Схема моделі AutoRec

2.3.9 Додаткові умови дослідження

Оскільки визначені для оцінки моделей метрики точності та повноти є ортогональними відносно одна одній, оцінювання моделей з урахуванням двох метрик є в значній мірі складним та нераціональним завданням, що потребує

визначення додаткових умов дослідження. З метою урахувати баланс між точністю та повнотою було прийнято рішення щодо використання третьої метрики, котра могла б спростити задачу порівняння ефективності моделей.

Для вирішення вищеописаної проблеми була обрана метрика f1-міра, котра являє собою гармонійне середнє між точністю та повнотою. Простота в обчисленні та збереження інформативної цінності робить дану метрику придатною для використання в задачах оцінки, налаштування та порівняння моделей, що значно спрощує подальші дослідження. Схема обчислення f1-міри наведена у формулі 2.12.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}, \tag{2.12}$$

де $F1$ – значення ефективності прогнозу моделі за метрикою f1-міри;

$Precision$ – оцінка точності для моделі;

$Recall$ – оцінка повноти для моделі.

Для проведення дослідження необхідним є поділ даних на тренувальну та тестувальну вибірки. Для того, щоб вирішити це питання, пропонується ділити загальний обсяг рядків у наборі даних за принципом 3:1, тобто 75% прецедентів відводиться для тренування, решта 25% використовується при верифікації моделі. Щодо самих даних, то в ході дослідження для проведення експериментів будуть використані два датасети. Перший з них передбачає наявність даних щодо клієнтських оцінок, загальна кількість записів в котрому досягає приблизно 500000, кількість товарів – 60000. Другий набір даних є меншим відносно другого, що може дати змогу для відстеження поведінки моделей у різних умовах. Розміри другого датасету складають відповідно 20000 сценаріїв оцінки та 1100 товарів.

Для проведення дослідження також визначені додаткові умови щодо переліку технологій, за допомогою яких проводиться дослідження. У якості

мови програмування обраний Python – один з основних інструментів у вирішенні складних математичних задач та задач машинного навчання. Дана мова програмування має в розпорядженні усі необхідні бібліотеки, котрі в повній мірі дозволяють виконати поставлені завдання. З метою реалізації алгоритмів та структур глибокого навчання використано відповідну платформу PyTorch, для операцій з великими обсягами даних та проміжних перетворень даних – бібліотеки PySpark та Petastorm. Були також використані допоміжні бібліотеки Numpy, Scikit-learn та інші для окремих операцій над даними, тензорами, для доступу до алгоритмів машинного навчання, візуалізації даних, тощо.

2.4 Результати досліджень рекомендаційних моделей

У ході навчання та верифікації моделей було проведено експерименти та встановлено оцінки точності, повноти, ефективності та продуктивності кожної з моделей. Для першого експерименту усі зібрані показники моделей можна побачити на рисунку 2.9. Графік відношення ефективності та продуктивності моделей наведений на рисунку 2.10.

```
C:\Users\HREN\AppData\Local\Programs\Python\Python38-32\python.exe C:\Users\HRE
Введіть:
1 - Для навчання моделі
2 - Для верифікації моделі
3 - Для узагальнення досліджень
Опція: 3

Введіть ім'я файлу результатів: research1

Architecture Name       Precision   Recall  F1-score    Secs For Request
CF_memory               0.5326      0.5976  0.5633      1.39
CB_memory               0.7588      0.1434  0.2413      0.09
WideNDeep               0.5271      0.423   0.4726      0.11
Deep&Cross              0.7915      0.6828  0.7332      0.23
DeepFM                  0.5119      0.5128  0.5124      0.21
NeuMF                   0.7538      0.5006  0.6017      0.09
Autorec                 0.6766      0.3029  0.4185      0.65
```

Рисунок 2.9 – Результат роботи модулю узагальнення першого тесту

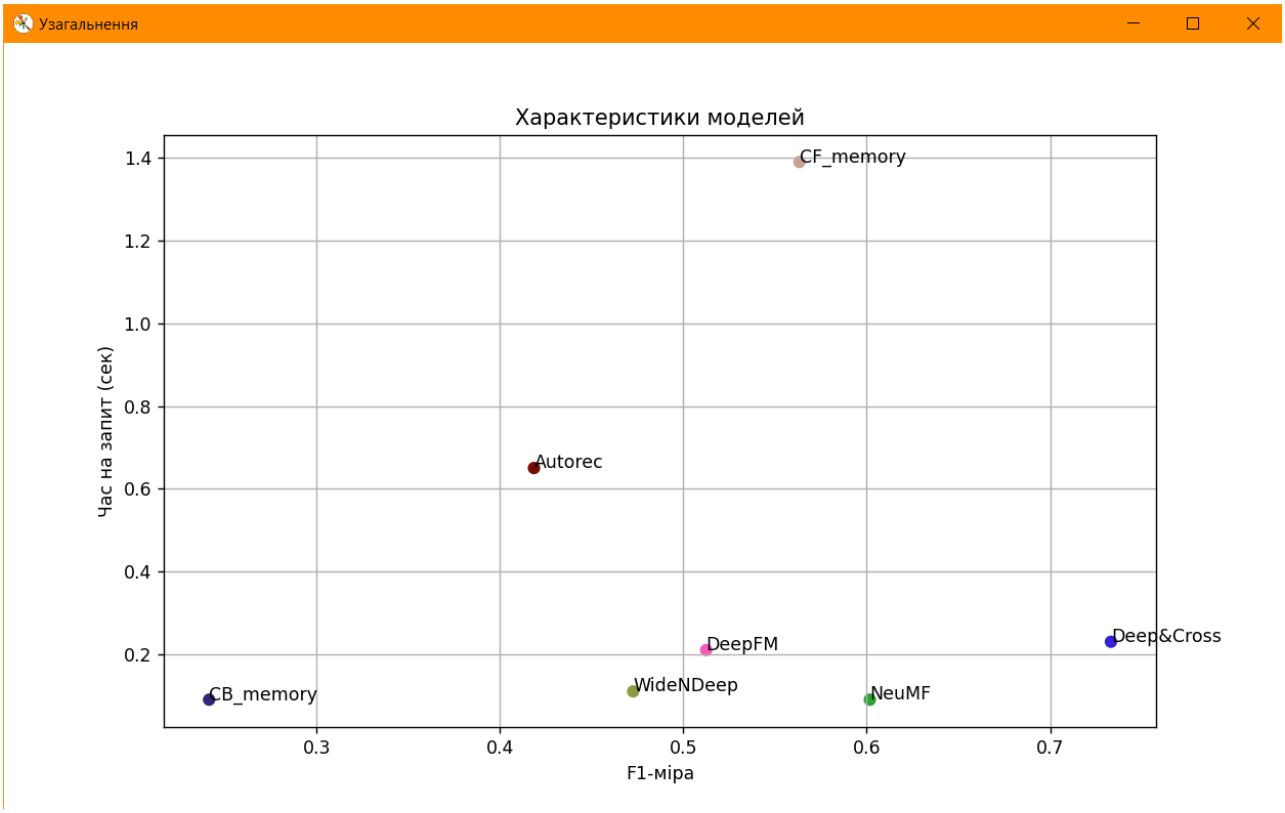


Рисунок 2.10 – Графік порівняльної оцінки моделей першого тесту

З отриманого графіку можна зробити наступні висновки. Розроблені моделі на основі пам’яті у якості рішення для задачі генерації рекомендацій для продажу товарів не є оптимальними. Класична модель колаборативної фільтрації демонструє критично високий для умов експлуатації час на виконання одиничного запиту, хоча й має порівняльно високий показник точності. Модель на основі контенту напроти має позитивний показник швидкодії, що не є важливим, беручи до уваги дуже низький показник точності. Можна зробити припущення, що вищезазначені результати випливають зі специфікою задачі, а саме – з високою розмірністю.

Моделі ж змішаного чи повністю глибокого типу майже усі демонструють допустимі рівні показників. Таким чином, лише архітектура AutoRec виявилася в достатній мірі повільною, приблизно у три рази повільнішою за вищу позицію, та не компенсуючи при цьому швидкодію точністю. Можна зробити припущення, що алгоритм автоенкодингу не є окремо достатньо комплексним для встановлення взаємозв’язків потрібної

глибини. Решта з моделей демонструють точність вище 0.4 та витрачають на запит не більше 0.23 секунд.

Найбільших показників досягли моделі Deep&Cross та NeuFM, що в найбільш високій мірі є представниками відповідних парадигм генерації рекомендацій – на основі контенту та колаборативної фільтрації відповідно. Deep&Cross має приблизно на 13 відсотків вищий показник f1-міри. Проте NeuFM при порівняльно високому показнику точності витрачає найменшу кількість часу на прогнозування.

Щодо другого експерименту, котрий виконувався на значно меншому обсязі даних, результати його проведення наведені на рисунках 2.10 та 2.11 відповідно.

Як можна побачити з графіку, при меншому масштабі бази даних, з котрою оперує рекомендаційна система, у значній мірі відрізняються показники ефективності та продуктивності. Так в цілому усі системи демонструють більші показники точності та швидкодії, а також більш рівні відносно один одного показники ф1-міри, як власне й швидкодії.

```
C:\Users\HREN\AppData\Local\Programs\Python\Python38-32\python.exe C:\Users\H
Введіть:
1 - Для навчання моделі
2 - Для верифікації моделі
3 - Для узагальнення досліджень
Опція: 3

Введіть ім'я файлу результатів: research2

Architecture Name       Precision   Recall  F1-score    Secs For Request
CF_memory               0.7239      0.8225  0.7701      0.74
CB_memory               0.6613      0.4537  0.5382      0.06
WideNDeep               0.5384      0.8049  0.6453      0.1
DeepNCross              0.8117      0.8564  0.8335      0.06
DeepFM                  0.7803      0.8029  0.7915      0.14
NeuMF                   0.8567      0.9407  0.8968      0.09
AutoRec                 0.8979      0.7255  0.8026      0.05
```

Рисунок 2.11 – Результат роботи модулю узагальнення другого тесту

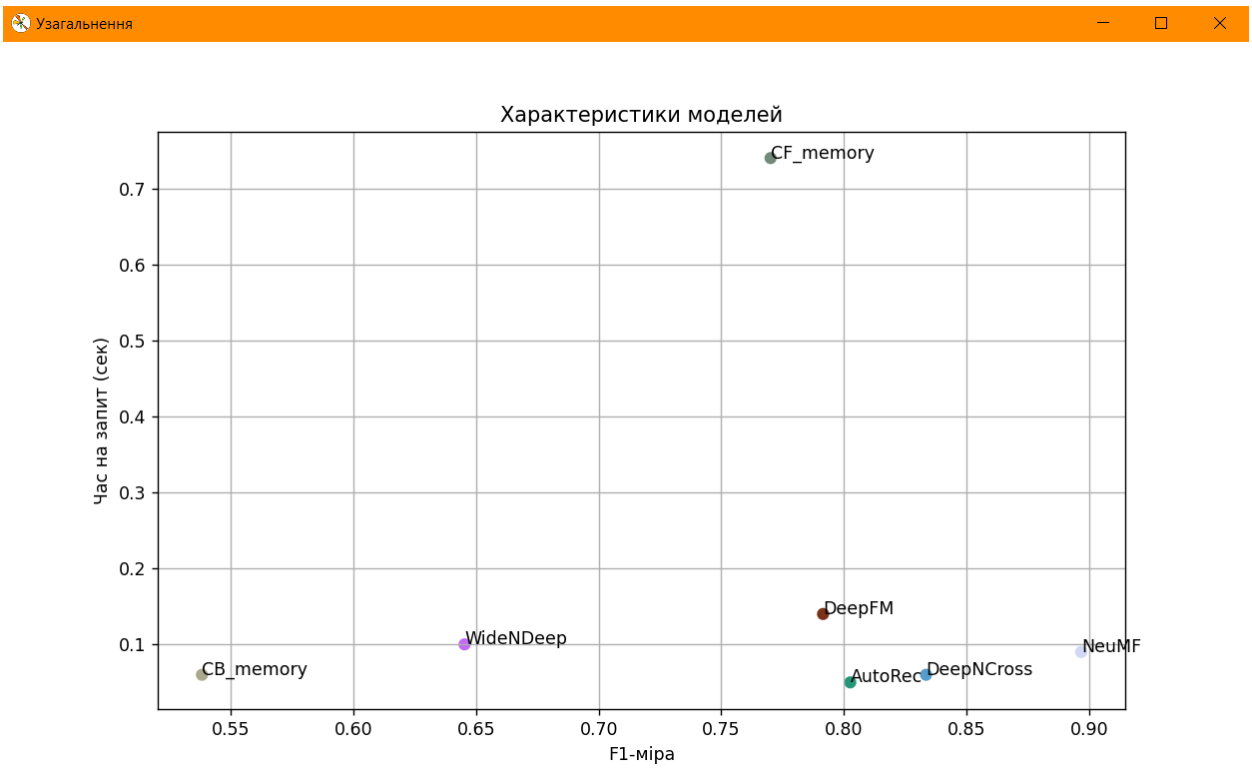


Рисунок 2.12 – Графік порівняльної оцінки моделей другого тесту

Можна також помітити, що моделі, котрі мають в основі конентно-орієнтований лінійний підхід, виявляються в значній мірі менш точними, що може бути пов’язано як с тенденціями саме в цьому датасеті, так й свідчити про менший рівень холодного старту на даному наборі даних. З теоретичних положень маємо, що з підвищенням об’ємів даних, зростає й рівень холодного старту серед умов прогнозування. У додачу до цього, можна побачити, що традиційна реалізація колаборативної фільтрації втратила свою позицію у точності відносно решти моделей. З цієї обставини, а також з урахуванням сказаного перед цим відносно перших двох моделей, можна зробити висновки, що потенціал експлуатації систем на основі моделі є значно вище на менших об’ємах даних. Це спостереження проте не зменшує ефективності систем на основі моделі у більш масштабній середі.

Таким чином системи Deep&Cross а також NeuMF аналогічно мають найбільші показники точності, ніж інші системи, проте NeuMF виявився точнішим за Deep&Cross приблизно на шість відсотків. При цьому показник

швидкодії першого з них не змінився, у той час коли Deep&Cross став значно швидшим.

Можна окремо виділити відмінності в характеристиках, що демонструє архітектура AutoRec. Так підхід на основі автоенкодеру мав відносно інших моделей значно нижчу точність та швидкодію, у даному експерименті демонструє протилежні результати. Модель має точність майже аналогічну з точністю моделі Deep&Cross. Дані відмінності можуть підтверджують припущення, що низька ефективність алгоритму в попередньому експерименті була пов’язана з непридатністю його до великих об’ємів даних. При зменшенні розмірності, як свідчить графік порівняння характеристик, алгоритм працює значно швидше, при цьому може знайти достатній латентний простір ознак, щоб побудувати на його основі задовільну для прогнозування основу взаємозв’язків.

3 ДОСЛІДЖЕННЯ ГІБРИДНИХ СИСТЕМ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ

3.1 Визначення принципу суміщення рекомендаційних моделей

Оскільки мета дослідження – максимально повно ідентифікувати сильні сторони кожної моделі в умовах експлуатації, важливо проаналізувати результати досліджень окремих систем задля того, щоб кількість переліку кандидатів на гібридизацію була максимально великою.

Проведення експерименту дало змогу побачити, що найбільш ключовим фактором, з котрого випливають показники систем в тих чи інших умовах тестування, є розмірність даних. Так, за деякими системами можна відстежити значне зниження точності прогнозу з підвищенням розмірності даних, котрими їм потрібно оперувати. Так само й зі швидкодією: деякі системи виявляються менш стійкими до підвищення обсягів інформації, демонструючи на відповідних валідаційних тестах незадовільно високі значення часу, що витрачається на запит.

Таким чином, оскільки з переліку рекомендаційних архітектур, котрі були досліджені, можна виділити цілий клас систем, придатних до менших вибірок даних, доцільним буде спроєктувати дві окремі системи – систему для вкрай великих обсягів даних та систему для порівняльно менших обсягів даних, котра буде використовувати аналізаційні алгоритми що є менш придатними в середі з високими габаритами. Іншими словами, за придатністю до експлуатації вони будуть відповідати комерційним проєктам з різними масштабами, та виконання даного фактору можна буде підтвердити за допомогою перевірки на відповідних датасетах.

Проєктування має також виконуватися з орієнтацією на швидкодію. По-перше, обчислення окремих модулів у системах великого масштабу має бути виключно паралельним, бо великі обсяги даних у поєднанні з проблемою високонавантаженості можуть призвести до значного підвищення часу на формування прогнозних оцінок, якщо до ці модулів будуть звертатися,

наприклад, послідовно [26]. По-друге, кількість часу на запит сама по собі в рамках подальших досліджень не є більше метрикою, що потрібно радикально мінімізувати. Потенційно високі показники швидкодії окремих систем варто взяти до уваги з метою використання потенційних часових проміжків для додаткових обчислень. Завдяки такому підходу можна досягти балансу між витратами на обчислювальні ресурси та точністю рекомендацій, що є одним з ключових факторів, беручи до уваги важливість обох показників для предметної області.

Таким чином задачею подальших досліджень є проєктування та подальше дослідження експериментальним шляхом властивостей ефективності двох окремих систем з метою максимальної адаптації кожної з них до визначеної відповідної специфіки: врахування обмежень для більшої системи, реалізація наявності гнучких обмежень – як в часі прогнозу так й опосередковано в складності алгоритмів.

3.2 Опис схем гібридизації рекомендаційних систем

Оскільки в рамках дослідження постає питання об’єднання різних рекомендаційних механізмів у єдину системи, варто визначити усі можливі схеми гібридизації. У цілому можна виділити наступні підходи до злиття алгоритмів [27]:

- Зваженою;
- Змішаною;
- Каскадною;
- З поєднанням ознак;
- З аугментацією ознак;
- З перемиканням.

Змішаний (Weighted) підхід є найбільш толерантним відносно кожного сегменту гібриду. Ідея даного методу полягає в комбінуванні результатів кількох різних рекомендаційних алгоритмів, та подальшому об’єднанні

результатів за певною схемою. Таким чином кожен компонент моделі виконує одну й ту саму функцію відносно один одного, при цьому виконує її незалежно від обставин таких як параметри запиту. Як приклад можна привести схему присвоєння кожному з алгоритмів членів гібриду певну вагу, після чого результати кожного алгоритму зважити та об'єднати для отримання остаточних рекомендацій. Перевага зваженої гібридної системи полягає в тому, що всі можливості системи використовуються в процесі рекомендації простим і зрозумілим способом, що дозволяє відносно легко виконувати постфактум коригувати схему гібриду розподілом ваг чи зміною типу змішення.

Схема композиції з перемиканням (Switching) – є стратегією до організації внутрішньої структури системи, де певні компоненти структури можуть використовуватися чи не використовуватися в залежно від тих чи інших обставин, спрощуючи – система може перемикатися між рекомендаційними техніками. Як приклад, визначений певний критерій, котрий поділяє користувачів на дві групи: так перша підсистема буде використовуватися для однієї групи, друга – для другої. Дана схема, за умови використання точно підібраної схеми перемикання, може як дати значний приріст в точності, так й вирішити ті чи інші специфічні проблеми.

Принцип змішаного гібриду (Mixed) – об’єднання результатів та представлення їх зовнішній системі як єдиний список рекомендацій. Даний метод, незважаючи на простоту, має ряд переваг. По-перше, є можливість об’єднати результати компонентів без будь-яких втрат інформації. Так в певному прецеденті використання системи виключається негативний вплив менш результативного елементу на більш результативний. По-друге, використання змішаного гібриду має позитивний вплив на прогноз в умовах холодного старту: система намагається підібрати як найбільшу кількість достатньо релевантних позицій. Проте задля отримання позитивного ефекту схема змішання має бути досить комплексною.

Системи з поєднанням ознак (Feature Combination) це спосіб злиття контентного та колаборативного підходів. Даний принцип полягає в тому, щоб

розглядати колаборативну інформацію просто як додаткові дані ознак, асоційовані з кожним прикладом, після чого застосувати контекнтно-орієнтовані алгоритми до цього доповненого набору даних. Іншими словами, вихідні дані одного алгоритму використовуються як додаткові ознаки для іншого алгоритму, що дозволяє системі розглядати спільні дані, не покладаючись виключно на них.

Ще однією з можливих схем є аугментація ознак (Feature Augmentation). У цьому підході один алгоритм генерує ознаки, котрі після цього використовуються іншим алгоритмом. Таким чином шляхом послідовного використання алгоритмів Feature Augmentation фокусується на збагаченні набору ознак для поліпшення рекомендацій за допомогою додаткових даних від іншої моделі. Аугментація ознак є ще одним з методів для поліпшення продуктивності базової моделі-ядра без необхідності втручання в сам алгоритм та внесення потенційно непередбачуваних в умовах експлуатації модифікацій.

Аналогічно зі схемою аугментації ознак, каскадна схема (Cascade) також передбачає послідовне використання рекомендаційних механізмів з метою підвищення ефективності системи. Проте на відміну від попереднього підходу каскадний підхід виключає використання вихідних даних моделі одного рівня в якості вхідних даних моделі наступного рівня в прямому сенсі. Даний алгоритм передбачає в більшості спочатку застосування однієї рекомендаційної техніки для створення грубого ранжування кандидатів, після чого використання другої техніки для уточнення рекомендацій серед цього набору кандидатів. Таким чином мета використання каскадного підходу – мінімізація обчислень через уникнення малоефективних та заздалегідь безрезультатних дій системою-ядром над низько пріоритетними елементами, які вже точно розрізнені допоміжною першою технікою.

3.3 Додання додаткових вхідних даних прогнозування

Оскільки метою проведення першого етапу було порівняння рекомендаційних підходів між собою та репрезентація рекомендаційних парадигм у чистому вигляді, системи в більшості були спрямовані на обробку строго визначеної інформації, пов’язаної з відповідними парадигмами, такої як перелік попередніх оцінок клієнта чи його вподобань. Таким чином попередні системи не приймали жодного контексту окрім вищеописаних даних. Оскільки метою наступного етапу є максимізація результатів при вирішенні певних задач, має сенс модифікувати системи додатковими даними, врахування котрих може покращити якість прогнозу. Таким чином було прийнято рішення щодо введення додаткових даних, котрими в нашому випадку, з урахуванням можливостей кожної окремої системи, специфіки задачі, що вирішується, а також наборами даних, котрі доступні для проведення дослідження, будуть демографічна інформація клієнта.

Сама обробка демографічних даних має за мету категоризацію клієнтів за персональними атрибутами, такими як вік та стать. Таким чином глибока система формує на основі взаємозв’язків, ідентифікованих при навчанні, демографічні класи [28], котрі є незалежною від вже досліджених рекомендаційних парадигм типом інформації, проте збагачує представлення системи про клієнта та його вподобання.

Тому випливаючи з наявних для дослідження даних можна визначити наступний перелік інформації, що буде відокремлюватися для подачі в рекомендаційну систему:

- Країна, в котрій проживає клієнт;
- Місто, в котрому проживає клієнт;
- Вік клієнта;
- Стать клієнта;
- Ім’я, друге ім’я, прізвище, тощо;
- Дата реєстрації в системі.

Також, певний вплив на якість прогнозу може дати подання на вхід рекомендатору певного контексту. У якості цього контексту, з урахуванням доступної для проведення дослідження інформації, доцільно використати дату оцінки. Дана інформація в значній мірі може впливати на рішення користувача, оскільки наприклад деякі товари придбаються переважно в певні сезони, або деякі користувачі схильні до діяльності в певні дні тижня. Тому застосовуючи прості за принципом механізми кодування на даному полі можна отримати вкрай цінну для системи інформацію, котра потенційно може мати великий вплив на ефективність моделі, незалежно від її специфіки та контексту експлуатації.

Таким чином, приклад додаткових демографічних та контекстуальних даних на вхід системи наведений на рисунку 3.1.

reviewerID	reviewerName	location	country	age	sex	registerDate	reviewTime
A00000262KYZUE4J5...	Steven N Elich	New York City	USA	25	M	11 17, 2012	11 21, 2012
A000008615DZQRRI9...	mj waldon	Coventry	UK	32	F	01 5, 2013	01 8, 2013
A00000922W28P2OCH...	Gabriel Merrill	Sydney	Australia	49	M	02 11, 2014	03 24, 2014
A00000922W28P2OCH...	Gabriel Merrill	Sydney	Australia	49	M	02 11, 2014	03 24, 2014
A00000922W28P2OCH...	Gabriel Merrill	Sydney	Australia	49	M	02 11, 2014	03 24, 2014
A00000922W28P2OCH...	Gabriel Merrill	Sydney	Australia	49	M	02 11, 2014	03 29, 2014
A000013090ZI3HIT9N5V	Elvis florian	Hannover	Germany	19	M	07 3, 2013	08 12, 2013
A00001362Q1PGIX2F...	Pamela Bellamy	Québec	Canada	25	F	08 30, 2013	08 31, 2013
A00001362Q1PGIX2F...	Pamela Bellamy	Québec	Canada	25	F	08 30, 2013	08 31, 2013
A00001483M88NBD66...	JARPD	Tokyo	Japan	38	M	10 4, 2012	11 19, 2012
A0000188NWOSI5X2PMSN	paula george	Dijon	France	30	F	04 14, 2014	04 21, 2014
A0000196KBA0ICH151EG	Khalid	Damanhur	Egypt	51	M	11 22, 2013	11 23, 2013
A000023026XVLM97B...	T.Dobrowolski	Katowice	Poland	49	F	04 16, 2014	04 19, 2014
A00003262KNLZOSMM...	Harue Rojas	Veracruz	Mexico	74	F	01 20, 2013	02 18, 2013
A00003262KNLZOSMM...	Harue Rojas	Veracruz	Mexico	74	F	01 20, 2013	02 18, 2013
A00003262KNLZOSMM...	Harue Rojas	Veracruz	Mexico	74	F	01 20, 2013	02 18, 2013
A00003322NZ9C82Y4...	Kyle Downey	Dublin	Ireland	22	M	05 14, 2014	06 26, 2014
A00003323X6I53YWRGN0	big show	Dallas	USA	50	M	11 17, 2012	11 17, 2012
A00003783ISYSII6M...	Joseph Consuegra	Gijón	Spain	33	M	10 16, 2013	10 17, 2013
A0000440NYTE2D2YB089	Kristen S.	Tridentum	Italy	42	F	09 23, 2013	09 26, 2013

Рисунок 3.1 – Приклад додаткових демографічних даних

Проте описані типи даних потребують додаткової обробки. Варто заздалегідь визначити усі відповідні рішення для даних задля уникнення

необхідності опису методів для кожної схемі, що буде розроблятися. Як вже було зазначено на схемах попередніх моделей, текстові ознаки, такі як ім’я клієнта, ефективним буде кодування за допомогою традиційного алгоритму ембедингу. Для полів типу дати та типів, що відповідають географічним об’єктам, доцільно використати специфічні рішення. При подальшому проєктуванні схем рекомендаційних систем алгоритми, що будуть описані, будуть фігурувати як відповідні модулі.

Так для географічних об’єктів варто застосувати алгоритм Loc2Vec. Оскільки особливістю методу Text Embedding є врахування семантичних особливостей слів, дана властивість не несе значущої користі для обробки географічних назв конкретно у випадку системи, що проєктується. Алгоритм Loc2Vec напроти спрямований виключно на встановлення та збереження інформації про просторові взаємозв'язки та характеристики тої чи іншої локації [29]. Локації, які розташовані поруч або мають схожі характеристики, матимуть близькі вектори у відповідному багатовимірному просторі. Таким чином, перед подачею на вхід даного блоку дані локації та країни необхідно попередньо конкатенувати.

Для дати, у свою чергу, варто застосувати схему Date2Vec, що являє собою відносно просте виділення компонентів, тобто розклад поля дати на день, місяць та рік та наступну конкатенацію окремих значень у відповідне числове представлення. Для явного врахування дню тижня варто також передати це поле в блок конкатенації, попередньо застосувавши до відповідного домену алгоритм Label Encoding.

3.4 Проєктування високомасштабної системи рекомендації товарів

Для проєктування першої системи, що як вже було зазначено, спрямована на максимізацію точності в умовах високої кількості даних, варто використати дві системи, що мають найвищі показники при тестуванні на відповідному датасеті. Таким чином, були обрані ортогональні за принципом алгоритми

Deep&Cross та NeuMF. Оскільки таке поєднання моделей випливає в гібрид колаборативного підходу з контентним, результуюча модель потенційно має більшу точність. У подальшому дослідження при звертанні до вищеописаної моделі буде використовуватися назва «Semi_CH». Схема моделі на ведена на рисунку 3.2.

Що до швидкодії, то схема масштабованого рекомендатору має бути максимально орієнтованою на оптимізацію даної характеристики, інакше вона може стати слабким місцем такої системи. Модель Deep&Cross, так само як модель NeuMF, має у своєму складі глибокий підкомпонент, що являє собою багатошаровий перцептрон із нелінійною функцією активації ReLU всередені. Дана обставина має позитивний вплив на швидкодію гібриду, оскільки наявність спільного елементу означає паралельну роботу трьох модулів замість чотирьох з можливістю використання даного спільного елементу з унікальними модулями обох систем з метою відтворення повноцінної продуктивності кожної з них.

Вищеописана властивість в певній мірі дає потенційній системі гнучкості, оскільки, наприклад, модуль GMF можна обчислити у будь-який час незалежно від перцептрону, після чого можна було б об’єднати результати без негативних наслідків для продуктивності. Проте, саме специфіка масштабованості накладає обмеження на схему системи – вона має бути виключно паралельною, оскільки послідовні обчислення мають адитивний вплив на час роботи гібридного алгоритму. Можна допустили, як приклад, що з метою підвищити швидкодію системи, гарантовано виконувалося б обчислення лише NeuMF, та Cross-компонент керувався б перемиканням, котре обчислювалося б у ході роботи системи, наприклад, на основі того, чи достатнім є результат роботи NeuMF. У випадку, якщо системі все ж таки потрібно задіяти перехресну мережу, на момент сигналу на перемикач усі обчислення пов’язані з протилежною моделлю вже були б виконані. Тому час роботи алгоритму в даній ситуації міг би досягати в окремих випадках недопустимих значень в контексті високонавантаженості. Тому усі три компонента мають виконуватися

паралельно, після чого об’єднуватися за певною схемою. У якості такої схеми варто використати змішаний принцип, тобто конкатенацію результатів роботи, оскільки обидві моделі продемонстрували в більшості однаковий чи як найменш співставний показник ефективність, та розподіл кредитів важливості між ними не є доцільним.

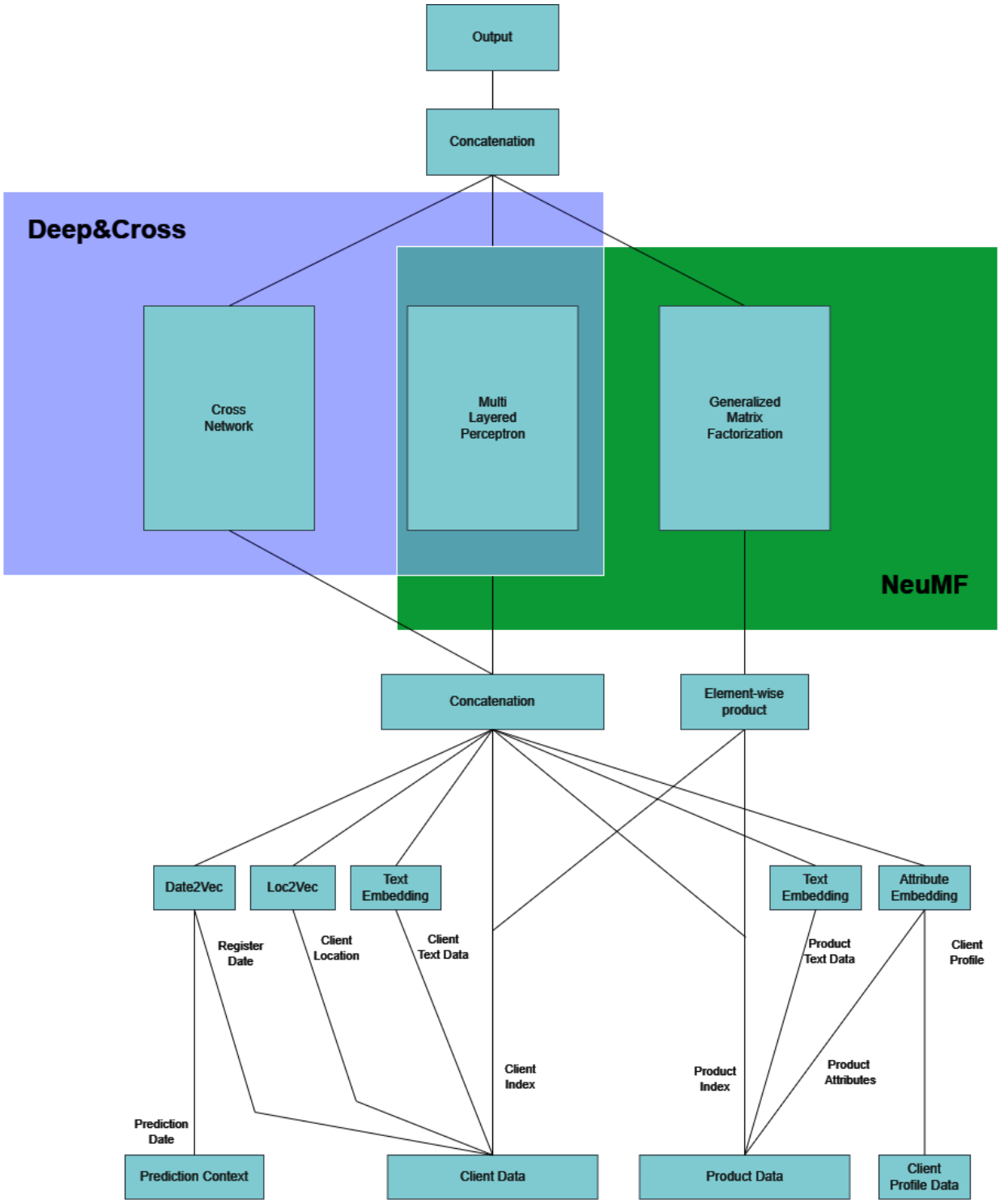


Рисунок 3.2 – Схема моделі Semi_CH

На вхід подаються чотири типи даних – дані клієнта, дані продукту, дані профілю клієнта та дані контексту, тобто поточна дата на момент прогнозування. Дані клієнта діляться на демографічні дані та індекс, дані продукту – на текстові дані, дані значень атрибутів та індекс. Додаткові демографічні показники, як видно зі схеми, подаються до модулів Cross та Deep, оскільки алгоритм GMF є орієнтованим виключно на роботу з матрицями, тобто в нашому випадку роботою з колаборативними взаємозв'язками, до котрий демографічні дані не мають відношення.

Демографічні дані клієнта, як видно зі схеми моделі «Semi_CH», конкатенуються також з даними продукту, клієнтським профілем, комбінацією індексів клієнта та товару, а також контекстом прогнозування. Алгоритм GMF, як й у випадку з його високомасштабним аналогом з дослідження окремих рекомендаційних моделей, є низькорозмірним, оскільки задачею системи, що проєктується, є прогнозування оцінки для одного конкретного товару для одного конкретного клієнта.

3.5 Проєктування низькомасштабної системи рекомендації товарів

Для другої системи з переліку базових моделей були обрані NeuMF та AutoRec. Система AutoRec, як можна побачити з експерименту оцінки окремих моделей, є більш орієнтованою на відносно невеликі за масштабом системи, при цьому виконує задачу прогнозування з достатнім рівнем точності. У цей ж час модель NeuFM є однією з найбільш точних в цілому, при цьому має значну взаємодію на концептуальному рівні. Оскільки обидві моделі передбачають реалізацію багатовимірну обробку векторів взаємодії клієнта з товаром, можна стверджувати, що гібрид цих двох систем буде являти собою колаборативну систему-гібрид, усі модулі котрої підтримують на рівні реалізації прогнозування оцінок для багатьох товарів.

Беручи до уваги останню описану властивість, доцільно буде спроєктувати гібридну систему таким чином, щоб вона могла за один сеанс

роботи прогнозувати рекомендації одразу для всього переліку товарів в базі даних системи. Так автоенкодер, так само як алгоритм GMF, будуть приймати повний вектор взаємодії клієнта з товарами, з тою різницею, що останній алгоритм потребує додаткової обробки цієї інформації операцією «Element-wise product».

Оскільки для даної системи не має значних обмежень зі сторони масштабованості, доцільно використати в схемі елемент перемикання, для більшої адаптації рекомендатору до зовнішніх та внутрішніх факторів. Таким чином, при вкрай великій завантаженості серверу можна контролювати час, котрий система витрачає на роботу шляхом обмеження переліку алгоритмів, котрі вона застосовує. Так AutoRec сам по собі є достатньо точним модулем, на схемі 3.3 ж пропонується в обмеженому варіанті частково використати NeuMF.

Так схема містить перемикач, та у випадку, якщо він не спрацьовує, прогноз виконується автоенкодером та компонентом моделі NeuMF – багатошаровим перцептроном, котрий окрім колаборативної інформації клієнта обробляє також й демографічну інформацію. Таким чином значна частина алгоритмів зосереджена в першій половині системи, тому її точність в найменш оптимістичних умовах знижується некритичним чином. Відносно ж важкий за концепцією алгоритм матричної факторизації використовується лише у другій половині системи, та при оптимістичному сигналі покращує прогноз, збагачуючи його оцінками на основі матричної факторизації, котрі конкатенуються з оцінками, отриманими в першій половині, котрі після обчислення гарантовано потрапляють до кінцевого прогнозу. Така схема орієнтована на використання надлишків швидкодії, що випливають зі високих показників швидкості обраних алгоритмів, при цьому може врахувати можливість прогнозу при наявності обмеженої доступності обчислювальних ресурсів на певному часовому проміжку.

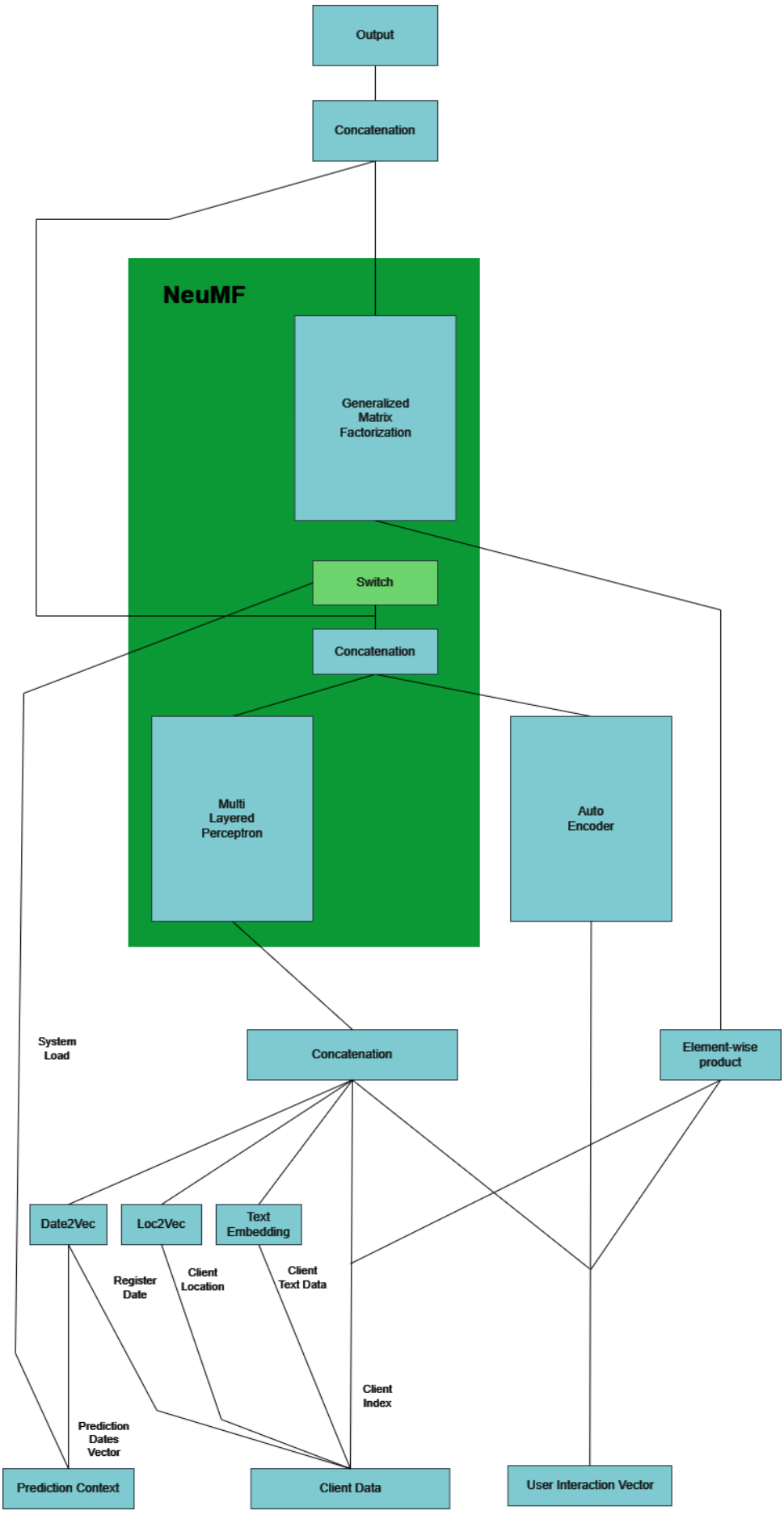


Рисунок 3.3 – Схема моделі Deep_Restricted

У свою чергу елемент перемикання приймає на вхід два значення, в залежності від котрих генерує сигнал впевненості – показник від нуля до одиниці, котрий відповідно перетворюється в логічний сигнал. До вищезазначених двох факторів відносяться, як вже було сказано, завантаженість системи, а також показник задоволеності відповіддю, котрий відображає ступінь, якій кількість задовільних оцінок та впевненість в прогнозах відповідають мінімальним вимогам, виходячи з котрих можна надавати рекомендації клієнту. Як перший, так й другий показники приймають значення від нуля до одиниці, при цьому для показника задоволеності відповіддю один означає, що кількість оцінок, достатніх, щоб потрапити в рекомендації є дорівнює або більше ніж двократна мінімальна довжина рекомендацій, котра в нашому випадку дорівнює десяти, а показник буде дорівнювати одиниці при значеннях двадцять та більше.

Сама ж схема перемикання побудована таким чином, щоб при критично великих значеннях завантаженості, незалежно від повноти відповіді, надавати сигнал відмови. При помірних значеннях навантаженості, відповідь контролюється значенням повноти відповіді, у випадках, коли навантаженість незначна доцільно для додаткової точності рекомендацій використовувати другу половину системи, тобто алгоритм GMF. Оскільки в ході проведення експерименту не має можливості контролювати навантаженість пристрою та імітувати відповідні умови, показник навантаженості ході експерименту не розраховується та є абстрактним параметром впливу на систему. Принцип, на котрому заснований перемикач, наведений у формулі 3.1, принцип обчислення задовільності відповіддю першої підсистеми – у формулі 3.2, обчислення задовільності оцінкою товару – у формулі 3.3.

$$switch(l,e) = (1-l) * \big(1-\sigma(l-t_h)\big) + (1-e) * \sigma(l-t_l) \quad (3.1)$$
$$* \big(1-\sigma(l-t_h)\big) + \left(1-\frac{e}{2}\right) * (1-\sigma(l-t_l)),$$

де $switch(l,e)$ – результат роботи перемикача

l – показник завантаженості серверу;

e – показник повноти відповіді першої підсистеми;

t_l – нижній поріг, при котрому важливість l є незначною, встановлено 0.3;

t_h– верхній поріг, при котрому важливість l є критичною, встановлено 0.8;

$\sigma(x)$ – функція сигмоїди.

$$e = \sum_{i=1}^{N} apply(r_i, c_i), \tag{3.2}$$

де e – повнота відповіді першої підсистеми

l – показник завантаженості серверу;

$apply(r_i, c_i)$ – задовільність оцінкою і та потрапляння в рекомендації;

N – кількість оцінок тобто товарів у системі – 1100;

r_i – прогноз оцінки для товару і;

c_i – впевненість прогнозу для оцінки товара і.

$$apply(r_i, c_i) = \begin{cases} 1, & r_i \geq R_{min} \text{ та } c_i \geq C_{min} \\ 0, & \text{інакше} \end{cases}, \tag{2.9}$$

де $apply(r_i, c_i)$ – обчислення задовільності оцінки товару і

R_{min} – мінімальна оцінка для потрапляння в рекомендації, встановлено 4;

C_{min} – мінімальний коефіцієнт впевненості прогнозу, встановлено 0.7.

3.6 Умови дослідження гібридних рекомендаційних моделей

Варто зазначити, що умови дослідження гібридних РС є аналогічними вже описаними у попередніх розділах умовам дослідження окремих рекомендаційних моделей. Таким чином, для верифікації моделей будуть використовуватися ті ж самі датасети – один з кількістю записів оцінок близько п’ятисот тисяч та кількістю товарів близько шістдесяти тисяч, другий з відповідними обсягами в двадцять тисяч та одну тисячу відповідно. Тому

тренувальні та тестові вибірки для першого та другого експериментів будуть складати приблизно 375000:125000 та 15000:5000 відповідно. У якості ж метрик оцінки так само беруться швидкодія, точність, повнота та f1-міра, при цьому. Проте оскільки кількість результатів порівняльно з першим дослідженням є меншою, було прийнято рішення схематично об’єднати два експерименти в один, для зручності порівняння.

При цьому, з урахуванням специфіки другої менш масштабованої системи, варто визначити додаткові правила відносно даної системи. Так, оскільки навантаженість системи є керованим параметром, варто перевірити точність та швидкодію при різних її показниках. Так для першого тесту варто використати наступну схему: для трьох чвертей усіх прогнозів варто позначити навантаженість 0.8 та більше. Для другого тесту навантаженість для відповідної частки буде складати 0.3 та менше. Решта ж некерованих явним чином записів ініціалізуються випадковим способом.

3.7 Результати досліджень гібридних рекомендаційних моделей

На основі вихідних даних проведеного проєктування гібридних моделей були побудовані та навчені моделі рекомендаційних систем «Semi_CH» та «Deep_Restricted». Кожна модель у результаті має два екземпляри – по одному на кожний експеримент. Таким чином була проведена верифікація, у результаті котрої отримані показники моделей у ході кожного експерименту. Результати першого з них наведені на рисунках 3.4 та 3.5.

Architecture Name	Precision	Recall	F1-score	Secs For Request
SH_unscaled	0.9696	0.9303	0.9496	0.2
DRLow_unscaled	0.7515	0.931	0.8317	0.08
DRHigh_unscaled	0.9148	0.9331	0.9239	0.17
SH_scaled	0.7	0.9131	0.7925	0.36
DRLow_scaled	0.6788	0.4377	0.5323	0.66
DRHigh_scaled	0.6775	0.5195	0.5881	0.85

Рисунок 3.4 – Результат роботи модулю узагальнення для гібридних моделей

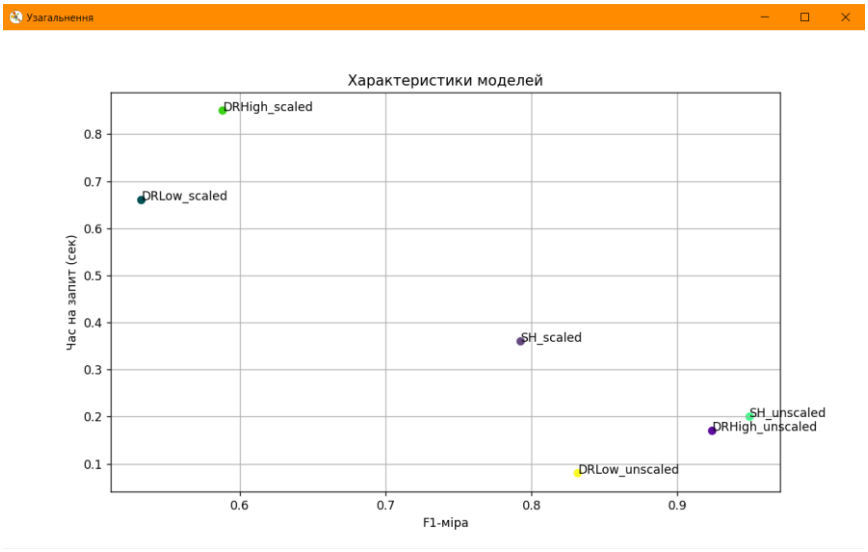


Рисунок 3.5 – Графік порівняльної оцінки гібридних моделей

Для додаткової простоти порівняння показники окремих модулів, а також показники гібридних рекомендаторів були зібрані та наведені у відповідній таблиці 3.1.

Таблиця 3.1 – Результати роботи моделей

Назва моделі	F1 у низькому масштабі	F1 у високому масштабі	Швидкість у низькому масштабі	Швидкість у високому масштабі
Deep&Cross	0.8335	0.7332	0.06	0.23
NeuMF	0.8968	0.6017	0.09	0.09
AutoRec	0.8026	0.4185	0.05	0.65
Semi_CH	0.9496	0.7925	0.2	0.36
DeepRestricted (Умови навантаження)	0.8317	0.5323	0.08	0.66
DeepRestricted (Умови без навантаження)	0.9239	0.6775	0.17	0.85

Виходячи з отриманих результатв, можна сказати що поєднання колаборативного алгоритму NeuMF та контентного алгоритму Deep&Cross дало змогу в значній мірі підвищити точність рекомендацій порівняльно з окремими алгоритмами, на основі котрих створювався гібрид. Так як в умовах низького масштабу так й в умовах великих обсягів даних модель Semi_CH демонструє найбільш показники точності з усіх розглянутих в ході дослідження моделей. Так показник точності для більшого датасету склав майже 80%, що на 6% більше за показник архітектури NeuMF. Проте, як можна побачити з показників швидкодії в обох випадках відносно масштабу час роботи гібриду моделей є навіть вищим за адитивний його компонент. Це свідчить проти припущення, що час роботу гибриду частково перекриваючих один одну моделей буде меншою за суму окремих їх показників, проте підтверджує, що обрана паралельна схема є ефективною, оскільки при послідовному розміщенні швидкість могла бути значно меншою. Можна допустити, що такий результат був отриманий через додаткову обробку демографічних даних при кожному окремому звертанні до рекомендаційної системи.

Показники моделі DeepRestricted у свою чергу демонструють протилежні за позитивністю тенденції в залежності від масштабованості даних експерименту, що збігається зі зробленими раніше відповідними припущеннями. Так при експлуатації на великому за обсягом датасеті гібридна система в обох випадках навантаження демонструє значний приріст точності, проте час, котрий вона витрачає на прогнозування не є припустимим, навіть при використанні механізмів його зменшення. Таким чином така система не призначена для масштабування, тому що не максимізує точність та критичним чином втрачає швидкодію.

Проте випробування на меншому за розміром датасеті в цілому показують факт досягнення поставлених для такої архітектури цілей. Таким чином, при навантаженні на сервер та частковому задіянні повного переліку алгоритмів модель все ще демонструє точність приблизно 83%, що на 6% менше за окремий алгоритм NeuMF. У цей ж час швидкість, котру потрібно

максимізувати, підвищується більш ніж у два рази. При відсутності ж обмежень та роботі на повну система досягає точності на 8% більше, при цьому має прийнятний показник швидкодії 0.17 секунд.

Узагальнюючи усе вищезазначене про модель, можна зробити висновок, що додання глибокого перцептрону разом з обробкою демографічної інформації значно підвищує точність прогнозу автоенкодером. При цьому продуктивність алгоритму GMF у кінцевому результаті є значно вищою за очікувану. Таким чином отримана модель може адаптуватися до різних умов та працює як на максимізацію точності, так й з урахуванням необхідності економії часу.

ВИСНОВКИ

У ході дослідження проведено аналіз предметної області формування рекомендацій товарів. Досліджено поняття рекомендаційної системи, ідентифіковані основні проблеми, що порушуються в межах задачі, та підходи до надання рекомендацій. Таким чином була підготовлена теоретична основа для проведення подальших досліджень.

Перед дослідженням рекомендаційних моделей визначено умови до дослідження: встановлено показники систем, що підлягають спостереженню, визначено метрики, що описують дані показники, накладено обмеження, котрі пов'язані зі специфікою предметної області, а також визначено часкові інтерфейси до систем. Таким чином забезпечено найбільший ступінь об'єктивності досліджень шляхом підбору максимально репрезентативних метрик та стандартизації вимог до систем.

Обрано схему вирішення проблеми передачі даних атрибутів товарів та послідовності товарів клієнта, а також обрано підхід для реалізації роботи з матрицями взаємодії. Завдяки цьому рішення, що були використані при проєктуванні та реалізації систем, стали максимально орієнтованими на адаптацію до специфіки предметної області – великих обсягів даних.

Проведено проєктування окремих моделей, де були розглянуті принципи відповідних архітектур та визначені особливості реалізації в умовах предметної області. Було проведене тестування моделей з використанням додаткової збалансованої метрики для підвищення зручності оцінки ефективності моделей. Результати тестування моделей проаналізовано відносно точності та швидкодії, зроблено висновки щодо їх характеристик для використання аналізу як основи для їх подальшого розглядання в якості компонентів гібридних систем.

Визначено принцип суміщення моделей, що дало змогу визначити орієнтовані цілі для гібридів та задачу подальшого дослідження. Розглянуті різні схеми гібридизації рекомендаційних моделей, після чого спроєктовано дві гібридні рекомендаційні системи. У ході проєктування системи було

спроєктовано відносно специфіки окремих моделей та цілей відповідних загальних гібридів, таким чином оптимізовано гібридні системи, а їх роботу максимізовано відносно особистих для кожної цілей. Зі створеними гібридними моделями проведено експеримент, у результаті котрого було проаналізовано показники роботи кожної моделі в різних умовах, що дало змогу підтвердити чи відхилити попередні гіпотези щодо їх властивостей.

Узагальнюючи результати проведених експериментів, можна сказати, що проєктування рекомендаційних систем на основі моделей в цілому є більш результативною практикою. Вихідні дані тестів підтверджують теоретичні положення, що використання глибокого навчання в рекомендаційних системах забезпечує баланс між точністю та швидкодією, водночас в значній мірі підвищує рівень масштабованості системи. Отримані результати також свідчать про високу ефективність поєднання різних за принципом алгоритмів, оскільки чим більшу область логічних припущень охоплює система, тим більш високі показники точності вона демонструє. Проте такі гібриди, як було встановлено, потребують значної орієнтації на оптимізацію швидкодії при проєктуванні системи.

Подальші дослідження можуть включати розширення переліку моделей, розглядання найбільш нових підходів, котрі ще не достатньо досліджені. Також доцільно проведення нових експериментів для дослідження поведінки окремих моделей в різних умовах та ідентифікації нових властивостей. Можливе дослідження найменш поширених алгоритмів для розглядання в якості ядра для окремого рекомендаційного алгоритму.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Aggarwal C. C. Recommender Systems. Cham : Springer International Publishing, 2016. URL: https://doi.org/10.1007/978-3-319-29659-3 (date of access: 04.03.2024).

2. Collaborative Filtering Recommender Systems / J. B. Schafer et al. The Adaptive Web. Berlin, Heidelberg. P. 291–324. URL: https://doi.org/10.1007/978-3-540-72079-9_9 (date of access: 24.03.2024).

3. Schafer J. B., Konstan J., Riedi J. Recommender systems in e-commerce. the 1st ACM conference, Denver, Colorado, United States, 3–5 November 1999. New York, New York, USA, 1999. URL: https://doi.org/10.1145/336992.337035 (date of access: 25.03.2024).

4. Saveski M., Mantrach A. Item cold-start recommendations. the 8th ACM Conference, Foster City, Silicon Valley, California, USA, 6–10 October 2014. New York, New York, USA, 2014. URL: https://doi.org/10.1145/2645710.2645751 (date of access: 18.03.2024).

5. Felfernig A., Burke R. Constraint-based recommender systems. the 10th international conference, Innsbruck, Austria, 19–22 August 2008. New York, New York, USA, 2008. URL: https://doi.org/10.1145/1409540.1409544 (date of access: 24.03.2024).

6. Knowledge-based recommendation / D. Jannach et al. Recommender Systems. Cambridge. P. 81–123. URL: https://doi.org/10.1017/cbo9780511763113.006 (date of access: 24.03.2024).

7. Item-based collaborative filtering recommendation algorithms / B. Sarwar et al. the tenth international conference, Hong Kong, Hong Kong, 1–5 May 2001. New York, New York, USA, 2001. URL: https://doi.org/10.1145/371920.372071 (date of access: 25.03.2024).

8. Deep Learning Based Recommender System / S. Zhang et al. ACM Computing Surveys. 2019. Vol. 52, no. 1. P. 1–38. URL: https://doi.org/10.1145/3285029 (date of access: 04.03.2024).

9. Гребенюк М., Ситнікова П. Е. Компактна гібридна модель користувача для покращення рекомендаційних систем // Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем (MEICS-2023) : Тези доповідей на VIII Всеукраїнській науково-практичній конференції (22–24 листопада 2023 р.). Дніпро, 2023. С. 106-107.

10. Hybrid recommendation approaches / D. Jannach et al. Recommender Systems. Cambridge. P. 124–142. URL: https://doi.org/10.1017/cbo9780511763113.007 (date of access: 18.03.2024).

11. Çano E., Morisio M. Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis. 2017. Vol. 21, no. 6. P. 1487–1524. URL: https://doi.org/10.3233/ida-163209 (date of access: 11.03.2024).

12. Nagraj S., Palayyan B. P. Personalized E-commerce based recommendation systems using deep-learning techniques. IAES International Journal of Artificial Intelligence (IJ-AI). 2024. Vol. 13, no. 1. P. 610. URL: https://doi.org/10.11591/ijai.v13.i1.pp610-618 (date of access: 19.03.2024).

13. Hybrid Recommendation Systems / E. Adeoye et al. SSRN Electronic Journal. 2024. URL: https://doi.org/10.2139/ssrn.4712941 (date of access: 24.03.2024).

14. A Recommendation System & Their Performance Metrics using several ML Algorithms. International Journal of Engineering and Advanced Technology. 2020. Vol. 9, no. 3. P. 2445–2451. URL: https://doi.org/10.35940/ijeat.c5791.029320 (date of access: 25.03.2024).

15. A Critical Study on Data Leakage in Recommender System Offline Evaluation / Y. Ji et al. ACM Transactions on Information Systems. 2022. URL: https://doi.org/10.1145/3569930 (date of access: 24.03.2024).

16. De Meulemeester H., De Moor B. Unsupervised Embeddings for Categorical Variables. 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, United Kingdom, 19–24 July 2020. 2020. URL: https://doi.org/10.1109/ijcnn48605.2020.9207703 (date of access: 09.06.2024).

17. More T., Kohle P. S. Recommendation System Using Matrix Factorization. International Journal for Research in Applied Science and Engineering Technology. 2022. Vol. 10, no. 9. P. 355–359. URL: https://doi.org/10.22214/ijraset.2022.46615 (date of access: 09.06.2024).

18. Embedding-Augmented Generalized Matrix Factorization for Recommendation with Implicit Feedback / L. Feng et al. IEEE Intelligent Systems. 2020. P. 1. URL: https://doi.org/10.1109/mis.2020.3036136 (date of access: 11.06.2024).

19. Distance Functions / G. I. Webb et al. Encyclopedia of Machine Learning. Boston, MA, 2011. P. 289. URL: https://doi.org/10.1007/978-0-387-30164-8_225 (date of access: 08.06.2024).

20. Use of Autoencoder and One-Hot Encoding for Customer Segmentation / T. Smutek et al. EUROPEAN RESEARCH STUDIES JOURNAL. 2024. XXVII, Special Issue 2. P. 72–82. URL: https://doi.org/10.35808/ersj/3388 (date of access: 11.06.2024).

21. Shao L., Wu D., Li X. Learning Deep and Wide: A Spectral Method for Learning Deep Networks. IEEE Transactions on Neural Networks and Learning Systems. 2014. Vol. 25, no. 12. P. 2303–2308. URL: https://doi.org/10.1109/tnnls.2014.2308519 (date of access: 07.06.2024).

22. Kütük Y. Activation Functions: Activation Functions in Deep Learning with LaTeX Applications. Lang GmbH, Internationaler Verlag der Wissenschaften, Peter, 2022.

23. Zhou J., Zhang Q., Li X. Fuzzy factorization machine. Information Sciences. 2021. Vol. 546. P. 1135–1147. URL: https://doi.org/10.1016/j.ins.2020.09.067 (date of access: 09.06.2024).

24. Deep & Cross Network for Ad Click Predictions / R. Wang et al. KDD '17: The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax NS Canada. New York, NY, USA, 2017. URL: https://doi.org/10.1145/3124749.3124754 (date of access: 07.06.2024).

25. Hierarchical Constrained Variational Autoencoder for interaction-sparse recommendations / N. Li et al. Information Processing & Management. 2024. Vol. 61, no. 3. P. 103641. URL: https://doi.org/10.1016/j.ipm.2024.103641 (date of access: 05.06.2024).

26. Tata R. K. Load Balancing Analyzer: A Recommendation System using Machine Learning. International Journal of Emerging Trends in Engineering Research. 2020. Vol. 8, no. 5. P. 2085–2090. URL: https://doi.org/10.30534/ijeter/2020/99852020 (date of access: 06.06.2024).

27. Burke R. User Modeling and User-Adapted Interaction. 2002. Vol. 12, no. 4. P. 331–370. URL: https://doi.org/10.1023/a:1021240730564 (date of access: 04.03.2024).

28. Jalali M., Gholizadeh H., Hashemi Golpayegani S. A. An improved hybrid recommender system based on collaborative filtering, content based, and demographic filtering. International Journal of Academic Research. 2014. Vol. 6, no. 6. P. 22–28. URL: https://doi.org/10.7813/2075-4124.2014/6-6/a.3 (date of access: 09.06.2024).

29. Loc2Vec-Based Cluster-Level Transition Behavior Mining for Successive POI Recommendation / Y. Wen et al. IEEE Access. 2019. Vol. 7. P. 109311–109319. URL: https://doi.org/10.1109/access.2019.2931075 (date of access: 05.06.2024).

30. Новіков М.В, Міщеряков Ю. В. Дослідження методів проєктування систем рекомендації товарів // 28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., – Харків: ХНУРЕ. С. 445-447.