

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА  
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

КОМП'ЮТЕРНА CRO-СИСТЕМА НА ОСНОВІ МАШИННОГО  
НАВЧАННЯ

COMPUTER CRO-SYSTEM BY USING MACHINE LEARNING

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти  
групи КІ-42

Конотопчик Артем Миколайович

(підпис)

Керівник:

к.т.н., доцент

Мельник Катера Вікторівна

(підпис)

Кваліфікаційну роботу

допущено до захисту

« 07 » червня 2024 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н.Черняшук

« 10 » 01 2024 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Конопчику Артему Миколайовичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Комп'ютерна CRO-система на основі машинного навчання*

Керівник роботи *к.т.н., доцент Мельник Катера Вікторівна*

затверджені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *11.06.2024р.*

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та вітчизняні публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

*Аналіз сучасного стану проблеми, існуючих методів та засобів проведення*

*CRO-досліджень, розробка й обґрунтування рекомендаційної системи, оцінка результатів дослідження*

5. Перелік графічного (ілюстративного) матеріалу:

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Мельник К.В., доцент</i>		
<i>Теоретичне дослідження та практична реалізація</i>	<i>Мельник К.В., доцент</i>		
<i>Практична реалізація CRO-системи</i>	<i>Мельник К.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		_____%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., асистент</i>		

7. Дата видачі завдання 10.01.2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Провести огляд літератури із досліджуваної проблеми</i>	до 15.02.2024 р.	Виконано
2.	<i>Провести аналіз практичного застосування машинного навчання в CRO-системах</i>	до 15.03.2024 р.	Виконано
3.	<i>Розробка та впровадження CRO-системи на основі машинного навчання</i>	до 04.05.2024 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 07.05.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 10.05.2024 р.	Виконано
6.	<i>Формування додатків</i>	до 15.05.2024 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 20.05.2024 р.	Виконано
8.	<i>Нормоконтроль</i>	до 01.06.2024 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 04.06.2024 р.	Виконано
10.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	до 11.06.2024 р.	Виконано

Здобувач вищої освіти

(підпис)

Конотопчик А.М.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Мельник К.В.

(прізвище, ініціали)

## АНОТАЦІЯ

Конотопчик А.М. Комп'ютерна CRO-система на основі машинного навчання. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024. 56 с.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

Перший розділ присвячено огляду предметної області, а саме основні поняття про CRO-системи та машинне навчання. Проводиться аналіз сучасних технологій та перспективу їх використання. Розглядаються методи A/B тестування, веб-аналітики та персоналізації контенту.

В другому розділі здійснено вибір та обґрунтування методів для розробки CRO-системи. Застосовуються методи машинного навчання: глибока нейронна мережа прямого поширення, кластеризація алгоритмом K-means та лінійна регресія. Розглянуто переваги обраних методів та проведений аналіз недоліків.

Третій розділ присвячено опису розробки CRO-системи та імплементації її у веб-сайт онлайн-магазину та аналізу роботи рекомендаційної системи на основі колаборативної фільтрації, кластерного аналізу та лінійної регресії.

Ключові слова: CRO, глибокі нейронні мережі, колаборативна фільтрація, кластеризація, лінійна регресія, машинне навчання, рекомендаційна система.

## ANNOTATION

Konotopchyk A.M. Computer CRO-system by using machine learning. Manuscript.

Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024. 56 p.

Qualification work consists of an introduction, three sections, conclusions, a references, appendices.

The first section is dedicated to an overview of the subject area, discussing the main concepts of CRO systems and machine learning. It describes popular technologies and provides a perspective on their use in achieving the task at hand. Methods such as A/B testing, web analytics, and content personalization are reviewed.

The second section includes the selection and justification of the methods used for developing the CRO system. The chosen machine learning methods are: deep feedforward neural network, K-means clustering algorithm, and linear regression. The advantages of the selected methods are considered, and an analysis of their drawbacks is provided.

The third section is dedicated to describing the development of the CRO system and its implementation on the website of an online store, as well as analyzing the operation of the recommendation system based on collaborative filtering, cluster analysis, and linear regression.

Keywords: CRO, deep neural networks, collaborative filtering, clustering, linear regression, machine learning, recommendation system.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ CRO-СИСТЕМ ТА МАШИННОГО НАВЧАННЯ.....	9
1.1 Визначення та характеристики CRO-систем .....	9
1.2 Огляд методів CRO .....	10
1.3 Огляд рекомендаційних систем.....	11
РОЗДІЛ 2 ПРАКТИЧНЕ ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ В CRO-СИСТЕМАХ.....	14
2.1 Методи машинного навчання для реалізації CRO-системи .....	14
2.2 Огляд алгоритмів CRO-системи.....	17
2.3 Аналіз обраних методів у CRO-системах.....	18
РОЗДІЛ 3 РОЗРОБКА ТА ВПРОВАДЖЕННЯ CRO-СИСТЕМИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ .....	20
3.1 Реалізація та наповнення веб-сайту інформацією .....	20
3.2 Впровадження та використання CRO-системи до веб-сайту .....	26
3.3 Аналіз тестування CRO-системи.....	33
ВИСНОВКИ.....	36
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	38
ДОДАТКИ.....	42

## ВСТУП

Актуальність теми обумовлена тим, що CRO-системи є популярними та важливими для маркетингу. Вони дозволяють підвищити ефективність сайтів, збільшити прибуток, зменшити витрати на приваблення нових відвідувачів. Дані системи будуються на основі різних методів, таких як A/B дослідження, веб-аналітика та персоналізація контенту [1, с. 59]. Однак, існують можливості їх покращити методами машинного навчання. Машинне навчання може покращити ефективність CRO-системи завдяки здатності аналізувати великі обсяги даних, виявляти закономірності та аномалії, адаптуватися до змін у поведінці відвідувачів. Передбачення поведінки користувача було розглянуто у роботі Мостафи Камалія та Тарека Азіз Баблу [2], виявлення закономірностей алгоритмами машинного навчання було розглянуто Ікбалем Саркерем [3], а проведення швидкого аналізу великих даних описано авторами Вей Лі, Юнбао Чай та Фазлулах Хан [4].

Метою роботи є розробка комп'ютерної CRO-системи на основі машинного навчання, яка здатна підвищити рейтинг конверсії сайту за допомогою адаптивних та інтелектуальних методів оптимізації.

Об'єкт дослідження – комп'ютерна CRO-система, що реалізована на основі машинного навчання.

Предмет дослідження – методи машинного навчання для оптимізації рейтингу конверсій.

Завдання, які необхідно виконати:

- Дослідити теоретичні та практичні аспекти CRO-систем.
- Визначити характеристики комп'ютерної CRO-системи.
- Обґрунтувати вибір методів машинного навчання для реалізації CRO-системи.
- Проаналізувати результати роботи CRO-системи на основі машинного навчання.

Апробація. Результати роботи доповідалися на I (VII) міжнародній науково-практичній конференції здобувачів вищої освіти і молодих вчених «Інформаційні технології: теорія і практика», Дніпро, 20-22 березня 2024 року.

Публікації. Опубліковано дві наукові праці [30, 31], а саме, стаття у фаховому виданні та у збірнику тез доповідей інтернет-конференції.

## РОЗДІЛ 1

### ТЕОРЕТИЧНІ ОСНОВИ CRO-СИСТЕМ ТА МАШИННОГО НАВЧАННЯ

#### 1.1 Визначення та характеристики CRO-систем

Оптимізація рейтингу конверсій (Conversion Rate Optimization, CRO) – це систематичний процес збільшення продажів на сайті [5]. CRO використовується для покращення веб-сайту, щоб користувачі з більшою вірогідністю виконували бажані дії, такі як покупка, залишення коментарів чи відгуків та залишення на ресурсі. У випадку онлайн-магазину основними діями, до яких необхідно залучити користувачів, є: покупка товару, залишення відгуку чи перегляд реклами (за рахунок збільшення проведеного часу на сайті).

Комп'ютерна CRO-система – це технічна реалізація автоматизованих методик CRO досліджень, інтегрована в систему взаємодії з користувачем та аналітична складова, що збирає, аналізує та відображає статистичні відомості поведінки користувачів. Дана система характеризується можливістю прийняття швидких змін, персоналізованого аналізу та можливістю проведення тестувань на великій вибірці користувачів.

Характеристикою, що визначає ефективність CRO-системи, є коефіцієнт конверсії, що обраховується за формулою (1.1) [6].

$$CR = \frac{k}{n} * 100\%, \quad (1.1)$$

де CR – рейтинг конверсій;

k – кількість конверсій (відвідувачів, що виконали бажану дію);

n – загальна кількість відвідувачів.

Значення рейтингу конверсій характеризує ефективність маркетингових заходів. Чим вище значення рейтингу конверсій, тим ефективнішим можна вважати реалізоване рішення.

## 1.2 Огляд методів CRO

Оптимізація рейтингу конверсій має на меті модифікувати методи взаємодії з користувачем так, щоб забезпечити утримання та зацікавленість до необхідних елементів веб-сайту найбільшої кількості користувачів, а за можливості – персоналізувати елементи веб-ресурсу, задля максимального збільшення рейтингу конверсій. Методи, що застосовуються для досягнення поставленої мети, можна розділити на три основні категорії, такі як:

- Аналітичні. Дані методи передбачають збір інформації та впровадження рішень на їх основі. Вони характеризуються потребою у природньо сформованих даних самими користувачами, що негативно впливає на швидкість прийняття рішень, однак максимально відповідають реальним рішенням клієнтів.

- Користувацький досвід (UX). Методи користувацького досвіду зосереджені на користувачі, що призводить до кращого загального користувацького досвіду, що збільшує задоволеність користувачів, лояльність та загальний коефіцієнт конверсії, однак, вимагає ітеративного тестування та уточнення, що є трудомістким.

- Методики наповнення контентом. Копірайтинг робить веб-сайт привабливішим та переконливішим, однак, вимагає глибокого розуміння цільової аудиторії та постійного тестування.

### 1.2.1 Аналітичні методи

Аналітичні методи мають на меті зафіксувати поведінку користувача та дослідити її. Для перевірки поведінки користувача використовується запис сеансу, що фіксує всі, або необхідні, поведінкові ознаки користувача [7]. Результати поведінки користувача репрезентуються у вигляді числових значень, текстового звіту або побудованих графіків, інфограм чи теплових карт.

Також, для перевірки вподобань користувачів використовується метод A/B тестувань, що передбачає розбиття клієнтів на групи, та демонстрація їм різних варіацій подібних елементів ресурсу [8]. Таким чином, отримується інформація про

ефективність різного виду дизайну, положення елементів чи інших видимих характеристик.

### 1.2.2 Методи користувацького досвіду (UX)

Покращений інтерфейс взаємодії користувача з сервісом допомагає покращити рейтинг конверсій ресурсу. Одним з методів покращення сайту є оптимізація швидкості завантаження сторінок, що допомагає збільшити утримування користувача на ресурсі та, відповідно, збільшити увагу на бажаних елементах [9]. Також використовується чіткий виклик до дій (CTA), що використовує психологічні аспекти принципів роботи підсвідомих характеристик людського мозку для покращення бажаного рейтингу конверсій [10].

### 1.2.3 Методи наповнення контентом

Добавлення товару чи послуги є невід’ємним фактором, що впливає на вибір продукту користувачем. Кожен елемент, конверсію якого необхідно покращити, повинен бути описаним та оформленим відповідно правилам копірайтингу, а весь візуальний контент – оптимізовано. Дані дії відносяться до сфери SEO, що оптимізує саму сторінку та її комунікацію з користувачем та пошуковими сервісами. Для коректного наповнення контентом необхідно знати потенційну аудиторію, включення ключових слів у текстові поля, виділення унікальних характеристик, надання пріоритету перевагам, включення професійного візуального наповнення та оптимізація ресурсу для користувачів різних девайсів [11].

Комбінація перелічених методів CRO дозволяє ефективно збільшувати рейтинг конверсій, однак, кожна з методик потребує ретельного підходу та навичок аналізу, що ускладнює швидкість та ефективність прийняття змін, що може бути вирішено застосуванням методів машинного навчання, у поєднанні з класичними методами оптимізації рейтингу конверсій.

## 1.3 Огляд рекомендаційних систем

Рекомендаційні системи є ключовим інструментом у сфері маркетингу, медіа, соціальних мереж та інших сферах, де необхідно забезпечити

персоналізоване обслуговування для користувачів. Ці системи допомагають підвищити рівень конверсій, збільшити залученість користувачів та покращити загальний досвід користувача.

Рекомендаційні системи використовуються для аналізу поведінки користувачів та рекомендацій відповідних продуктів або послуг. Це дозволяє підвищити конверсії шляхом точної та персоналізованої рекомендації, що відповідає інтересам конкретного користувача. Наприклад, на основі історії покупок або перегляду веб-сторінок, рекомендаційна система може пропонувати подібні товари або контент, що збільшує ймовірність покупки або взаємодії.

Рекомендаційні системи можуть бути побудовані на основі різних характеристик та подій, що відбуваються в системі за участі користувача. Основними типами рекомендаційних систем є: системи на основі змісту, системи на основі співпадінь та гібридні системи (рис. 1.1).



Рисунок 1.1 – Схематичне зображення видів рекомендаційних систем [12]

Системи на основі змісту (content-based) – це системи, що аналізують характеристики або властивості продуктів або контенту, а потім рекомендують схожий контент. Наприклад, якщо користувач переглядає фільми певного жанру, система може рекомендувати інші фільми цього ж жанру.

Системи на основі співпадінь (collaborative filtering) – ці системи аналізують поведінку користувачів та їх взаємодію з продуктами для створення рекомендацій. Це може бути здійснено на основі історії покупок, оцінок або перегляду.

Гібридні системи комбінують інші рекомендаційні методи або моделі для надання більш точних та персоналізованих рекомендацій.

Рекомендаційні системи існують для полегшення пошуку та навігації для користувачів, підвищення рейтингу конверсій та залученості користувачів. Вони допомагають покращити досвід користувача шляхом забезпечення персоналізованих пропозицій, що відповідають конкретним потребам та інтересам користувачів.

Машинне навчання використовується у рекомендаційних системах для аналізу даних та побудови моделей, які прогнозують інтереси користувачів та рекомендують відповідний контент. Популярні методи включають алгоритми колаборативного фільтрування, які використовують матричні рекомендації та алгоритми на основі глибокого навчання, що дозволяють автоматично витягувати характеристики з даних.

Системи рекомендацій є важливим інструментом для підвищення рейтингу конверсій та поліпшення досвіду користувача в різних галузях, використовуючи різноманітні методи машинного навчання для аналізу даних та рекомендацій.

## РОЗДІЛ 2

### ПРАКТИЧНЕ ЗАСТОСУВАННЯ МАШИННОГО НАВЧАННЯ В CRO-СИСТЕМАХ

#### 2.1 Методи машинного навчання для реалізації CRO-системи

Методи машинного навчання є статистичними алгоритмами, здатними навчатись на даних і узагальнюватися на нових даних. Алгоритми в машинному навчанні відрізняються не лише на основі принципу роботи, однак і на основі вхідних даних та отриманого результату. Вибір методів машинного навчання для реалізації CRO-системи є важливим етапом у процесі створення ефективної та функціональної системи. При виборі методів враховуються такі фактори, як: обсяг даних, тип даних, складність завдання CRO, часові обмеження та ресурси. Одним з найпоширеніших підходів є використання нейронних мереж, оскільки вони можуть ефективно працювати з великим обсягом даних та виявляти складні зв'язки між параметрами.

Додатково, для реалізації CRO-системи використовуються методи класифікації, регресії та кластеризації. Класифікаційні моделі можуть допомогти в ідентифікації класів аномальної поведінки, або в розпізнаванні певних шаблонів. Регресійні моделі здатні прогнозувати майбутні значення параметрів на основі даних з навчальної вибірки. Кластерний аналіз може бути використаним для групування об'єктів або подій, що мають взаємозв'язок у CRO-системі.

##### 2.1.1 Нейронні мережі

Нейронна мережа – це програма або модель машинного навчання, яка приймає рішення подібно до людського мозку, використовуючи процеси, що імітують спільну роботу біологічних нейронів, щоб ідентифікувати явища, зважувати варіанти та робити висновки [13]. Кожна нейронна мережа складається з шарів вузлів або штучних нейронів – вхідного рівня, одного або кількох прихованих шарів і вихідного рівня [14]. Кожен вузол підключається до інших і має власну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі.

Нейронні мережі можна класифікувати на різні типи, які використовуються для різних цілей. Найпоширеніші типи нейронних мереж, що зустрічають при вирішенні типових задач:

- Перцептрон – це найстаріша нейронна мережа, створена Френком Розенблатом у 1958 році.

- Нейронні мережі прямого поширення, або багат шарові перцептрони (MLP) – це нейронні мережі, що складаються з вхідного, прихованого шару або шарів і вихідного шару. Хоча, дані мережі можуть бути названими MLP, вони можуть складатись з сигмовидних нейронів, а не перцептронів, у зв'язку з тим, що більшість проблем реального світу є нелінійними [14]. Такі моделі складають основу для комп'ютерного зору, обробки природньої мови та інших нейронних мереж.

- Згорткові нейронні мережі (CNN) – подібні з нейронними мережами прямого поширення, однак більш узагальнені до виконання багатовимірних обчислень, таких як зображення та відео [15].

- Рекурентні нейронні мережі (RNN) – характеризуються своїми зворотними зв'язками [16]. Ці алгоритми навчання в основному використовуються при роботі з часовими рядами для прогнозування майбутніх результатів, таких як прогнозування ринку акцій або прогнозування продажів.

Глибоке навчання та нейронні мережі часто використовують взаємозамінно. Нейронна мережа, яка складається з більше ніж трьох шарів, які включають вхідні дані та вихідні, може вважатися алгоритмом глибокого навчання. Нейронна мережа, яка має лише два або три шари, є базовою нейронною мережею.

### 2.1.2 Класифікаційні моделі

Класифікація – це контрольований метод машинного навчання, коли модель намагається передбачити правильну мітку заданих вхідних даних [17]. Під час класифікації модель повністю навчається з використанням навчальних даних, а потім її оцінюють на тестових даних перед використанням для виконання прогнозу на нових, невідомих, даних. Класичні методи машинного навчання були успішно застосовані в області багатьох завдань класифікації, такими методами є [18]:

– Модель змішаних гауссових розподілів (GMM) – це ймовірна модель, яка припускає, що всі точки даних генеруються з суміші скінченної кількості гауссових розподілів з невідомими параметрами [19].

– Метод опорних векторів (SVM) – це керований алгоритм машинного навчання, що класифікує дані, знаходячи оптимальну лінію або гіперплощину, яка максимілізує відстань між кожним класом в N-вимірному просторі [20]. Даний алгоритм працює найкраще на малих та комплексних наборах даних.

– Випадковий ліс (Random Forest) – це мета-оцінювач, який встановлює декілька класифікаторів дерев рішень на різних підвибірках набору даних і використовує усереднення для покращення точності прогнозування та контролю за перенавчанням [21].

– Наївний Баєсів класифікатор (Naive Bayes classifier) – генеративний навчальний алгоритм, мета якого – моделювати розподіл вхідних даних в межах конкретного класу або категорії. Широко використовується у класифікації тексту, фільтрації спаму та рекомендаційних системах [22].

### 2.1.3 Регресійні моделі

Одним з найбільш поширених та всеосяжних статистичних алгоритмів машинного навчання є лінійна регресія. Лінійна регресія використовується для виявлення лінійної залежності між одним або декількома предикатами. Лінійна регресія має два типи: проста регресія та множинна регресія (MLR) [23].

Проста лінійна регресія – розглядається у двовимірному просторі вибірки, що утворений однією незалежною та однією незалежною змінною. Множинна лінійна регресія – це узагальнення лінійної регресії з урахуванням декількох незалежних змінних.

### 2.1.4 Кластерний аналіз

Кластеризація – це процес групування об'єктів на основі їх схожості. Існує багато різних алгоритмів кластеризації, оскільки кластери важко визначити і немає встановленого критерію для хорошої кластеризації. Кожен алгоритм може давати різні кластери в залежності від обраної моделі кластеру [24]. У машинному навчанні кластеризація використовується як перший крок у розумінні набору

даних. Коли приклади не промарковані (не мають мітки) – це неконтрольована кластеризація, тоді як у випадку маркованих даних (з мітками) – це класифікація.

Існують різні моделі кластерів, що відрізняються типом поведінки з даними. Вони включають в себе моделі зв'язності, центроїдні моделі, моделі розподілу, моделі густини, групові моделі, графові моделі та нейронні моделі. Кластеризація поділяється на жорстку та м'яку. Жорстка кластеризація повністю призначає точки даних одному кластеру, тоді як м'яка кластеризація призначає ймовірнісний бал для належності точок даних до різних кластерів.

## **2.2 Огляд алгоритмів CRO-системи**

Пропонована CRO-система імплементується у веб-сайт онлайн-магазину, що характеризує дані, які необхідно обробляти. Онлайн-магазин містить інформацію про користувача, товар, історію покупок та оцінок користувачів. Особливою характеристикою у даному випадку є динамічність даних та їх тип.

Інформація про користувача є структурованою самим користувачем, товар охарактеризований завчасно платформою, а оформлення послуги та її оцінка користувачем зазначається динамічно у реальному часі. Відповідно даним характеристикам були зроблені наступні висновки: аналіз динамічних даних доречно проводити швидкими алгоритмами з рекомендаційним результатом.

Аналіз оцінок користувачів та куплених (переглянутих) товарів проводиться на основі глибокої нейронної мережі та лінійної регресії. Оцінки користувача, будучи динамічним показником, зберігаються довго та мають свій вплив на майбутні дії користувача, у зв'язку з чим було прийнято рішення навчання на наборі даних оцінок на товари від користувача глибокою нейронною мережею на основі колаборативної фільтрації, а її динамічне використання обумовлене наявністю збереженого результату навчання мережі. Лінійна регресія використовується для сортування товарів для користувача зважаючи на його попередні покупки та перегляд товарів без оформлення покупки. Даний тип сортування товарів дозволяє коректніше їх відображати зважаючи на історію цінових переглядів користувача, а не лише сортування від малих чи великих цін.

Для аналізу товарів був обраний алгоритм кластеризації K-Means на основі аналізу описів товарів. Дана методика дозволяє здійснювати пошук потенційних зацікавлень користувача на основі його текстового запиту шляхом поділу товарів на кластери по їх опису.

### **2.3 Аналіз обраних методів у CRO-системах**

Глибока нейронна мережа потребує довгої фази навчання для детального аналізу виставлених оцінок користувачів та пошуку закономірностей. Пошук закономірностей заснований на алгоритму колаборативної фільтрації, мета якого пошук користувачів, що мають спільні оцінки на певні товари, що дозволяє зробити висновки по вподобанню конкретного користувача на основі оцінок подібних користувачів. Дана методика має низку недоліків, а саме: потреба у довгому навчанні, потребу у повному перенавчанні у разі виникнення помилки та потребі у перенавчанні при змінні вхідних даних. Однак, позитивними сторонами є: детальний аналіз та ретельний пошук закономірностей, передбачення можливої оцінки користувача відразу на всі товари, зазначені у вибірці та найдійність обчислень.

Лінійна регресія для сортування товарів є простою та невибагливою у контексті обчислювальних ресурсів, дозволяє враховувати історію покупок та переглядів користувача та підвищувати рейтинг конверсії на сайті у зв'язку з підняттям релевантних по ціні товарів у пошуку. Недоліком даного методу є обмежена точність порівняно з більш складними моделями, у зв'язку з чим, застосування даного алгоритму несе виключно рекомендаційний характер та не вносить зміни у вибірку рекомендованих товарів, а лише вказує бажаний порядок їх відображення користувачу.

Кластеризація на основі алгоритму K-means дозволяє покращити пошук товарів на основі текстового запиту. Назва товару може відрізнитись від побутового вжитку, або бути важкою для запам'ятовування, що може бути зазначено в описі товару, що містить більше конкретних та важливих характеристик. Аналіз опису товару та розподіл товарів на кластери по текстовому опису дозволяє зменшити

потенційну вибірку товарів для користувача та ефективно їх згрупувати для персоналізованої рекомендації. Недоліком даної системи є залежність від якості та об'єму текстових даних, можливість збоїв при надмірних чи недостатніх даних та потребу у ретельному проведенні етапу нормалізації даних.

## РОЗДІЛ 3

### РОЗРОБКА ТА ВПРОВАДЖЕННЯ СRO-СИСТЕМИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

#### 3.1 Реалізація та наповнення веб-сайту інформацією

Система підвищення коефіцієнту конверсії передбачає взаємодію з інформацією веб-сайту онлайн-магазину для реалізації систем рекомендацій та аналітики системи. Насичення веб-сайту було взяте з відкритих датасетів з різнотипних магазинів [25-26].

Пропонований веб-сайт насичений такими даними:

– продукт – елемент магазину з вказаними параметрами назви, ціни, ідентифікатора, опису, зображень. Вигляд окремої сторінки продукту продемонстровано на рисунку 3.1;

– користувач – репрезентація покупця, що здійснює покупки чи залишає оцінки для продуктів. Може взаємодіяти з веб-сайтом створюючи події: покупка продукту, встановлення оцінки продукту. Вигляд окремої сторінки користувача продемонстровано на рисунку 3.2;

– оцінка товару – окремо виділений набір даних для навчання глибокої нейронної мережі на основі алгоритму колаборативної фільтрації, оцінки були нормалізовані та передбачені у датасеті в значеннях від 1 до 5.

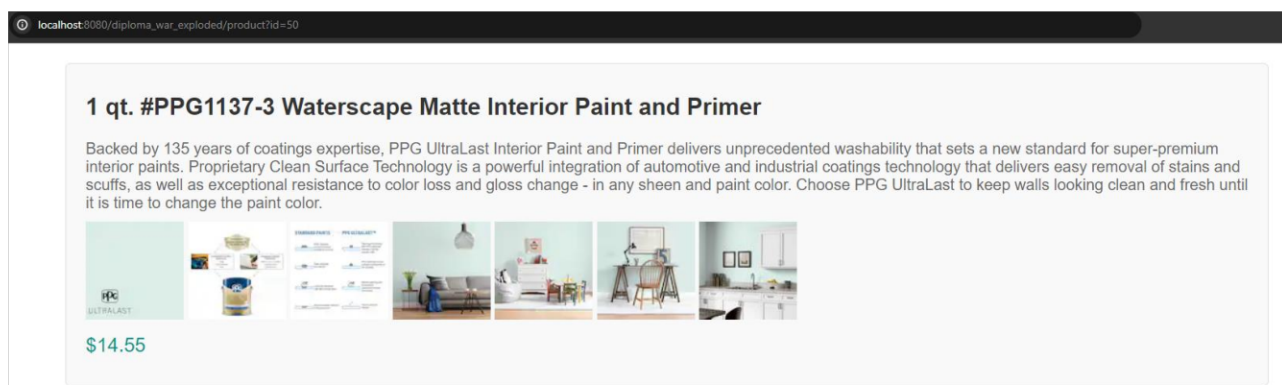


Рисунок 3.1 – Сторінка продукту

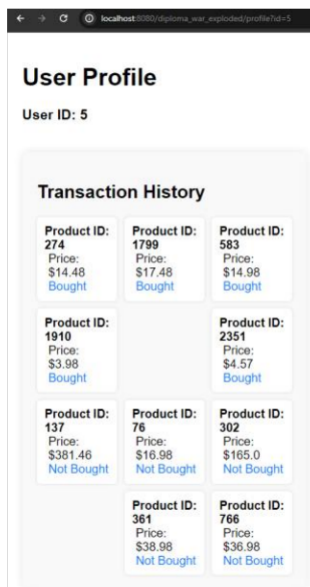


Рисунок 3.2 – Сторінка профілю користувача

Для наглядності були добавлені результати роботи алгоритмів машинного навчання окремими текстовими полями (рис. 3.3).

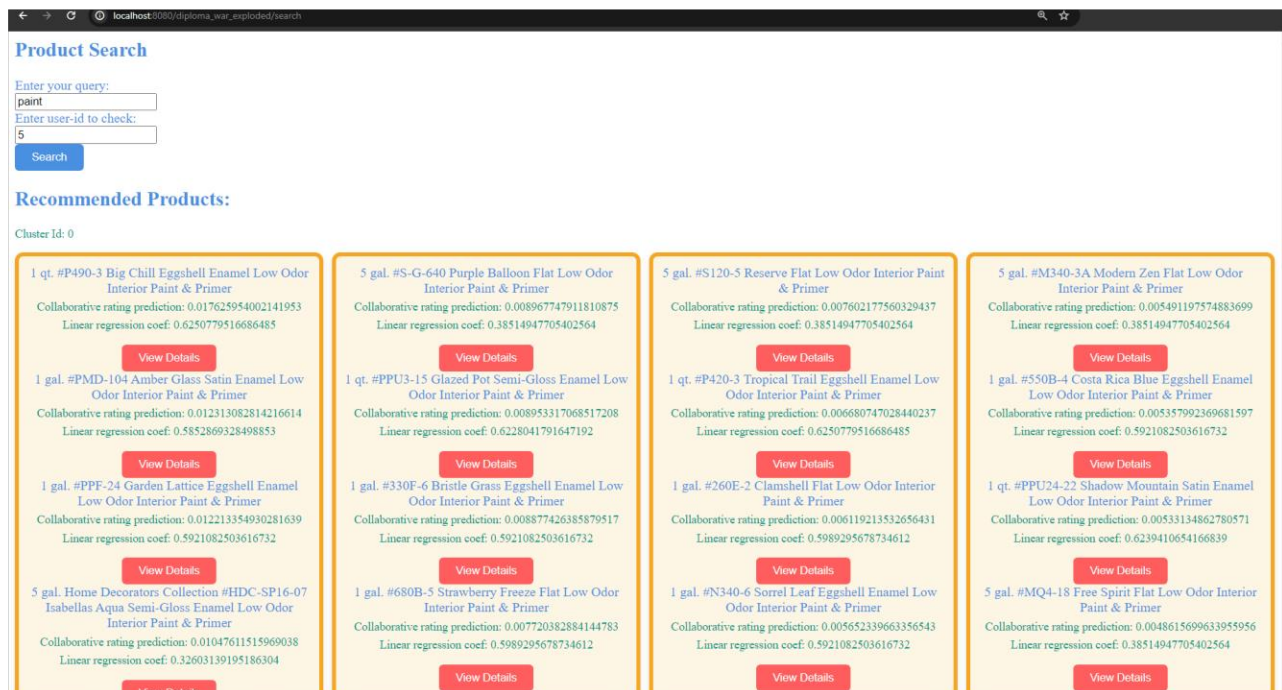


Рисунок 3.3 – Приклад сторінки з коефіцієнтами, що обчислені застосованими алгоритмами машинного навчання

Визначення рекомендацій у даній гібридній системі рекомендацій здійснюється у декілька кроків, що продемонстровані на рисунку 3.4.

Рекомендований список продуктів визначається шляхом обмеження вибірки та впорядкування її в залежності від результатів, отриманих використаними алгоритмами машинного навчання.

На першому кроці формується підвибірка продуктів, що відповідають текстовому запиту користувача. Визначається кластер пошукового запиту та отримуються ті продукти, які були визначені в даному кластері.

На другому кроці визначається рейтинг продуктів для користувача алгоритмом колаборативної фільтрації на основі глибокої нейронної мережі. Завдяки тому, що нейронна мережа була навчена завчасно, обрана вибірка швидко впорядковується по обчисленому значенню оцінки.

На третьому кроці визначається відповідність ціни товару зацікавленості користувача на основі його попередньої історії покупок та тих товарів, що користувач переглянув, однак не здійснив покупку (рис. 3.4).



Рисунок 3.4 – Покроковий алгоритм формування рекомендацій

Реалізація веб-сайту здійснена засобами Java Servlet API [27] з застосуванням бібліотеки Jakarta Servlet API 5.0.0 [28] використовуючи веб-сервер Apache Tomcat [29].

Технологія Java Servlet API – це основний набір функцій та інструкцій для створення веб-додатків. Вона дозволяє створювати динамічний веб-контент, обробляючи HTTP-запити. Jakarta Servlet API – це сучасна версія Java Servlet API, що надає додаткові інструменти взаємодії серверу з користувачем та впроваджує певні полегшення синтаксису розробки програмного забезпечення.

Apache Tomcat – це відкритий сервер застосунків, який використовується для виконання Java сервлетів і JavaServer Pages (JSP). Даний сервер реалізує середовище для запуску веб-додатків. Окрім можливості створення динамічних сторінок з виконанням програмного коду використаної мови програмування даний серверний застосунок забезпечує надійне та безпечне з'єднання з користувачем, збереженням інформації про підключення користувача. Здатний самостійно визначати наявність сесії, продовжувати її чи створювати в разі потреби, що значно покращує досвід користування сервісом. Реалізує форматування відображення сторінки у полі введення адреси сайту в браузері користувача.

Формування списку рекомендацій продуктів вимагає два вхідних параметри: текстовий запит та ідентифікатор користувача, від імені якого виконується запит. Відповідно описаному алгоритму формування рекомендацій була реалізована функція генерування впорядкованого списку з порядком продуктів та обчислених значень, що використовуються для визначення порядку рекомендованого товару. Функція формування списку рекомендацій продемонстрована на лістингу 3.1.

### Лістинг 3.1 – Функція формування списку рекомендацій

```

private Recommendations fetchRecommendedProducts(String query, int userid)
throws IOException {
    ClusteringKMeans.ClusterRecommendation clusteringRecommendations =
ClusteringKMeans.getRecommendations(query);
    CollaborativeFiltering.CollaborativeRecommendation
collaborativeRecommendation = CollaborativeFiltering.getRecommendations(userid);

    List<ProductObject> recommendedProducts = new ArrayList<>();
    List<Integer> clu = clusteringRecommendations.getRecProducts();
    List<CollaborativeFiltering.ProductRating> ratings =
collaborativeRecommendation.getProductRatings();
    List<Integer> newSorted = new ArrayList<>();
    for (CollaborativeFiltering.ProductRating r : ratings) {
        if (clu.contains(r.getProductId())) {
            newSorted.add(r.getProductId());
        }
    }

    newSorted.forEach(i -> {
        try {
recommendedProducts.add(ProductServlet.getRecommendations(String.valueOf(i)));
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    });
    return new Recommendations(userid,
String.valueOf(clusteringRecommendations.getClusterId()), recommendedProducts,
ratings);
}

```

#### Кінець лістингу 3.1

Відображення елементів у вигляді веб-сторінки реалізовано засобами HTML, CSS та JSP (Java Servlet Pages). Елементи HTML та CSS використовуються для реалізації статичної веб-сторінки та встановлення розмітки з нанесенням візуальних стилів. JSP використовується для виконання програмного коду java на сторінці та взаємодії сторінки з сервером. Завдяки використанні даної технології сторінка набуває динамічного типу та здатна проводити обчислення в реальному часі.

У даному випадку була реалізована сторінка «search», структура якої описана у файлі «search.jsp». Вміст даного файлу зображений на лістингу 3.2. Дана сторінка складається з двох основних елементів: поля для введення вхідних даних та місця для рекомендованих товарів. Після введення запиту та ідентифікатора користувача, від імені якого виконується запит, відбувається запит до серверу для отримання списків рекомендацій. Останній етап формування рекомендацій, а саме

використання лінійної регресії, реалізоване при виклику самої сторінки, адже її застосування немає негативного впливу на час виконання операцій в даному випадку.

### Лістинг 3.2 – Вміст файлу search.jsp

```

<body>
<h2>Product Search</h2>
<form id="searchForm" action="search" method="post">
  <label for="query" class="label">Enter your query:</label><br>
  <input type="text" id="query" name="query"><br>
  <label for="userid" class="label">Enter user-id to check:</label><br>
  <input type="number" id="userid" name="userid"><br>
  <button type="submit" class="search-button">Search</button>
</form>
<% if (request.getAttribute("recommendedProducts") != null) { %>
<% SearchServlet.Recommendations recommendedProducts =
(SearchServlet.Recommendations) request.getAttribute("recommendedProducts");%>

<span><h2 class="label">Recommended Products:</h2><p class="form-label">Cluster
Id: <%=recommendedProducts.getClusterId()%></p></span>
<div class="product-grid">
  <%
    LinearRegression lr = new
LinearRegression(UserActivityGenerator.linearRegressionDataLoad(recommendedProdu
cts.getUserId()));
  %>

  <% if (recommendedProducts == null) { %>
  <p class="no-products">No recommended products available.</p>
  <% } else {
    int count = 0;
    for (ProductObject product : recommendedProducts.getObjects()) {
      if (count % 4 == 0) { %>
        <div class="product-item">
          <% } %>
          <div class="product-name"><%= product.getName() %></div>
          <p class="collaborative-rating">Collaborative rating prediction:
<%=recommendedProducts.getColRatings().stream().filter(pr ->
String.valueOf(pr.getProductId()).equals(product.getId())).findAny().get().getRa
ting()%></p>
          <p class="linear-regression-coef">Linear regression coef:
<%=lr.getPreferenceCoefficient(product.getPrice())%></p>
          <a href="product?id=<%= product.getId() %>"><button>View
Details</button></a>
          <br>
          <% if ((count + 1) % 4 == 0 || count ==
recommendedProducts.getObjects().size() - 1) { %>
            </div>
          <% }
          count++;
        }
      }
    }%>
</div>
</body>

```

Кінець лістингу 3.2

### 3.2 Впровадження та використання CRO-системи до веб-сайту

CRO-система була впроваджена модифікацією сайту та додаванням необхідних ресурсів в кореневу систему сайту. На відміну від стандартної процедури завантаження HTML-сторінок, використання алгоритмів машинного навчання не можуть забезпечити достатньо швидкий показник завантаження сторінки у випадку проведення всіх етапів навчання при взаємодії користувача з веб-сторінкою. Для вирішення даної проблеми певні етапи навчання, що можуть бути проведеними раніше, були проведені до увімкнення веб-сервісу, а саме – навчання нейронної мережі для колаборативної фільтрації та кластеризація текстів.

Реалізованими алгоритмами машинного навчання для підвищення рейтингу конверсій є:

- колаборативна фільтрація на основі глибокої нейронної мережі;
- кластеризація алгоритмом K-means;
- лінійна регресія.

Перераховані методи використовуються для реалізації рекомендаційної системи на веб-сайті.

Колаборативна фільтрація – це алгоритм, метою якого є пошук взаємозв'язків між користувачами та продуктами, що були оцінені іншими користувачами [30]. Отже, користувачу будуть пропонуватись продукти, що були високо оцінені іншими користувачами, що мають спільні оцінки на попередні покупки, що дозволяє вказати на наявність між ними спільних інтересів до товарів.

Реалізація колаборативної фільтрації на основі глибокої нейронної мережі була здійснена мовою програмування Python використовуючи бібліотеки pandas, keras та sklearn. Побудова даної функції веб-сайту була здійснена в наступних етапах:

- Підготовка даних. Дані були нормалізовані та перевірені на коректність.
- Побудова архітектури мережі. Для реалізації алгоритму колаборативної фільтрації була обрана глибока нейронна мережа прямого

поширення, схематичне відображення архітектури цієї мережі зображено на рисунку 3.5.

– Навчання мережі. Алгоритм колаборативної фільтрації передбачає встановлення взаємозв'язків між подібними оцінками продуктів різними користувачами, що надає можливість зробити висновок у наявності взаємозв'язку між оцінками даних користувачів. Для досягнення поставленої цілі мережі з даною конфігурацією даних були обрані гіперпараметри, що описані у таблиці 3.1. Для запобігання виникнення проблеми перенавчання була реалізована умова для зупинки проведення навчання (лістинг 3.3), що зупиняє процес навчання у випадку, якщо обчислене значення втрат не змінюється протягом трьох епох.

– Використання мережі. Для створення рекомендацій товарів користувачу була реалізована функція «recommend\_product\_for\_user» (лістинг 3.4), що генерує предбачувані оцінки заданого користувача на всі продукти та сортує їх по значенню оцінки. Використання даного python-скрипту на веб-сайті реалізоване запуском у java-програмі (лістинг 3.5).

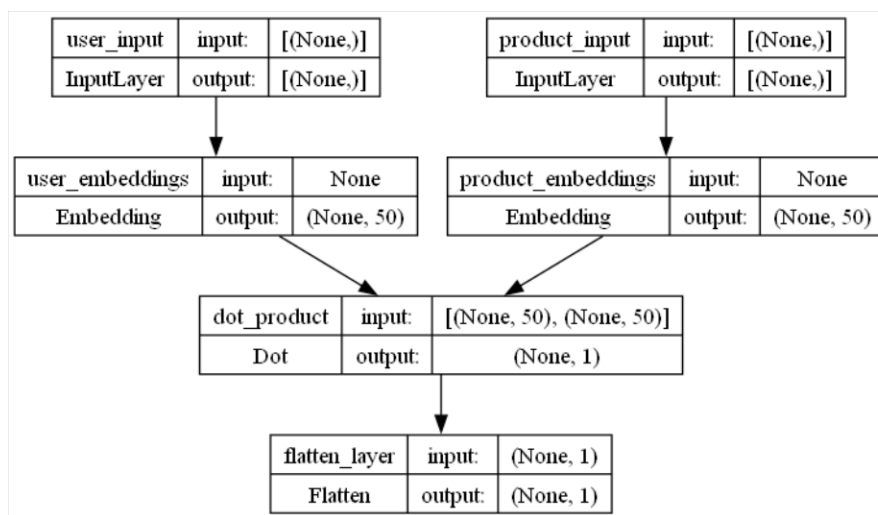


Рисунок 3.5 – Схема архітектури нейронної мережі створена функцією plot\_model з бібліотеки keras

Таблиця 3.1 – Гіперпараметри при навчанні нейронної мережі

Гіперпараметр	Опис
Кількість прихованих шарів	2
Кількість епох	10
Валідаційна вибірка	0.1 (10%)
Оптимізатор	Adam
Розмір міні-вибірки	128
Функція визначення втрат	Середньоквадратичне відхилення

Лістинг 3.3 – Встановлення умови зупинки навчання

```
early_stopping = EarlyStopping(monitor='val_loss', patience=3,
restore best weights=True)
```

Кінець лістингу 3.3

Лістинг 3.4 – Програмний код функції «recommend\_product\_for\_user»

```
# Функція для рекомендації продуктів для конкретного користувача
def recommend_product_for_user(user_id):
    user_ids = [user_id] * len(allProducts)
    data = pd.DataFrame({'UserId': user_ids, 'ProductId': allProducts})

    data = data[['UserId', 'ProductId']]

    # Передбачення рейтингу для користувача та всіх товарів
    predicted_ratings = model.predict([data['UserId'], data['ProductId']])

    # Фрейм з передбаченими оцінками
    all_products_per_user = pd.DataFrame({
        'UserId': user_ids,
        'ProductId': allProducts,
        'Rating': predicted_ratings.flatten()
    })

    all_products_per_user_sorted =
all_products_per_user.sort_values(by='Rating', ascending=False)
    return all_products_per_user_sorted
```

Кінець лістингу 3.4

## Лістинг 3.5 – Виконання python-скрипту на веб-сайті в java-програмі

```

public static CollaborativeRecommendation getRecommendations(int userId) throws
IOException {
    ProcessBuilder pb = new ProcessBuilder(PYTHON_COMMAND, PYTHON_SCRIPT_PATH,
Integer.toString(userId));
    Process p = pb.start();

    BufferedReader in = new BufferedReader(new
InputStreamReader((p.getInputStream())));
    String s;
    List<ProductRating> pr = new ArrayList<>();
    while ((s = in.readLine()) != null) {
        try {
            System.out.println(s);
            pr.add(new ProductRating(Integer.parseInt(s.split(" ")[0]),
Double.parseDouble(s.split(" ")[1])));
        } catch (NumberFormatException ignored) {}
    }
    return new CollaborativeRecommendation(userId, pr);
}

```

## Кінець лістингу 3.5

У результаті, перед запуском веб-сайту було проведено навчання мережі та збереження навченої моделі у файлі. При зверненні веб-сайту до програми повторного навчання не відбувається, замість цього відбувається підвантаження збереженої моделі, що значно оптимізує час затримки генерування списку рекомендацій продуктів. Однак, недоліком даного підходу є потреба у перетреновуванні мережі у зв'язку з появою нових користувачів, відсутності виставлених оцінок або їх зміни.

Кластеризаційний алгоритм використовується для створення пошукової системи, а саме, визначення того, до якого кластеру (групи товарів) відносити пошуковий запит користувача [31]. Даний метод оптимізує пошук продуктів не лише по назві, але і по їх опису, що може вмещувати у собі більш детальну інформацію, яку може обрати для пошукового запиту користувач.

Для вирішення завдання кластеризації використовується алгоритм K-means (k-середніх), що реалізований мовою програмування Python використовуючи бібліотеки pandas та sklearn. Алгоритм k-середніх – це алгоритм для впорядкування множин об'єктів у однорідні групи завдяки обраному значенню кількості центроїдів [32].

Реалізований алгоритм навчається на описах товарів, тобто на великому наборі текстових даних. Для цього текстові дані перетворені у матрицю ознак TF-IDF (Term Frequency-Inverse Document Frequency – це статистичний метод для оцінки важливості слова в документі в контексті набору документів) використовуючи `TfidfVectorizer` (лістинг 3.6). Дана матриця ознак використовується для подальшої кластеризації алгоритмом K-Means (лістинг 3.7).

### Лістинг 3.6 – Векторизація опису товарів

---

```
vectorizer = TfidfVectorizer(stop_words='english')
X = vectorizer.fit_transform(product_descriptions["description"])
```

---

Кінець лістингу 3.6

### Лістинг 3.7 – Застосування моделі K-Means для генерації термів

---

```
model = KMeans(n_clusters=true_k,
               init='k-means++',
               max_iter=1000,
               n_init=1).fit(X)
order_centroids = model.cluster_centers_.argsort()[:, :-1]
terms = vectorizer.get_feature_names_out()
```

---

Кінець лістингу 3.7

У зв'язку з тим, що алгоритм K-means є алгоритмом машинного навчання без підкріплення («без вчителя») дані для його навчання не мали ідентифікуючих міток, що є причиною унеможливлення автоматичного підбору кількості кластерів. У зв'язку з набором даних та ручного тестування було прийнято рішення розподілу даних на 15 кластерів. Дані кластери складаються з «термів», тобто слів, які характеризуються даним кластером, що було визначено алгоритмом. Перші терми кожного кластеру продемонстровані на рисунку 3.6.

Використання даного алгоритму надає можливість взаємодіяти з текстовими запитами користувача. Користувач вводить текстовий пошуковий запит, що також проходить процес векторизації, що дозволяє визначити належність запиту до кластеру. Визначення кластеру, до якого відноситься запит користувача, зменшує вибірку рекомендованих продуктів.

```

Кластер 0: year, interior, marquee, delivering, fresh, performance, stay, color, guarantee, coverage
Кластер 1: fading, moisture, exterior, stains, provides, appearance, formulated, displays, enamel, withstand
Кластер 2: ppg, timeless, resistance, paint, diamond, coat, coverage, provides, complete, color
Кластер 3: design, use, light, room, collection, easy, perfect, home, space, modern
Кластер 4: glidden, essentials, results, value, premium, paint, voc, years, exceptional, coating
Кластер 5: creates, windows, outdoor, furniture, great, enamel, doors, exterior, trim, plus
Кластер 6: dirt, exterior, doors, railings, wrought, home, like, iron, freshly, behr
Кластер 7: ultra, exterior, paint, excellent, peace, minutes, little, behr, mind, 60
Кластер 8: scuff, defense, ultra, paint, durability, primer, behr, settle, breakthrough, desire
Кластер 9: sample, rugs, ft, covers, oz, sq, 16, idea, nuloom, lets
Кластер 10: plus, premium, independent, labs, ranked, 3rd, party, requirements, benefits, meets
Кластер 11: exterior, stucco, brick, look, matte, masonry, flat, siding, minimizes, traditional
Кластер 12: looks, great, trim, guard, sunlight, subtle, exterior, damage, pearl, helps
Кластер 13: lustre, cultivated, gorgeous, exterior, home, minor, bring, freshly, best, behr
Кластер 14: latex, paint, zero, application, acrylic, low, hide, interior, 35, applied

Process finished with exit code 0

```

Рисунок 3.6 – Перші терми кожного з кластерів

Лінійна регресія використовується для встановлення взаємозв'язків між простими значеннями на основі пошуку лінійного зв'язку між даними. На представленому веб-ресурсі лінійна регресія використовується для прогнозування порядку рекомендованих продуктів в залежності від цінової категорії попередніх обраних товарів. Дана методика реалізована для підбирання більш привабливого товару користувача на основі його ціни.

Прогнозування бажаної ціни пропонованих товарів реалізоване мовою програмування Java використовуючи бібліотеку Apache Commons [33]. Для прогнозування ціни використовується історія покупок користувача та відвідин, що не співпадають з покупками.

Набір даних для регресійного аналізу в даному випадку складається з матриці, де кожному значенню ціни відповідає булеве значення. Дане булеве значення відповідає тому, чи здійснив користувач дану покупку чи ні, що дозволяє визначити залежність ціни від покупки у випадку перегляду продукту. Програмний код реалізації лінійної регресії продемонстрований у листингу 3.8.

### Лістинг 3.8 – Реалізація лінійної регресії

```
package com.gmail.artemkonotopchik2003.diploma.ml;

import org.apache.commons.math3.stat.regression.SimpleRegression;

public class LinearRegression {

    private double[][] data;

    public LinearRegression(double[][] data) {
        this.data = data;
    }

    public double getPreferenceCoefficient(double newPrice) {
        SimpleRegression regression = new SimpleRegression();

        for (double[] point : this.data) {
            regression.addData(point[0], point[1]);
        }

        double slope = regression.getSlope();
        double intercept = regression.getIntercept();

        return slope * newPrice + intercept;
    }
}
```

#### Кінець лістингу 3.8

Результатом використання даного коду є визначення того, чи пропонується ціна є більш відповідною для користувача зважаючи на його історію покупок та переглядів, що сортує товари в залежності від коефіцієнту зацікавленості ціни. Важливим є використання даного методу (лінійної регресії) виключно з попередніми рекомендаційними методами. Продемонстрований варіант використання лінійної регресії чутливий до шумів та може спотворювати результати, якщо попередньо не обмежити вибірку пропонованих продуктів. Також, важливою відмінністю від стандартного цінового фільтру є те, що користувачу не будуть пропонуватись лише дорогі чи лише дешеві продукти, а саме продукти, цінова категорія яких відповідає цінам попередньо зроблених ним покупок.

Приклад товарів у пошуку користувача з зазначеними коефіцієнтами машинного навчання продемонстрований на рисунку 3.7.

**Product Search**

Enter your query:

Enter user-id to check:

[Search](#)

**Recommended Products:**

Cluster Id: 8

<p>1 gal. #HDGWN58U River Birch Beige Semi-Gloss Interior Paint with Primer</p> <p>Collaborative rating prediction: 0.017354052513837814</p> <p>Linear regression coef: 0.5761551767216737</p> <p><a href="#">View Details</a></p>	<p>5 gal. #HDGWN09U Heatherbelle Semi-Gloss Interior Paint</p> <p>Collaborative rating prediction: 0.010865186341106892</p> <p>Linear regression coef: 0.5124261352074979</p> <p><a href="#">View Details</a></p>	<p>1 gal. #HDGY13U Canvasback Brown Flat Interior Paint with Primer</p> <p>Collaborative rating prediction: 0.00942001212388277</p> <p>Linear regression coef: 0.5812946155534621</p> <p><a href="#">View Details</a></p>	<p>1 qt. PPG1158-5 Cosmopolitan Eggshell Interior Latex Paint</p> <p>Collaborative rating prediction: 0.005109677091240883</p> <p>Linear regression coef: 0.5895177176843234</p> <p><a href="#">View Details</a></p>
<p>1 gal. PPG1127-6 Winning Ticket Eggshell Interior Paint</p> <p>Collaborative rating prediction: 0.015350720845162868</p> <p>Linear regression coef: 0.5833503910861774</p> <p><a href="#">View Details</a></p>	<p>1 gal. PPG1202-5 Buffalo Trail Eggshell Interior Latex Paint</p> <p>Collaborative rating prediction: 0.009790061041712761</p> <p>Linear regression coef: 0.5792388400207467</p> <p><a href="#">View Details</a></p>	<p>1 gal. #HDGCN26D Tundra Pond Satin Interior Paint with Primer</p> <p>Collaborative rating prediction: 0.008332662284374237</p> <p>Linear regression coef: 0.5771830644880314</p> <p><a href="#">View Details</a></p>	<p>5 gal. #HDGB39D Mallard Trek Turquoise Satin Interior Paint with Primer</p> <p>Collaborative rating prediction: 0.004657480865716934</p> <p>Linear regression coef: 0.4959593731904479</p> <p><a href="#">View Details</a></p>
<p>1 gal. #HDGR60U Legacy Rose Eggshell Interior Paint with Primer</p> <p>Collaborative rating prediction: 0.01240113191306591</p> <p>Linear regression coef: 0.5792388400207467</p> <p><a href="#">View Details</a></p>	<p>5 gal. PPG1008-4 Gray By Me Flat Exterior Paint</p> <p>Collaborative rating prediction: 0.009758551605045795</p> <p>Linear regression coef: 0.513434652185284</p> <p><a href="#">View Details</a></p>	<p>5 gal. PPG1159-3 Heavenly Blue Semi-Gloss Interior Latex Paint</p> <p>Collaborative rating prediction: 0.006048777140676975</p> <p>Linear regression coef: 0.49081993435865956</p> <p><a href="#">View Details</a></p>	<p>1 gal. #HDGV14U French Blue Room Semi-Gloss Interior Paint with Primer</p> <p>Collaborative rating prediction: 0.004259902518242598</p> <p>Linear regression coef: 0.5761551767216737</p> <p><a href="#">View Details</a></p>
<p>1 gal. PPG1158-7 Stunning Sapphire Flat Interior Paint</p> <p>Collaborative rating prediction: 0.010906031355261803</p> <p>Linear regression coef: 0.5854061666188928</p> <p><a href="#">View Details</a></p>	<p>5 gal. PPG1171-1 Silent Delight Satin Interior Latex Paint</p> <p>Collaborative rating prediction: 0.009724339470267296</p> <p>Linear regression coef: 0.4959593731904479</p> <p><a href="#">View Details</a></p>	<p>1 qt. PPG1113-4 Green Gray Mist Eggshell Interior Latex Paint</p> <p>Collaborative rating prediction: 0.005121746566146612</p> <p>Linear regression coef: 0.5895177176843234</p> <p><a href="#">View Details</a></p>	<p>1 gal. PPG1136-2 Rainsong Semi-Gloss Interior Paint</p> <p>Collaborative rating prediction: 0.004244686104357243</p> <p>Linear regression coef: 0.5812946155534621</p> <p><a href="#">View Details</a></p>

Рисунок 3.7 – Приклад результату здійснення рекомендацій користувачу з зазначенням обчислених системою коефіцієнтів, що впливають на вибір рекомендацій

### 3.3 Аналіз тестування CRO-системи

Алгоритми машинного навчання є чутливими та вимагають ретельного дослідження сфери їх можливого застосування. Для перевірки того, чи дані алгоритми є доцільними необхідне проведення аналізу тестування, валідації та результатів роботи програми.

Тестування системи передбачає перевірку того, чи відповідає пошуковий запит тому, які продукти рекомендуються системою. Для проведення тестування були обрані методи тестування алгоритмів відповідні їх методології. Тестування глибокої нейронної мережі, що використовується як реалізація алгоритму колаборативної фільтрації, проведено метриками середньоквадратичного відхилення та середнього абсолютного відхилення (рис. 3.8). Дані метрики обчислюють відхилення результату від значення, що зазначено у вибірці. Значення похибок свідчить про наявність допустимого відхилення та доцільності використання мережі.

```
Test Metrics: {'mean_squared_error': 0.6940356492996216, 'mean_absolute_error': 0.7537084221839905}
```

### Рисунок 3.8 – Показники метрик втрат нейронної мережі

Для аналізу результатів були проведені тестові запити, що продемонстровані на рисунках 3.9-3.11. У зв'язку з необхідністю тестування для різних користувачів була добавлена можливість введення ідентифікатора користувача, від імені якого здійснюється запит, та сам текстовий запит. На рисунках 3.9 та 3.10 запит відбувається однаковий, однак у зв'язку з різницею ідентифікатора користувача – рекомендації відрізняються.

#### Product Search

Enter your query:

Enter user-id to check:

[Search](#)

#### Recommended Products:

Cluster Id: 6

<p>6 ft. x 42 ft. White Privacy Fence Screen Mesh Cover Screen with Reinforced Grommets for Garden Fence (Custom Size)            Collaborative rating prediction: 0.013966879807412624            Linear regression coef: 0.3862184297425773</p> <p style="text-align: center;"><a href="#" style="background-color: #dc3545; color: white; padding: 5px 10px; border-radius: 3px;">View Details</a></p> <p>5 ft. x 56 ft. Blue Privacy Fence Screen HDPE Mesh Windscreen with Reinforced Grommets for Garden Fence (Custom Size)            Collaborative rating prediction: 0.013141890056431293            Linear regression coef: 0.4024465222276161</p> <p style="text-align: center;"><a href="#" style="background-color: #dc3545; color: white; padding: 5px 10px; border-radius: 3px;">View Details</a></p> <p>3 ft. x 21 ft. Beige Privacy Fence Screen HDPE Mesh Windscreen with Reinforced Grommets for Garden Fence (Custom Size)            Collaborative rating prediction: 0.0025147576816380024            Linear regression coef: 0.3361707652875509</p> <p style="text-align: center;"><a href="#" style="background-color: #dc3545; color: white; padding: 5px 10px; border-radius: 3px;">View Details</a></p> <p>4 ft. x 14 ft. Beige Privacy Fence Screen HDPE Mesh Windscreen with Reinforced Grommets for Garden Fence (Custom Size)            Collaborative rating prediction: -0.002074938267469406            Linear regression coef: 0.33076455711429703</p> <p style="text-align: center;"><a href="#" style="background-color: #dc3545; color: white; padding: 5px 10px; border-radius: 3px;">View Details</a></p>	<p>8 ft. x 129 ft. Grey Privacy Fence Screen HDPE Mesh Windscreen with Reinforced Grommets for Garden Fence (Custom Size)            Collaborative rating prediction: -0.013887200504541397            Linear regression coef: 0.568808140112457</p> <p style="text-align: center;"><a href="#" style="background-color: #dc3545; color: white; padding: 5px 10px; border-radius: 3px;">View Details</a></p>
--	--

Рисунок 3.9 – Тестування рекомендацій для користувача «3» з пошуковим запитом «key»

**Product Search**

Enter your query:  
  
 Enter user-id to check:

**Recommended Products:**

Cluster Id: 5

<p>5 gal. #M440-2 Serene Breeze Satin Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: 0.014101085253059864            Linear regression coef: 0.7676365487229445  <input type="button" value="View Details"/></p> <p>1 gal. #P470-7 The Real Teal Satin Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: 0.00770652806386516            Linear regression coef: 0.6971711276636164  <input type="button" value="View Details"/></p> <p>5 gal. #PPU4-11 Porcelain Peach Satin Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: 0.006974945776164532            Linear regression coef: 0.7676365487229445  <input type="button" value="View Details"/></p> <p>1 gal. #PWN-53 White Mink Semi-Gloss Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: 0.006173935253173113            Linear regression coef: 0.6979539898018159  <input type="button" value="View Details"/></p>	<p>5 gal. #220A-1 Powdered Peach Semi-Gloss Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: 0.004882452078163624            Linear regression coef: 0.7715508594139415  <input type="button" value="View Details"/></p> <p>5 gal. #BIC-44 Chamois Leather Satin Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: 0.0011891366448253393            Linear regression coef: 0.7676365487229445  <input type="button" value="View Details"/></p> <p>1 gal. #P340-4 Lime Tree Semi-Gloss Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: 0.001045813551172614            Linear regression coef: 0.6979539898018159  <input type="button" value="View Details"/></p> <p>5 gal. #S-G-130 Luscious Semi-Gloss Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: 2.864367270376533E-4            Linear regression coef: 0.7715508594139415  <input type="button" value="View Details"/></p>	<p>1 gal. #P100-7 Sultana Semi-Gloss Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: -1.938305504154414E-4            Linear regression coef: 0.6971711276636164  <input type="button" value="View Details"/></p> <p>1 gal. #BIC-54 Vert Pierre Satin Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: -4.279756685718894E-4            Linear regression coef: 0.6971711276636164  <input type="button" value="View Details"/></p> <p>1 gal. #N470-4 NorWester Satin Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: -8.205018239095807E-4            Linear regression coef: 0.6971711276636164  <input type="button" value="View Details"/></p> <p>5 gal. #BWC-20 Melting Icicles Satin Enamel Exterior Paint &amp; Primer            Collaborative rating prediction: -8.922705892473459E-4            Linear regression coef: 0.7676365487229445  <input type="button" value="View Details"/></p>
---	--	--

Рисунок 3.10 – Тестування рекомендацій для користувача «10» з пошуковим запитом «key»

**Product Search**

Enter your query:  
  
 Enter user-id to check:

**Recommended Products:**

Cluster Id: 11

<p>1 gal. #HDG006 Lotus Blossom Semi-Gloss Interior Paint with Primer            Collaborative rating prediction: 0.008621900342404842            Linear regression coef: 0.6889510752125227  <input type="button" value="View Details"/></p> <p>3 ft. x 21 ft. Beige Privacy Fence Screen HDPE Mesh Windscreen with Reinforced Grommets for Garden Fence (Custom Size)            Collaborative rating prediction: 0.008007708005607128            Linear regression coef: 0.6994101133788667  <input type="button" value="View Details"/></p> <p>5 gal. #HDPG57 Frosty Mint Flat Interior Paint and Primer            Collaborative rating prediction: 0.007818911224603653            Linear regression coef: 0.7179248029472826  <input type="button" value="View Details"/></p> <p>1 gal. #HDGCN21 Blue Forest Eggshell Interior Paint with Primer            Collaborative rating prediction: 0.007553504314273596            Linear regression coef: 0.6877767820052236  <input type="button" value="View Details"/></p>	<p>5 gal. #HDGV55U Soft Violet Semi-Gloss Interior Paint with Primer            Collaborative rating prediction: 0.007020902819931507            Linear regression coef: 0.727710579674775  <input type="button" value="View Details"/></p> <p>5 gal. PPG1030-1 Brainstorm Eggshell Interior Paint with Primer            Collaborative rating prediction: 0.0047423355281353            Linear regression coef: 0.7202733893618808  <input type="button" value="View Details"/></p> <p>1 gal. PPG1009-2 Tornado Flat Interior Paint with Primer            Collaborative rating prediction: 0.0044153654016554356            Linear regression coef: 0.6869939198670242  <input type="button" value="View Details"/></p> <p>1 qt. PPG1035-1 Watery Blue Semi-Gloss Interior Paint with Primer            Collaborative rating prediction: 0.00437537394464016            Linear regression coef: 0.6830796091760273  <input type="button" value="View Details"/></p>	<p>5 gal. #HDGY06 Com Silk Semi-Gloss Interior Paint with Primer            Collaborative rating prediction: 0.0031875506974756718            Linear regression coef: 0.727710579674775  <input type="button" value="View Details"/></p> <p>1 gal. #HDGV12 Shady Blue Satin Interior Paint with Primer            Collaborative rating prediction: 0.002330581657588482            Linear regression coef: 0.6885596441434231  <input type="button" value="View Details"/></p> <p>5 gal. My Alibi PPG1018-2 Semi-Gloss Interior Paint with Primer            Collaborative rating prediction: 0.002255981555208564            Linear regression coef: 0.727710579674775  <input type="button" value="View Details"/></p> <p>1 gal. Windsor Haze PPG1167-2 Eggshell Interior Paint with Primer            Collaborative rating prediction: 0.002067478373646736            Linear regression coef: 0.6877767820052236  <input type="button" value="View Details"/></p>
---	---	---

Рисунок 3.11 – Тестування рекомендацій для користувача «10» з пошуковим запитом «string»

## ВИСНОВКИ

В ході дослідження була реалізована CRO-система на основі трьох алгоритмів машинного навчання, таких як: глибокі нейронні мережі, кластеризація текстів та лінійної регресії.

Проаналізовані популярні методи проведення оптимізації рейтингу конверсій без залучення машинного навчання, а саме: A/B тестування, аналіз веб-аналітики та персоналізації контенту. Визначено переваги та недоліки перелічених методів. A/B тестування характеризується більшою гнучкістю, тобто дозволяє постійно реагувати на зміни поведінки користувачів, однак, вимагає значних витрат часу та ресурсів. Аналіз веб-аналітики допомагає визначити поведінку користувача та ефективність елементів веб-сайту, але, даний вид аналізу залежить від якості даних. Персоналізація контенту дозволяє адаптувати наповненість системи до індивідуальних потреб користувача, покращити їх залученість до виконання бажаних дій, але даний метод чутливий до шумів у даних та потребує приватну інформацію користувачів.

Описана архітектура CRO-системи та визначені її характеристики для взаємодії з веб-сайтом онлайн-магазину. Основними характеристиками є: час відклику сайту, формування візуальних списків рекомендованих товарів, сортування продуктів на основі обчислених значень CRO-системи.

Були реалізовані алгоритми колаборативної фільтрації на основі глибокої нейронної мережі, кластеризація текстів алгоритмом K-means та лінійна регресія. Дані методи були застосовані в CRO-системі, визначенна їх область застосування. Колаборативна фільтрація застосована для передбачення оцінок користувачів на основі сусідства. Кластеризація текстів використовується для визначення набору продуктів, що відповідають текстовому пошуковому запиту користувача. Лінійна регресія реалізована для здійснення сортування продуктів на основі їх ціни, аналізуючи історію покупок та переглядів користувачів. Спроектвана CRO-система була застосована до веб-сайту онлайн-магазину. Наповненість онлайн-магазину відбулося на основі відкритих датасетів з попередньою обробкою для

використання методів машинного навчання. Описаний алгоритм здійснення формування рекомендацій та поетапний аналіз застосованих методів.

Проаналізовано результати роботи CRO-системи на основі машинного навчання. Для нейронної мережі обчислені значення похибок навчання, які свідчать про допустимі відхилення та доцільність використання. Проаналізовані результати сформованих рекомендацій на основі кластерного аналізу та лінійної регресії. Було встановлено, що за допомогою кластерного аналізу вирішується проблема «холодного старту», коли відсутня достатня кількість маркованих даних. А лінійна регресія є ефективною для обробки великих обсягів даних.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Purnomo Y. J. Digital marketing strategy to increase sales conversion on e-commerce platforms. *Journal of contemporary administration and management (ADMAN)*. 2023. Т. 1, № 2. С. 54–62. URL: <https://doi.org/10.61100/adman.v1i2.23> (дата звернення: 05.05.2024).
2. Kamal M., Aziz Bablu T. Machine learning models for predicting click-through rates on social media: factors and performance analysis. *International journal of applied machine learning and computational intelligence*. 2022. URL: <https://neuralslate.com/index.php/Machine-Learning-Computational-I/article/view/36> (дата звернення: 05.05.2024).
3. Sarker I. H. Machine learning: algorithms, real-world applications and research directions. *SN computer science*. 2021. Т. 2, № 3. URL: <https://doi.org/10.1007/s42979-021-00592-x> (дата звернення: 05.05.2024).
4. A comprehensive survey on machine learning-based big data analytics for iot-enabled smart healthcare system / W. Li та ін. *Mobile networks and applications*. 2021. URL: <https://doi.org/10.1007/s11036-020-01700-6> (дата звернення: 22.05.2024).
5. Pritesh B. The 7 step framework of CRO - conversion rate optimization in 2020. *WP Pluginsify*. URL: <https://wppluginsify.com/wordpress-plugins/cro-conversion-rate-optimization/> (дата звернення: 07.05.2024).
6. Hylewski M. Conversion rate optimization with AI in 2023 (+6 examples). *Landingi*. URL: <https://landingi.com/blog/conversion-rate-optimization-with-ai/> (дата звернення: 07.05.2024).
7. Msn K. Improve Conversions with session recording for ecommerce. *Digital Experience Error Tracking Product Analytics Zipy*. URL: <https://www.zipy.ai/blog/session-recording-for-ecommerce> (дата звернення: 09.05.2024).
8. A/B testing: a systematic literature review / F. Quin та ін. *Journal of systems and software*. 2024. С. 112011. URL: <https://doi.org/10.1016/j.jss.2024.112011> (дата звернення: 11.05.2024).

9. Camarena A. What is page speed & how to improve it. *Semrush Blog*. URL: <https://www.semrush.com/blog/page-speed/> (дата звернення: 12.05.2024).
10. What Is eCommerce CTA and why is it important?. *DGI Group*. URL: <https://dgi.com/blog/what-is-ecommerce-cta-and-why-is-it-important> (дата звернення: 12.05.2024).
11. Train W. SEO copywriting for e-commerce: crafting compelling product descriptions. *Medium*. URL: <https://medium.com/@worldtrain/seo-copywriting-for-e-commerce-crafting-compelling-product-descriptions-18daf3dc8203> (дата звернення: 12.05.2024).
12. Гордєєв Р. С., Граф М. С. Аналіз існуючих алгоритмів музичних рекомендаційних систем. *Технічна інженерія*. 2022. № 2(90). С. 87–93. URL: [https://doi.org/10.26642/ten-2022-2\(90\)-87-93](https://doi.org/10.26642/ten-2022-2(90)-87-93) (дата звернення: 13.05.2024).
13. Explaining deep neural networks and beyond: a review of methods and applications / W. Samek та ін. *Proceedings of the IEEE*. 2021. Т. 109, № 3. С. 247–278. URL: <https://doi.org/10.1109/jproc.2021.3060483> (дата звернення: 14.05.2024).
14. What is a neural network?. *IBM*. URL: <https://www.ibm.com/topics/neural-networks> (дата звернення: 15.05.2024).
15. What are convolutional neural networks?. *IBM*. URL: <https://www.ibm.com/topics/convolutional-neural-networks> (дата звернення: 15.05.2024).
16. What are recurrent neural networks?. *IBM*. URL: <https://www.ibm.com/topics/recurrent-neural-networks> (дата звернення: 16.05.2024).
17. Keita Z. Classification in machine learning: a guide for beginners. *DataCamp*. URL: <https://www.datacamp.com/blog/classification-machine-learning> (дата звернення: 18.05.2024).
18. Tan H. Machine learning algorithm for classification. *Journal of physics: conference series*. 2021. Т. 1994, № 1. С. 012016. URL: <https://doi.org/10.1088/1742-6596/1994/1/012016> (дата звернення: 18.05.2024).

19. Contreras O. C. Gaussian mixture model explained. *Built In*. URL: <https://builtin.com/articles/gaussian-mixture-model> (дата звернення: 18.05.2024).
20. Saini A. Guide on support vector machine (SVM) algorithm. *Analytics Vidhya*. URL: <https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/> (дата звернення: 20.05.2024).
21. Sruthi E. R. Understand random forest algorithms with examples. *Analytics Vidhya*. URL: <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/> (дата звернення: 22.05.2024).
22. Ray S. Naive bayes classifier explained: applications and practice problems of naive bayes classifier. *Analytics Vidhya*. URL: <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/> (дата звернення: 22.05.2024).
23. Maulud D., Abdulazeez A. M. A review on linear regression comprehensive in machine learning. *Journal of applied science and technology trends*. 2020. Т. 1, № 4. С. 140–147. URL: <https://doi.org/10.38094/jastt1457> (дата звернення: 22.05.2024).
24. Aravind C. R. Exploring clustering algorithms: explanation and use cases. *neptune ai*. URL: <https://neptune.ai/blog/clustering-algorithms> (дата звернення: 22.05.2024).
25. Amazon - Ratings (Beauty Products). *Kaggle: Your Machine Learning and Data Science Community*. URL: <http://surl.li/rxrvk>.
26. Home Depot Product Datasets. *Data world*. URL: <http://surl.li/rxrwy>.
27. Oracle. Java. SE. Версія 17. URL: <https://docs.oracle.com/en/java/javase/17/docs/api/index.html> (дата звернення: 23.05.2024).
28. Eclipse Foundation. Jakarta servlet. Версія 5.0.0. URL: <https://jakarta.ee/specifications/servlet/5.0/> (дата звернення: 22.05.2024).
29. Apache. Tomcat. Версія 10.1.24. URL: <https://tomcat.apache.org/> (дата звернення: 23.05.2024).

30. Конотопчик А. М., Мельник К. В., Лавренчук С. В. Підвищення коефіцієнта конверсії за допомогою машинного навчання: колаборативна фільтрація. *Інформаційні технології: теорія і практика* : міжнар. науково-практ. конф., м. Дніпро, 20–22 берез. 2024 р. Дніпро, 2024. С. 187–190. URL: <http://ir.nmu.org.ua/handle/123456789/166565> (дата звернення: 23.05.2024).

31. Конотопчик, А., Мельник К., Лавренчук С. (2024). Методики CRO на основі машинного навчання. *Information Technology: Computer Science, Software Engineering and Cyber Security*.

32. Ahmed M., Seraj R., Islam S. M. S. The k-means algorithm: a comprehensive survey and performance evaluation. *Electronics*. 2020. Т. 9, № 8. С. 1295. URL: <https://doi.org/10.3390/electronics9081295> (дата звернення: 24.05.2024).

33. Apache. Commons. Версія 3.5. URL: <https://commons.apache.org/> (дата звернення: 24.05.2024).

# ДОДАТКИ

## Додаток А

### Лістинг програми реалізації алгоритму колаборативної фільтрації глибокою нейронною мережею

```

import sys
import keras
import numpy as np
import pandas as pd
from keras.callbacks import EarlyStopping
from keras.models import Model
from sklearn.preprocessing import MinMaxScaler
from keras.layers import Dot, Embedding, Flatten, Input, Dense
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from keras import metrics
# Завантаження та очистка даних
df = pd.read_csv('B:\\\\competition\\comp1\\diploma\\newratings_Beauty.csv',
nrows=2777)
allProducts = df['ProductId'].unique()
# Функція для побудови моделі
def build_model(num_users, num_products, embedding_size):
    user_input = Input(shape=(), name='user_input')
    product_input = Input(shape=(), name='product_input')
    user_embedding = Embedding(input_dim=num_users, output_dim=embedding_size,
name='user_embeddings')(user_input)
    product_embedding = Embedding(input_dim=num_products,
output_dim=embedding_size, name='product_embeddings')(product_input)
    dot_product = Dot(axes=1, name='dot_product')([user_embedding,
product_embedding])
    flatten = Flatten(name='flatten_layer')(dot_product)
    model = Model(inputs=[user_input, product_input], outputs=flatten)
    return model
# Побудова та компіляція моделі
model = build_model(len(df['UserId'].unique()), len(df['ProductId'].unique()), 50)
model.compile(optimizer=Adam(learning_rate=0.001), loss='mse',
metrics=[metrics.mean_squared_error, metrics.mean_absolute_error])
# Розділення на тренувальні та тестові дані
x_train, x_test, y_train, y_test = train_test_split(df[['UserId', 'ProductId']],
df[['Rating']], test_size=0.1)
# Раннє зупинення для уникнення перетренування
early_stopping = EarlyStopping(monitor='val_loss', patience=3,
restore_best_weights=True)
# Тренування моделі
model.fit(x={'user_input': x_train['UserId'], 'product_input':
x_train['ProductId']}, y=y_train['Rating'], batch_size=128, epochs=10,
validation_split=0.1, callbacks=[early_stopping])
# Функція для рекомендації продуктів для конкретного користувача
def recommend_product_for_user(user_id):
    user_ids = [user_id] * len(allProducts)
    data = pd.DataFrame({'UserId': user_ids, 'ProductId': allProducts})
    data = data[['UserId', 'ProductId']]
    # Передбачення рейтингу для користувача та всіх товарів
    predicted_ratings = model.predict([data['UserId'], data['ProductId']])
    # Фрейм з передбаченими оцінками
    all_products_per_user = pd.DataFrame({'UserId': user_ids, 'ProductId':
allProducts, 'Rating': predicted_ratings.flatten()})
    all_products_per_user_sorted =
all_products_per_user.sort_values(by='Rating', ascending=False)
    return all_products_per_user_sorted
if __name__ == "__main__":
    if len(sys.argv) < 2:
        print("Usage: python colaborative.py <userid>")
        sys.exit(1)
    query = int(sys.argv[1])
    recommendations = recommend_product_for_user(query)
    for i, r in recommendations.iterrows():
        print(int(r['ProductId']), r['Rating'])

```

## Додаток Б

### Лістинг програми реалізації кластерного аналізу опису продуктів

```

import sys

import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans

# Імпортування датасету
product_descriptions =
pd.read_csv('B:\\competition\\comp1\\diploma\\data_fixed_indexed.csv')

# Використання TfidfVectorizer для перетворення текстових даних у матрицю Term
Frequency-Inverse Document Frequency (TF-IDF)
vectorizer = TfidfVectorizer(stop_words='english')
X = vectorizer.fit_transform(product_descriptions["description"])

# Виконання кластеризації алгоритмом KMeans з 15-ма кластерами
kmeans = KMeans(n_clusters=15, init='k-means++')
y_kmeans = kmeans.fit_predict(X)

# Вивід десяти "термів" (слів) з зазначеного кластеру
def print_cluster_terms(i):
    print(f"Кластер {i}: {' '.join(terms[ind] for ind in order_centroids[i,
:10])}")

# Кількість кластерів
true_k = 15

# Застосування моделі KMeans та вивід 10 найвживаніших термів в кожному кластері
model = KMeans(n_clusters=true_k, init='k-means++', max_iter=1000,
n_init=1).fit(X)
order_centroids = model.cluster_centers_.argsort()[:, :-1]
terms = vectorizer.get_feature_names_out()

def get_products_in_cluster(cluster_id):
    products_in_cluster = []
    for index, cluster_label in enumerate(y_kmeans):
        if cluster_label == cluster_id:
            products_in_cluster.append(product_descriptions.iloc[index]["id"])
    return products_in_cluster

def show_recommendations(product):
    Y = vectorizer.transform([product])
    prediction = model.predict(Y)
    cluster_id = prediction[0]
    print(cluster_id)
    products_in_cluster = get_products_in_cluster(cluster_id)
    return products_in_cluster

if __name__ == "__main__":
    if len(sys.argv) < 2:
        print("Usage: python cluster.py <query>")
        sys.exit(1)

    query = sys.argv[1]
    recommendations = show_recommendations(query)
    print(recommendations)

```