

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Факультет інформаційних технологій
(факультет)

Кафедра системного аналізу та управління
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

Здобувача вищої освіти Жижикіної Єлизавети Дмитрівни
академічної групи 124-21-1
спеціальності 124 Системний аналіз
за освітньо-професійною програмою Системний аналіз

на тему: «Аналіз ключових показників для розробки рекомендаційної системи»

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	Інституційною	
кваліфікаційної роботи	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			
розділів:	2			
Інформаційно- аналітичний	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			
Спеціальний розділ	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			
Рецензент	<i>д.т.н, професор, зав. каф. ПЗКС Алексєєв М.О.</i>			
Нормоконтролер	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			

Дніпро
2025

ЗАТВЕРДЖЕНО:
завідувач кафедри
Системного аналізу та управління
(повна назва)

_____ к.т.н., доц. Желдак Т.А.
(підпис) (прізвище, ініціали)

« ____ » _____ 20 ____ року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавра

здобувачу вищої освіти Жижикіній Є. Д. академічної групи 124- 21-1
спеціальності: 124 Системний аналіз
за освітньо-професійною програмою Системний аналіз
на тему «Аналіз ключових показників для розробки рекомендаційної системи»

затверджену наказом ректора НТУ «Дніпровська політехніка» від 05.05.2025
р. №336-с

Розділ	Зміст	Терміни виконання
1. Інформаційно-аналітичний розділ	Аналіз літератури за темою, вибір напрямів дослідження, обґрунтування актуальності та постановка проблеми.	10.01.2025 – 01.03.2025
2. Спеціальний розділ	Опис методів дослідження, розробка рішень, виклад результатів.	01.03.2025 – 28.05.2025

Завдання видано _____ доц. Хом'як Т.В.
(підпис) (прізвище, ініціали)

Дата видачі: 06.09.2024 р.

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ Жижикіна Є. Д.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 54 с., 9 Рисуноків, 0 табл., 3 додатки, 18 джерел.

Об'єкт дослідження: набір даних «Women's E-Commerce Clothing Reviews» (23 486 відгуків про жіночий одяг).

Предмет дослідження: ключові показники (рейтинги, відгуки, рекомендації) та їхній вплив на персоналізовані рекомендації.

Мета: аналіз даних відгуків для розробки рекомендаційної системи веб-сайту «Elizabeth's Elegance» для підвищення персоналізації та ефективності e-commerce.

Методи: системний аналіз, обробка даних у Power BI, програмування (FastAPI, SQL, SQLAlchemy, JavaScript), колаборативна та контентна фільтрація, асоціативні правила.

Інформаційно-аналітичний розділ: теоретичні основи рекомендаційних систем (колаборативна, контентна, гібридна фільтрація), оцінка ключових показників, методи аналізу даних (асоціативні правила, машинне навчання, веб-аналітика).

Спеціальний розділ: аналіз даних, визначення популярності товарів (рейтинг, відгуки, рекомендації), розробка веб-сайту з рекомендаційною системою (рекомендації за віком, схожі товари, фільтри, сортування, інтерактивні графіки).

Практична цінність: створення рекомендаційної системи для «Elizabeth's Elegance» (цільова аудиторія — жінки 30–50 років), підвищення залученості, оптимізація асортименту та маркетингу.

Ключові слова: рекомендаційна система, e-commerce, аналіз даних, Power BI, FastAPI, SQLAlchemy, персоналізація, рейтинги, відгуки, JavaScript, Chart.js.

ABSTRACT

Explanatory note: 54 pages, 9 figures, 0 tables, 3 appendices, 18 sources.

Research object: the ‘Women's E-Commerce Clothing Reviews’ dataset (23,486 reviews of women's clothing).

Subject of research: key indicators (ratings, reviews, recommendations) and their impact on personalised recommendations.

Objective: to analyse review data to develop a recommendation system for the Elizabeth's Elegance website to improve personalisation and e-commerce efficiency.

Methods: system analysis, data processing in Power BI, programming (FastAPI, SQL, SQLAlchemy, JavaScript), collaborative and content filtering, associative rules.

Information and analytical section: theoretical foundations of recommendation systems (collaborative, content, hybrid filtering), evaluation of key indicators, data analysis methods (associative rules, machine learning, web analytics).

Special section: data analysis, determining product popularity (ratings, reviews, recommendations), developing a website with a recommendation system (recommendations by age, similar products, filters, sorting, interactive graphs).

Practical value: creation of a recommendation system for Elizabeth's Elegance (target audience: women aged 30–50), increased engagement, optimisation of product range and marketing.

Keywords: recommendation system, e-commerce, data analysis, Power BI, FastAPI, SQLAlchemy, personalisation, ratings, reviews, JavaScript, Chart.js.

ЗМІСТ

ВСТУП.....	7
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
РОЗДІЛ 1. ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ.....	10
1.1 Поняття та принципи рекомендаційних систем.....	10
1.2 Ключові показники оцінки рекомендаційних систем (рейтинги, відгуки, рекомендації).....	12
1.3 Огляд існуючих підходів до аналізу даних для рекомендаційних систем.	14
РОЗДІЛ 2. СПЕЦІАЛЬНИЙ.....	16
2.1 Аналіз даних для розробки рекомендаційної системи.....	16
2.1.1 Опис набору даних (відгуки, рейтинги, коментарі користувачів про жіночий одяг).....	16
2.1.2 Аналіз розподілу рейтингів, рекомендацій та відгуків за категоріями, департаментами та віком користувачів.	18
2.1.3 Визначення ключових показників популярності товарів (рейтинг, кількість позитивних відгуків, рекомендації)	20
2.2 Реалізація рекомендаційної функціональності.....	22
2.2.1 Рекомендації товарів на основі віку користувачів.....	22
2.2.2 Рекомендації товарів на основі обраного товару	24
2.2.3 Інтеграція рекомендаційної функціональності у веб-сайт	26
2.3 Проектування веб-сайту з рекомендаційною системою	30
2.3.1 Вимоги до веб-сайту та структура інтерфейсу (головна сторінка, фільтри, топ-товарів).....	30
2.3.2 Реалізація фільтрів та сортування товарів.....	36
2.3.3 Інтерактивні елементи для аналізу даних.....	46
2.4 Висновки до розділу	56

ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
ДОДАТКИ	64
Додаток А. Відомість матеріалів кваліфікаційної роботи	64
Додаток Б. Відгук.....	65
Додаток В. Участь у конференції.....	66

ВСТУП

Сучасний розвиток електронної комерції зумовлює зростання попиту на персоналізовані рішення, які підвищують зручність взаємодії користувачів із платформами. Рекомендаційні системи стали ключовим інструментом для аналізу вподобань клієнтів і пропонування релевантних товарів, що сприяє збільшенню продажів і лояльності аудиторії. Аналіз великих обсягів даних про відгуки, рейтинги та рекомендації дозволяє виявляти закономірності в поведінці користувачів, оптимізувати асортимент і вдосконалювати маркетингові стратегії. Актуальність теми дослідження зумовлена необхідністю створення ефективних рекомендаційних систем, які відповідають потребам цільової аудиторії та підвищують конкурентоспроможність онлайн-магазинів.

Метою кваліфікаційної роботи є аналіз ключових показників для розробки рекомендаційної системи, що базується на даних відгуків користувачів про жіночий одяг. Основні завдання включають: аналіз розподілу рейтингів, рекомендацій і відгуків за категоріями, департаментами та віком користувачів у Power BI; визначення показників популярності товарів; розробку та інтеграцію рекомендаційної функціональності у веб-сайт; створення адаптивного інтерфейсу з інтерактивними елементами для аналізу даних.

Об'єктом дослідження є набір даних «Women's E-Commerce Clothing Reviews», що містить 23 486 відгуків про жіночий одяг. Предметом дослідження виступають ключові показники (рейтинги, відгуки, рекомендації) та їхній вплив на формування персоналізованих рекомендацій. Методологічною основою роботи є методи системного аналізу, обробки даних у Power BI, програмування (FastAPI, SQL, JavaScript) та принципи побудови рекомендаційних систем.

Робота складається з двох розділів. У першому, інформаційно-аналітичному, розглянуто теоретичні основи рекомендаційних систем, принципи їхньої роботи, ключові показники оцінки та підходи до аналізу даних. У другому, спеціальному, представлено практичну реалізацію: аналіз даних, розробку веб-

сайту «Elizabeth's Elegance» із рекомендаційною системою, інтеграцію фільтрів, сортування та інтерактивних елементів для аналізу статистики. Робота завершується висновками, списком використаних джерел і додатками.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Позначення	Розшифрування
Clothing.ID	Унікальний ідентифікатор товару в наборі даних
Age	Вік користувача, який залишив відгук (18–99 років)
Rating	Оцінка товару (від 1 до 5)
Review.Text	Текстовий відгук користувача
Recommended.IND	Бінарний індикатор рекомендації (1 — рекомендується, 0 — не рекомендується)
Positive.Feedback.Count	Кількість корисних оцінок відгуку іншими користувачами
Division.Name	Назва підрозділу товару (наприклад, General, Petite)
Department.Name	Назва департаменту товару (наприклад, Tops, Dresses)
Class.Name	Назва категорії товару (наприклад, Dresses, Knits)
Title	Назва товару в наборі даних
Price	Ціна товару в наборі даних
Number	Номер відгуку для сортування
Power BI	Програмне забезпечення для аналізу та візуалізації даних
FastAPI	Фреймворк для створення API-ендпоінтів
SQL	Мова структурованих запитів для роботи з базами даних
SQLAlchemy	Бібліотека для роботи з реляційними базами даних через ORM
Microsoft SQL Server	Реляційна база даних для зберігання та обробки даних
Jinja2	Шаблонізатор для створення HTML-сторінок
Chart.js	Бібліотека JavaScript для побудови інтерактивних графіків
noUiSlider	Бібліотека JavaScript для створення слайдерів діапазону
CSS	Каскадні таблиці стилів для оформлення інтерфейсу
HTML	Мова розмітки для створення веб-сторінок
JavaScript	Мова програмування для інтерактивності веб-сайту

РОЗДІЛ 1. ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ

1.1 Поняття та принципи рекомендаційних систем

Рекомендаційні системи є програмними інструментами, які аналізують дані користувачів і пропонують релевантні продукти, послуги чи контент, підвищуючи зручність взаємодії з інформаційними платформами. Вони широко застосовуються в електронній комерції, соціальних мережах і потокових сервісах для персоналізації досвіду користувача. Основна мета таких систем — передбачити вподобання користувача на основі його попередньої поведінки, оцінок чи інших даних. Рекомендаційні системи допомагають зменшити інформаційне перевантаження, пропонуючи лише ті елементи, які відповідають інтересам користувача. Наприклад, у сфері електронної комерції вони сприяють підвищенню продажів шляхом рекомендації товарів, подібних до тих, що користувач уже переглядав [1, 2].

Основними принципами роботи рекомендаційних систем є використання колаборативної фільтрації, контентної фільтрації та гібридних підходів. Колаборативна фільтрація базується на аналізі схожості між користувачами або елементами, використовуючи оцінки чи історію взаємодій. Наприклад, якщо два користувачі мають схожі вподобання, система може запропонувати одному з них товари, які сподобалися іншому. Контентна фільтрація, навпаки, аналізує характеристики самих об'єктів (наприклад, категорію одягу чи бренд) і зіставляє їх із профілем користувача. Гібридні підходи поєднують обидва методи, щоб підвищити точність рекомендацій і подолати обмеження, такі як проблема «холодного старту», коли недостатньо даних про нового користувача [1, 2, 7].

Сучасні рекомендаційні системи також використовують асоціативні правила та методи машинного навчання для вдосконалення рекомендацій. Асоціативні правила виявляють зв'язки між об'єктами, наприклад, пропонуючи аксесуари до придбаного одягу. Методи машинного навчання, такі як нейронні

мережі, дозволяють обробляти великі обсяги даних і враховувати складні шаблони поведінки. У групових рекомендаційних системах, які пропонують контент для кількох користувачів, застосовуються алгоритми досягнення консенсусу, щоб збалансувати інтереси всіх учасників. Такі системи часто використовують агрегацію даних і аналіз групової поведінки [7, 9].

Важливим аспектом рекомендаційних систем є їхня адаптивність до контексту, наприклад, часу, місця чи типу пристрою. Наприклад, система може пропонувати теплий одяг узимку або легкі сукні влітку, враховуючи сезонність. Крім того, рекомендаційні системи мають бути стійкими до маніпуляцій і забезпечувати справедливість рекомендацій, уникаючи упередженості щодо певних груп користувачів. Ці принципи забезпечують високу якість взаємодії з користувачем і сприяють довірі до системи [1, 9].

1.2 Ключові показники оцінки рекомендаційних систем (рейтинги, відгуки, рекомендації)

Оцінка ефективності рекомендаційних систем є критично важливою для забезпечення їхньої якості та відповідності потребам користувачів. Ключові показники оцінки включають точність, покриття, різноманітність, новизну та робастність рекомендацій. Точність визначає, наскільки запропоновані елементи відповідають інтересам користувача, і часто вимірюється метриками, такими як середня абсолютна похибка (MAE) або середньоквадратична похибка (RMSE). Покриття оцінює, яка частка елементів із каталогу може бути рекомендована користувачам. Різноманітність гарантує, що рекомендації не обмежуються схожими об'єктами, а новизна сприяє пропонуванню нових, раніше невідомих користувачу елементів [3].

Рейтинги є одним із основних показників оцінки, оскільки вони відображають пряму оцінку користувачами якості товару чи послуги. У рекомендаційних системах рейтинги використовуються для побудови профілів користувачів і прогнозування їхніх уподобань. Наприклад, високий середній рейтинг товару може свідчити про його популярність і релевантність для рекомендації. Однак низька кількість оцінок може знижувати достовірність цього показника. Важливо також враховувати розподіл рейтингів, щоб уникнути упередженості до товарів із невеликою кількістю високих оцінок [3, 5].

Відгуки користувачів слугують додатковим джерелом інформації для оцінки рекомендаційних систем. Якісні відгуки, які містять розгорнуті описи досвіду використання товару, дозволяють оцінити релевантність рекомендацій. Згідно з критеріями Google, відгуки мають бути експертними, містити переваги та недоліки продукту, а також порівняння з аналогами. Такі відгуки підвищують довіру до системи, оскільки користувачі можуть оцінити, наскільки рекомендація відповідає їхнім очікуванням. Аналіз відгуків також допомагає виявити сентимент, що впливає на якість рекомендацій [4].

Ефективність рекомендаційних систем також оцінюється через показники, пов'язані з бізнес-цілями, такі як конверсія, кліки та дохід. Конверсія вимірює, скільки користувачів здійснили покупку після перегляду рекомендації. Кліки показують, наскільки рекомендації привертають увагу користувачів. Ці показники є важливими в інтернет-маркетингу, оскільки вони безпосередньо впливають на комерційний успіх платформи. Наприклад, рекомендаційна система, яка збільшує кількість кліків на товари, сприяє підвищенню продажів і лояльності клієнтів [5].

Робастність рекомендаційних систем оцінює їхню стійкість до маніпуляцій, таких як штучне завищення рейтингів або створення фальшивих відгуків. Система має бути здатною виявляти аномалії в даних, щоб забезпечити достовірність рекомендацій. Крім того, справедливість рекомендацій є важливим показником, який гарантує, що система не віддає перевагу певним групам користувачів або товарам. Усі ці показники разом дозволяють комплексно оцінити якість рекомендаційної системи та її відповідність потребам користувачів і бізнесу [3, 4].

1.3 Огляд існуючих підходів до аналізу даних для рекомендаційних систем

Аналіз даних є основою функціонування рекомендаційних систем, оскільки він дозволяє виявляти шаблони поведінки користувачів і формувати персоналізовані пропозиції. Існують різні підходи до обробки даних, зокрема колаборативна фільтрація, контентна фільтрація, асоціативні правила та методи машинного навчання. Колаборативна фільтрація аналізує взаємодії користувачів із елементами, використовуючи матриці оцінок або історії переглядів. Цей підхід ефективний для великих обсягів даних, але стикається з проблемою «холодного старту» для нових користувачів. Контентна фільтрація, навпаки, базується на характеристиках елементів і профілях користувачів, що робить її менш залежною від кількості взаємодій [7, 9].

Асоціативні правила є популярним підходом для аналізу даних у рекомендаційних системах, особливо в електронній комерції. Вони дозволяють виявляти зв'язки між елементами, наприклад, пропонувати аксесуари до придбаного одягу на основі частих комбінацій покупок. Цей метод є простим у реалізації, але потребує значних обчислювальних ресурсів для обробки великих наборів даних. У групових рекомендаційних системах асоціативні правила доповнюються алгоритмами консенсусу, які враховують інтереси кількох користувачів. Такі системи аналізують групову поведінку, щоб запропонувати компромісні рекомендації [7].

Методи машинного навчання, такі як кластерний аналіз і нейронні мережі, широко застосовуються для аналізу даних у сучасних рекомендаційних системах. Кластерний аналіз групує користувачів або елементи за схожими характеристиками, що покращує точність рекомендацій. Нейронні мережі дозволяють обробляти складні нелінійні залежності в даних, наприклад, враховувати контекстні фактори, такі як час або місце. Однак ці методи потребують великих обсягів даних і високої обчислювальної потужності. У

телекомунікаційних підприємствах аналіз даних із використанням ІІІ також включає обробку великих даних для персоналізації пропозицій [9].

Аналіз даних для рекомендаційних систем може включати елементи веб-аналітики, яка оцінює поведінку користувачів на сайті. Наприклад, інструменти веб-аналітики дозволяють відстежувати кліки, час перебування на сторінці та послідовність переглядів. Ці дані використовуються для побудови профілів користувачів і вдосконалення рекомендацій. У контексті кібербезпеки аналіз даних допомагає виявляти аномалії в поведінці користувачів, що підвищує надійність системи. Комплексне застосування цих підходів забезпечує високу якість рекомендацій і адаптивність до змін у вподобаннях користувачів [6, 8].

РОЗДІЛ 2. СПЕЦІАЛЬНИЙ

2.1 Аналіз даних для розробки рекомендаційної системи

2.1.1 Опис набору даних (відгуки, рейтинги, коментарі користувачів про жіночий одяг)

Набір даних для рекомендаційної системи містить 23 486 відгуків на товари жіночого одягу, зібраних із платформи електронної комерції. Дані представлені у форматі таблиці з 10 основними полями, що охоплюють характеристики товару, оцінки, текстові коментарі та демографічні дані користувачів. Ключові поля включають Clothing ID (унікальний ідентифікатор товару), Age (вік рецензента, 18–99 років), Rating (оцінка від 1 до 5), Review Text (текстовий відгук), Recommended IND (бінарний індикатор рекомендації, 0 або 1), Positive Feedback Count (кількість корисних оцінок відгуку), Division Name, Department Name і Class Name (категорії товару). Наприклад, один запис описує сукню (Class Name = Dresses) з рейтингом 5, рекомендовану 34-річним користувачем.

Поле Clothing ID забезпечує унікальну ідентифікацію товарів для групування відгуків. Поле Rating відображає якість товару, де 5 — найвища оцінка. Поле Recommended IND показує, чи рекомендує користувач товар (1 — так, 0 — ні), що є ключовим для оцінки популярності. Поле Positive Feedback Count вказує, скільки інших користувачів визнали відгук корисним, впливаючи на довіру до товару. Текстові відгуки (Review Text) містять суб'єктивні описи, але використовуються лише для відображення, без аналізу настрою. Категорії (Class Name) включають «Dresses», «Tops», «Bottoms», «Jackets», «Intimates», а департаменти (Department Name) — «Dresses», «Tops», «Bottoms», «Intimate».

Дані зберігаються в базі даних Microsoft SQL Server і обробляються через SQLAlchemy у фреймворку FastAPI. Вік користувачів переважно становить 30–45 років, що відповідає цільовій аудиторії платформи. Набір даних є основою для

аналізу вподобань за категоріями, віком і оцінками, що забезпечує персоналізацію рекомендацій. Високий відсоток рекомендацій (82.24%) свідчить про загальну задоволеність товарами.

2.1.2 Аналіз розподілу рейтингів, рекомендацій та відгуків за категоріями, департаментами та віком користувачів.

Аналіз розподілу рейтингів, рекомендацій і відгуків проводився в Power BI під час літньої виробничої практики. Дані групувалися за категоріями (Class.Name), департаментами (Department.Name) і віковими групами користувачів за допомогою інструментів Power BI для обробки та створення візуалізацій. Набір даних містить 23 486 відгуків про жіночий одяг, включаючи такі змінні, як вік, рейтинг, рекомендації, категорії та департаменти товарів.

Категорія «Dresses» є найпопулярнішою, містячи 6 319 одиниць товарів, що становить найбільшу кількість відгуків. Категорії «Knits» і «Blouses» також мають значну кількість відгуків, тоді як «Chemises» є найменш представленою (лише 1 одиниця). Департаменти «Tops» і «Dresses» лідирують за кількістю позитивних відгуків, тоді як «Intimate», «Jackets» і «Trend» мають меншу активність. Наприклад, департамент «Tops» включає популярні категорії, такі як «Blouses» і «Knits», які отримують високі оцінки.

Аналіз у Power BI показав, що категорія «Socks» має найвищий середній рейтинг (близько 5), тоді як «Chemises» і «Trend» мають найнижчі середні рейтинги. Розподіл рейтингів, візуалізований за допомогою кругової діаграми, демонструє, що 55.91% відгуків мають оцінку 5, 21.62% — 4, 12.22% — 3, 6.66% — 2 і 3.59% — 1. Це вказує на переважання позитивних оцінок у наборі даних. Рекомендації (Recommended.IND = 1) становлять 82.24% відгуків, що свідчить про високу задоволеність користувачів. Категорії «Casual Bottoms» і «Chemises» мають найвищий відсоток рекомендацій (близько 100%), тоді як «Trend» — найнижчий. Гістограми, створені в Power BI, підтвердили, що «Dresses» і «Knits» отримують найбільше позитивних відгуків, тоді як «Casual Bottoms» і «Chemises» — найменше.

Аналіз за віком, виконаний у Power BI, показав, що пік кількості позитивних відгуків припадає на вік 39 років (1 269 відгуків), а основна аудиторія зосереджена у віковій групі 30–50 років. Користувачі віком 30–39 років

залишають найбільше відгуків, тоді як після 50 років активність знижується. Візуалізації, такі як гістограми та кругові діаграми, створені в Power BI, ілюструють розподіл відгуків, рейтингів і рекомендацій, допомагаючи виявити популярні категорії та вікові групи.

Результати аналізу можуть бути використані для оптимізації асортименту товарів, наприклад, шляхом збільшення запасів популярних категорій («Dresses», «Knits») і перегляду пропозицій у менш популярних категоріях («Chemises», «Trend»). Також дані дозволяють налаштувати маркетингові стратегії, зосередивши їх на цільовій аудиторії 30–50 років, та покращити якість товарів у категоріях із низькими рейтингами, таких як «Trend».

2.1.3 Визначення ключових показників популярності товарів (рейтинг, кількість позитивних відгуків, рекомендації)

Ключовими показниками популярності товарів у рекомендаційній системі обрано середній рейтинг, кількість позитивних відгуків (`Positive.Feedback.Count`) і частку рекомендацій (`Recommended.IND = 1`). Середній рейтинг обчислюється за допомогою SQL-запиту, який групує відгуки за `Clothing.ID` і визначає середнє значення поля `Rating`. Наприклад, категорія «Socks» має середній рейтинг близько 5, тоді як «Dresses» і «Knits» мають рейтинги в діапазоні 4–5. Цей показник є основою для формування списку «Топ 5» товарів, який відображається в інтерфейсі системи. Високий рейтинг свідчить про якість товару і його привабливість для користувачів.

Кількість позитивних відгуків, представлена полем `Positive.Feedback.Count`, відображає число схвальних оцінок відгуку іншими користувачами. Товари з високим значенням цього показника, наприклад, у категорії «Knits», вважаються популярними, оскільки їхні відгуки сприймаються як корисні. SQL-запит для підрахунку позитивного зворотного зв'язку показав, що товари з департаменту «Tops» мають найбільшу кількість таких оцінок. Частка рекомендацій розраховується як відсоток відгуків із `Recommended.IND = 1` для кожного товару. У наборі даних 82.24% відгуків є позитивними рекомендаціями, причому категорії «Casual Bottoms» і «Chemises» досягають майже 100%.

Ці показники інтегровано в рекомендаційну систему через FastAPI-ендпоінт, який повертає товари з рейтингом ≥ 4 і `Recommended.IND = 1`, з можливістю фільтрації за департаментом. Візуалізації, такі як кругові діаграми, ілюструють розподіл позитивних відгуків і рекомендацій за категоріями, що допомагає виявити найпопулярніші товари. Наприклад, діаграма показує, що «Dresses» становлять значну частку позитивних відгуків. Обмеженням системи є відсутність аналізу текстових відгуків (`Review.Text`), що могло б додати глибину

оцінки популярності. Показники використовуються для персоналізації рекомендацій, зокрема для фільтрації за віком і категоріями, що підвищує релевантність пропозицій.

2.2 Реалізація рекомендаційної функціональності

2.2.1 Рекомендації товарів на основі віку користувачів

Рекомендації товарів на основі віку користувачів реалізовано у веб-додатку за допомогою FastAPI-ендпоінта `/age-recommendations`, який обробляє запит із введеним віком користувача. Поле `Age` із набору даних «Women's E-Commerce Clothing Reviews» використовується для сегментації користувачів шляхом фільтрації відгуків у діапазоні віку ± 5 років від введеного значення. Наприклад, для віку 39 років система аналізує відгуки користувачів віком від 34 до 44 років. Це дозволяє пропонувати товари, популярні серед певної вікової групи. Ендпоінт повертає до 7 товарів, відсортованих за кількістю відгуків у вибраному віковому діапазоні та середнім рейтингом. Реалізація базується на SQL-запиті з використанням Common Table Expression (CTE) для ефективного підрахунку відгуків .

SQL-запит створює тимчасову таблицю (`age_specific_cte`), яка містить кількість відгуків для кожного товару (`clothing_id`) у заданому віковому діапазоні. Основний запит об'єднує цю таблицю з даними про товари, додаючи інформацію про категорію (`class_name`), середній рейтинг (`rating`) і загальну кількість відгуків (`positive_feedback_count`). Товари, які не мають відгуків у вибраному діапазоні, відфільтровуються умовою `HAVING`. Наприклад, для віку 30 років система може рекомендувати товари з категорії «Dresses», оскільки ця категорія має значну кількість відгуків від користувачів 25–35 років. Запит оптимізовано для швидкої обробки великих наборів даних, що підтверджується логуванням часу виконання в `main.py`. Результати відображаються в шаблоні `age_recommendations.html`, який містить слайдер із картками товарів .

Інтерфейс дозволяє користувачу ввести вік у числовому полі форми, після чого відправляється GET-запит до ендпоінта `/age-recommendations`. Сторінка відображає товари з інформацією про `clothing_id`, категорію, рейтинг і кількість

відгуків у вибраному віковому діапазоні. Наприклад, для віку 39 років, який є піком активності (1 269 відгуків), система може показати товари з високим рейтингом (≥ 4) у категорії «Knits». Обмеженням є статична природа фільтрації, яка не враховує додаткові параметри, як-от сезонність чи стиль. Крім того, відсутність аналізу текстових відгуків (review_text) обмежує глибину персоналізації. Ця функціональність забезпечує релевантні рекомендації для цільової аудиторії 30–50 років, яка становить основну частку користувачів .

2.2.2 Рекомендації товарів на основі обраного товару

Рекомендації товарів на основі обраного товару реалізовано в ендпоінті `/product/{clothing_id}` у `main.py`, який повертає схожі товари для вибраного продукту. Схожі товари визначаються за однаковою категорією (`class_name`), наприклад, якщо користувач переглядає товар із категорії «Dresses», система пропонує інші товари цієї категорії. Запит повертає до 7 схожих товарів, відсортованих за середнім рейтингом (`rating`) у порядку спадання. Це дозволяє рекомендувати найбільш популярні товари в тій самій категорії, підвищуючи ймовірність зацікавленості користувача. Реалізація використовує SQL-запит із групуванням за `clothing_id` і виключенням поточного товару з результатів.

SQL-запит для схожих товарів (`similar_products`) обирає товари з тією ж категорією (`class_name`), що й поточний продукт, і виключає його за умовою `clothing_id != product.clothing_id`. Запит повертає `clothing_id`, назву категорії, заголовок (`title`), середній рейтинг і кількість відгуків (`positive_feedback_count`). Наприклад, для товару з `clothing_id = 1080` (категорія «Dresses») система може рекомендувати інші сукні з рейтингом ≥ 4 , такі як `clothing_id = 1077`. Результати відображаються в шаблоні `product.html` у секції «Схожі товари», яка містить слайдер із картками товарів. Картки включають зображення, `clothing_id`, категорію, рейтинг і кількість відгуків, із можливістю переходу до сторінки кожного товару.

Інтерфейс секції «Схожі товари» дозволяє користувачу переглядати рекомендації за допомогою стрілок навігації, реалізованих через JavaScript-функцію `scrollSlider`. Обмеженням є те, що рекомендації базуються лише на категорії, без урахування інших факторів, таких як стиль чи колір, через відсутність відповідних даних у наборі. Крім того, система не аналізує текстові відгуки для визначення схожості, що могло б покращити точність. Функціональність інтегрована в сторінку товару, забезпечуючи зручний перегляд схожих продуктів, що сприяє підвищенню залученості користувачів.

Наприклад, категорія «Тops» часто пропонує популярні товари з високим рейтингом, що відповідає попиту аудиторії.

2.2.3 Інтеграція рекомендаційної функціональності у веб-сайт

Рекомендаційна функціональність інтегрована у веб-сайт за допомогою фреймворку FastAPI, який обробляє запити до даних і забезпечує динамічне відображення через Jinja2-шаблони. Основні ендпоінти включають `/age-recommendations` для рекомендацій на основі віку користувача та `/product/{clothing_id}` для відображення схожих товарів на сторінці конкретного продукту. Ці ендпоінти використовують SQL-запити для фільтрації даних із набору «Women's E-Commerce Clothing Reviews» (23 486 записів), повертаючи товари з високим рейтингом і рекомендаціями. Інтерфейс веб-сайту виконаний у зеленій кольоровій гамі (#2E8B57) із використанням шрифтів Roboto для основного тексту та Great Vibes для заголовка «Elizabeth's Elegance», що створює естетичний вигляд (Рисунок 2.1 – головна сторінка сайту). Статичні файли, такі як CSS (`style.css`, `prod.css`) і зображення, монтуються через StaticFiles у `main.py`, забезпечуючи швидкий доступ до ресурсів.

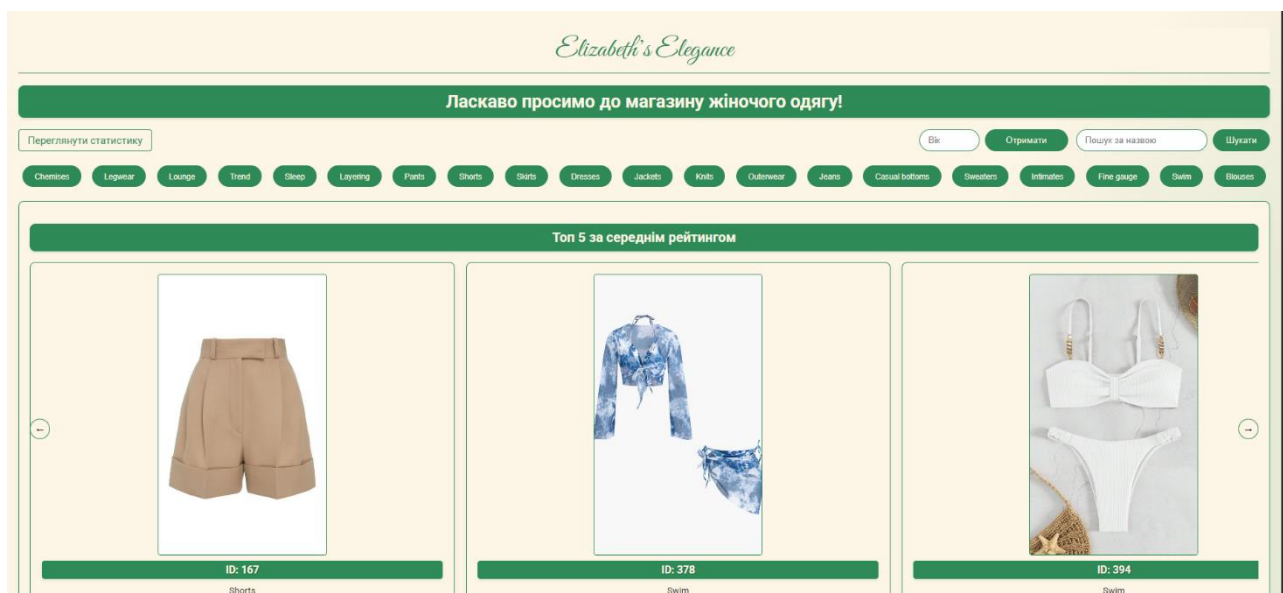


Рисунок 2.1 – Головна сторінка сайту

Сторінка рекомендацій за віком, реалізована через шаблон `age_recommendations.html`, містить форму для введення віку користувача, яка відправляє GET-запит до ендпоінта `/age-recommendations`. Користувач вводить вік у числове поле, після чого система повертає до семи товарів, популярних

серед користувачів у діапазоні ± 5 років від вказаного віку. Наприклад, для віку 39 років можуть відображатися товари з категорії «Dresses» із рейтингом ≥ 4 , оскільки ця вікова група є найактивнішою (1 269 відгуків). Результати представлені у вигляді слайдера з картками товарів, що включають `clothing_id`, категорію, рейтинг і кількість відгуків у віковому діапазоні. Навігація слайдером здійснюється за допомогою стрілок, реалізованих через JavaScript-функцію `scrollSlider` (Рисунок 2.2 – сторінка рекомендацій за віком). Пошукова панель і кнопки категорій у верхній частині сторінки забезпечують швидкий доступ до інших функцій сайту.



Рисунок 2.2 – Сторінка рекомендацій за віком

Сторінка товару, реалізована через `product.html`, відображає деталі продукту та секцію «Схожі товари», яка містить рекомендації на основі категорії (`class_name`). Наприклад, для товару з категорії «Knits» система пропонує інші товари цієї категорії з високим рейтингом, відсортовані за спаданням. Секція «Схожі товари» також використовує слайдер із навігаційними стрілками, що покращує взаємодію з користувачем. Фільтри на сторінці товару дозволяють налаштувати відгуки за віком (за допомогою слайдера `noUiSlider`), рейтингом (чекбоксы для 1–5 зірок) і рекомендаціями (`Recommended.IND`), що підвищує персоналізацію (Рисунок 2.3 – сторінка товару з фільтрами). Наприклад,

користувач може відфільтрувати відгуки від осіб 30–50 років із рейтингом 5, що відповідає основній аудиторії сайту.

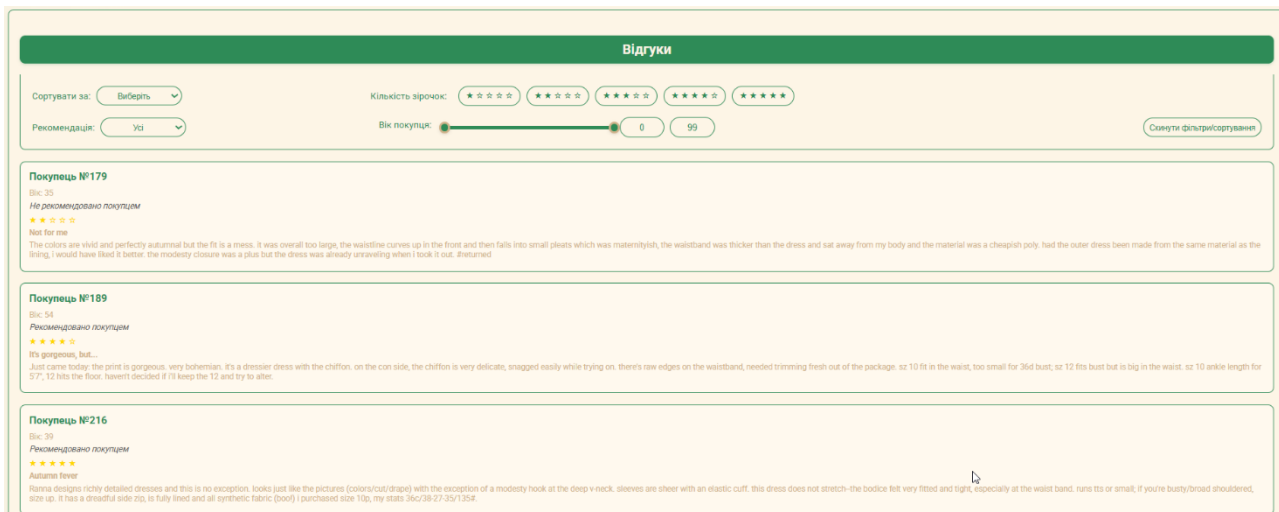


Рисунок 2.3 – Сторінка товару з фільтрами

Інтерактивні елементи для аналізу даних реалізовані на сторінці /category-stats (category_stats.html) (Рисунок 2.4 – Сторінка з інтерактивною статистикою), яка дозволяє користувачам досліджувати статистику товарів і відгуків із набору даних «Women's E-Commerce Clothing Reviews».

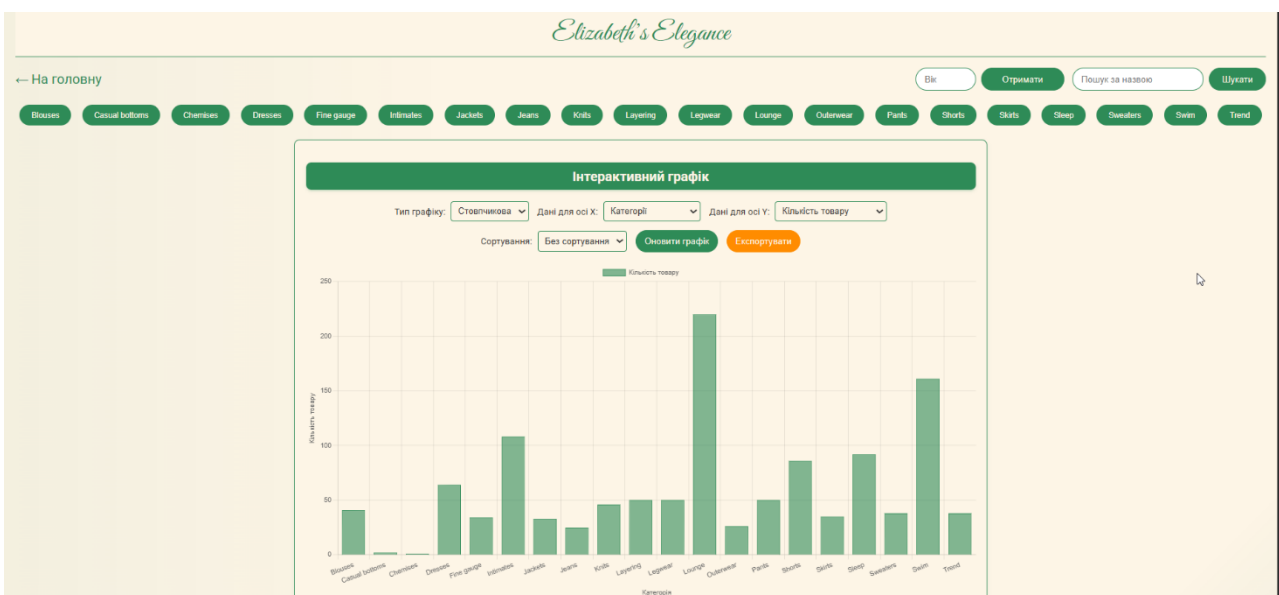


Рисунок 2.4 – Сторінка з інтерактивною статистикою

Сторінка містить два типи графіків: інтерактивний графік, побудований за допомогою бібліотеки Chart.js, і статичні графіки, які попередньо підготовлені на основі даних із бази (Рисунок 2.5 – Статичні графіки).

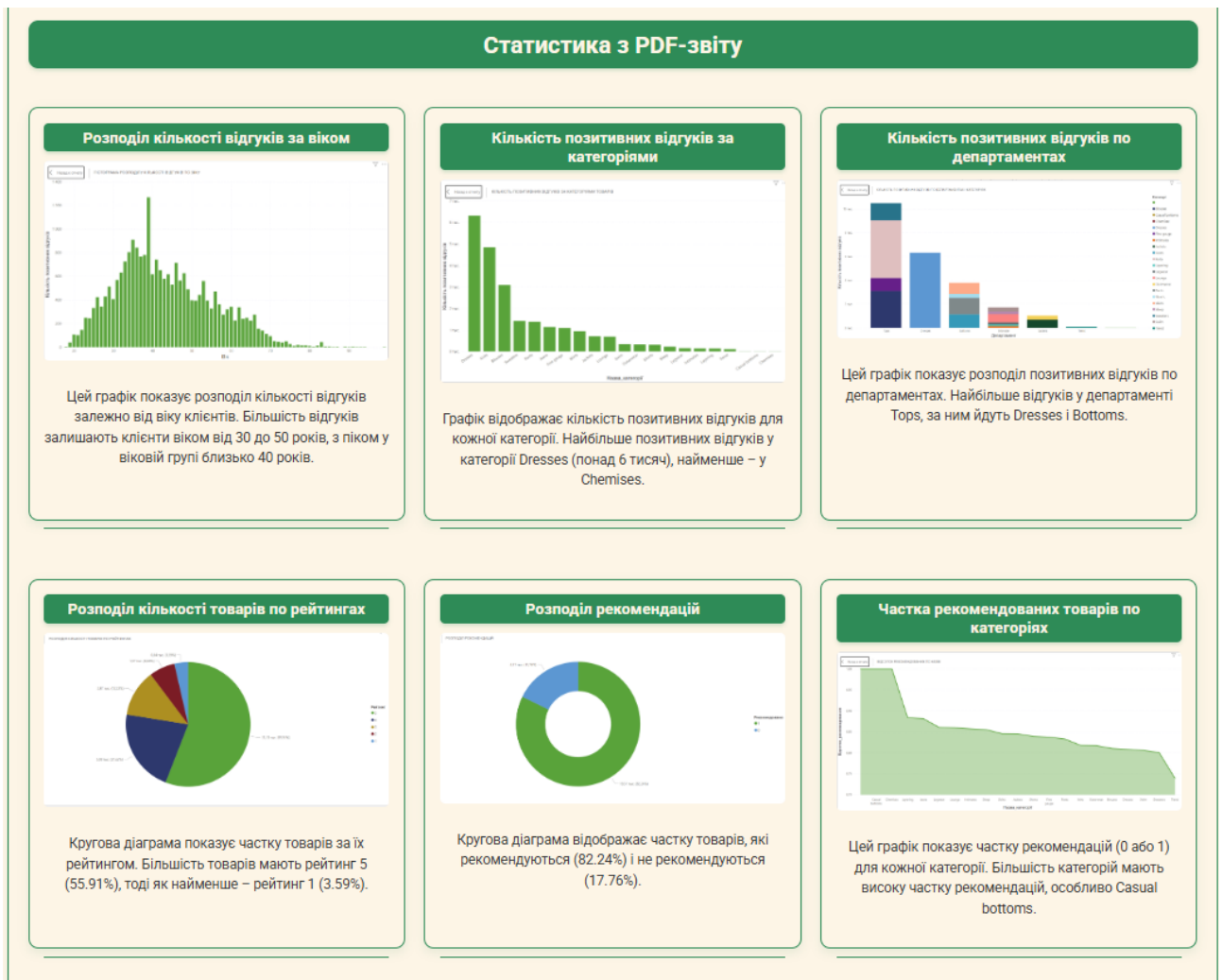


Рисунок 2.5 – Статичні графіки

Сім статичних графіків для підтримки аналізу рекомендацій включають: гістограму відгуків за віком, позитивні відгуки за категоріями, позитивні відгуки за департаментами, розподіл рейтингів, розподіл рекомендацій, частку рекомендацій за категоріями і середній рейтинг за категоріями. Ці графіки зберігаються в папці static і відображаються в шаблоні category_stats.html, допомагаючи користувачам зрозуміти популярність товарів.

2.3 Проєктування веб-сайту з рекомендаційною системою

2.3.1 Вимоги до веб-сайту та структура інтерфейсу (головна сторінка, фільтри, топ-товарів)

Веб-сайт «Elizabeth's Elegance» створений для продажу жіночого одягу з інтегрованою рекомендаційною системою, яка базується на відгуках користувачів із набору даних «Women's E-Commerce Clothing Reviews». Основна мета сайту — забезпечити зручний інтерфейс для пошуку товарів, аналізу відгуків та отримання рекомендацій. Ключові вимоги включають адаптивність для різних пристроїв (десктопи, планшети, смартфони), швидке завантаження сторінок, інтуїтивну навігацію, підтримку української мови та естетичний дизайн, що відображає бренд магазину. Адаптивність реалізовано через медіа-запити в CSS-файлі `style.css`, які змінюють розташування елементів залежно від розміру екрана. Наприклад, на екранах до 480 пікселів ширина пошукового поля зменшується до 70% екрана, а навігація стає вертикальною. Ось приклад CSS-коду для адаптивності:

```
/* Адаптивність пошукового поля та навігації в style.css */
.search-bar {
  width: 250px;
  padding: 10px;
  border: 2px solid #2E8B57;
  border-radius: 5px;
}

.nav-menu {
  display: flex;
  gap: 20px;
}

@media (max-width: 480px) {
  .search-bar {
    width: 70%;
  }
  .nav-menu {
```

```

        flex-direction: column;
        align-items: center;
    }
}

```

Швидке завантаження забезпечується використанням FastAPI для асинхронних запитів до бази даних, що зменшує затримки при отриманні списків товарів чи відгуків. Застосування SQLAlchemy для ефективною обробки даних і Jinja2 для динамічного рендерингу шаблонів сприяє швидкому відображенню контенту на веб-сайті. Дизайн сайту виконано в смарагдово-бежевій палітрі: основні кольори — смарагдовий (#2E8B57) та бежевий (#FDF5E6), які створюють елегантний і професійний вигляд. Шрифти Roboto (для тексту) і Great Vibes (для логотипу та заголовків) підкреслюють бренд, хоча для сумісності в CSS визначено резервні шрифти, як-от Arial. Логотип «Elizabeth's Elegance» розташовано у верхній частині сторінки, виконаний шрифтом Great Vibes розміром 48 пікселів із смарагдовим кольором на бежевому фоні.

Навігаційна панель, реалізована в index.html, містить посилання на головну сторінку, сторінку статистики (/category-stats) та форму рекомендацій за віком (age_recommendations.html). Кнопки навігації мають заокруглені краї (властивість border-radius: 5px) і змінюють колір фону на світліший смарагдовий (#3CB371) при наведенні курсору. Ось приклад стилізації кнопок навігації з style.css:

```

/* Стилізація кнопок навігації в style.css */
.nav-menu a {
    padding: 10px 20px;
    background-color: #2E8B57;
    color: #FFFFFF;
    text-decoration: none;
    border-radius: 5px;
    transition: background-color 0.3s ease;
}

.nav-menu a:hover {
    background-color: #3CB371;
}

```

Головна сторінка (index.html) складається з кількох функціональних блоків: логотипу, навігаційної панелі, пошукової системи, категорій, фільтрів, блоків «Топ-5» та списку товарів. Ось спрощений приклад HTML-структури головної сторінки з index.html, який показує основні блоки:

```
<!-- Фрагмент index.html із основними блоками -->
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Elizabeth's Elegance</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h1 class="logo">Elizabeth's Elegance</h1>
    <nav class="nav-menu">
      <a href="/index">Головна</a>
      <a href="/category-stats">Статистика</a>
      <a href="/age-recommendations">Рекомендації за віком</a>
    </nav>
    <input type="text" class="search-bar" placeholder="Пошук
товарів...">
  </header>
  <main>
    <section class="categories">
      <button onclick="filterCategory('Dresses')">Dresses</button>
      <button onclick="filterCategory('Tops')">Tops</button>
      <button onclick="filterCategory('Bottoms')">Bottoms</button>
    </section>
    <section class="top-5">
      <h2>Топ-5 за рейтингом</h2>
      <div id="top-rated-slider" class="slider">
        <!-- Товари генеруються через Jinja2 -->
        {% for item in top Rated %}
        <div class="item">
          
          <p>{{ item.class_name }} (Рейтинг: {{ item.avg_rating
| round(1) }})</p>
        </div>
```

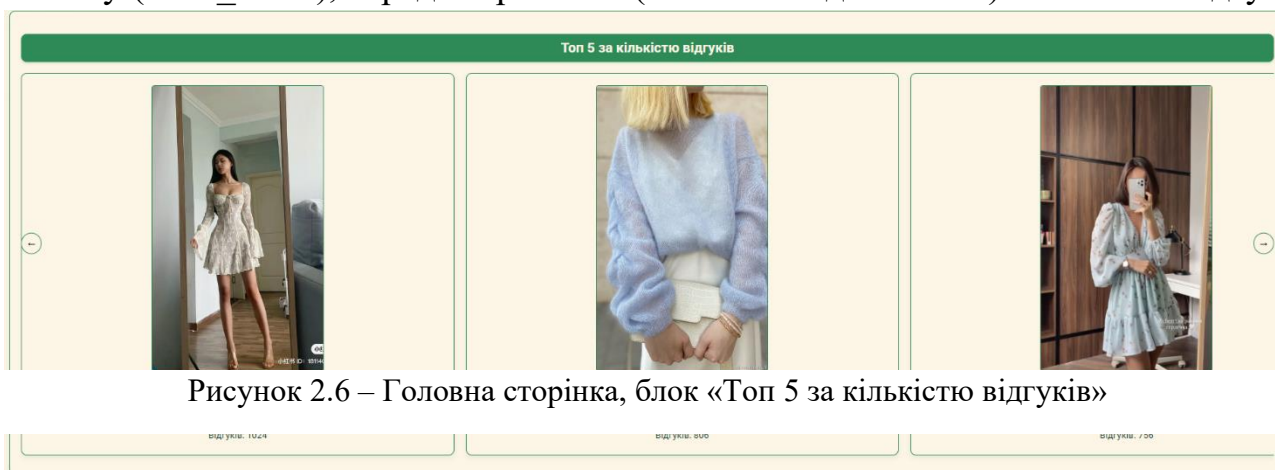
```

        {% endfor %}
    </div>
    <button onclick="scrollSlider('left', 'top-rated-
slider')"></button>
    <button onclick="scrollSlider('right', 'top-rated-
slider')"></button>
</section>
<aside class="filter-panel">
    <!-- Фільтри додано в попередньому розділі -->
</aside>
<section class="product-grid">
    <!-- Список товарів із пагінацією -->
</section>
</main>
<script src="script.js"></script>
</body>
</html>

```

Пошукова система дозволяє вводити назву товару чи ключові слова, а результати обробляються ендпоінтом /search у main.py, який виконує SQL-запит до таблиці відгуків. Наприклад, пошук за словом «Dress» повертає товари з категорії «Dresses» із відповідними class_name. Блок категорій відображає кнопки, як-от «Dresses», «Tops», «Bottoms», які генеруються динамічно через шаблон Jinja2 і фільтрують товари за параметром category.

Блоки «Топ-5 за середнім рейтингом» і «Топ-5 за кількістю відгуків» показують найпопулярніші товари, відібрані за критеріями рейтингу (≥ 3) та кількості відгуків. Кожен елемент у блоці містить зображення (clothing_id.jpg), назву (class_name), середній рейтинг (з точністю до 1 знака) і кількість відгуків.



Наприклад, товар із `clothing_id=1078` може мати рейтинг 4.5 і 120 відгуків.

Дані для цих блоків отримуються через SQL-запити в `main.py`, як-от:

```
# Топ-5 товарів за середнім рейтингом
top_by_average_score = (
    db.query(
        Product.clothing_id,
        Product.class_name,
        func.avg(Product.rating).label("average_rating"),
        func.count(Product.clothing_id).label("review_count"),
    )
    .group_by(Product.clothing_id, Product.class_name)
    .having(func.avg(Product.rating) >= 3.0)
    .order_by(func.avg(Product.rating).desc(),
func.count(Product.clothing_id).desc())
    .limit(5)
    .all()
)
```

Для прокрутки елементів у блоках «Топ-5» використовується JavaScript-функція `scrollSlider`, яка плавно переміщує товари вліво чи вправо при натисканні стрілок. Ось приклад цієї функції:

```
// Функція для прокрутки блоку "Топ-5" у index.html
function scrollSlider(direction, sliderId) {
    const slider = document.getElementById(sliderId);
    const scrollAmount = 300; // Кількість пікселів для прокрутки
    if (direction === 'left') {
        slider.scrollBy({ left: -scrollAmount, behavior: 'smooth' });
    } else if (direction === 'right') {
        slider.scrollBy({ left: scrollAmount, behavior: 'smooth' });
    }
}
```

Список товарів відображається у вигляді сітки: 4 товари в ряд для десктопів, 2 для планшетів і 1 для смартфонів, що реалізовано через CSS-властивість `grid-template-columns` у `style.css`. Пагінація дозволяє переглядати товари по 12, 24, 36, 48, 60 або 96 на сторінці, обробляючись параметром `page` у `main.py`. Фільтри розташовані в бічній панелі та включають вибір підрозділу (`division`), департаменту (`department`), категорії (`category`) та діапазону рейтингу.

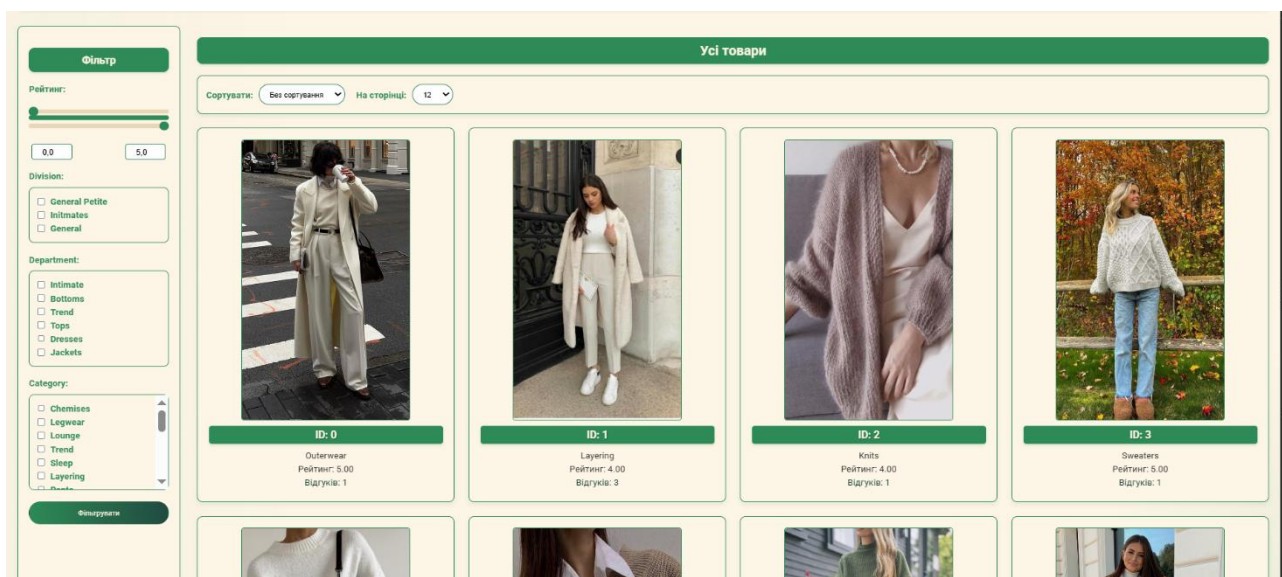


Рисунок 2.7 – Головна сторінка, блок «Усі товари»

Дизайн фільтрів відповідає стилю сайту: смарагдові кнопки з білим текстом і заокругленими краями.

Сайт підтримує українську мову для всіх елементів інтерфейсу, включаючи підписи кнопок і повідомлення про помилки. Адаптивність забезпечує коректне відображення на малих екранах: наприклад, бічна панель фільтрів ховається в меню-гамбургер для екранів до 768 пікселів. Обмеження інтерфейсу включають відсутність автодоповнення в пошуковій системі та статичне оновлення сторінки при зміні фільтрів. Загалом, структура інтерфейсу продумана для зручності користувача, поєднуючи естетику, функціональність і швидкодію.

2.3.2 Реалізація фільтрів та сортування товарів

Система фільтрів і сортування на веб-сайті «Elizabeth's Elegance» розроблена для спрощення пошуку товарів і аналізу відгуків на основі даних із «Women's E-Commerce Clothing Reviews». На головній сторінці (index.html) фільтри розташовані в бічній панелі та дозволяють вибирати підрозділ (division, наприклад, «General»), департамент (department, наприклад, «Tops»), категорію (category, наприклад, «Dresses») і діапазон середнього рейтингу (від 1 до 5). Фільтри реалізовані як HTML-форма з методом GET, яка формує URL-параметри, оброблювані ендпоінтом / у main.py. Ось приклад форми фільтрів:

```
<!-- Форма фільтрів у index.html -->
<form method="GET" action="/index" class="filter-panel">
  <label for="division">Підрозділ:</label>
  <select name="division" id="division">
    <option value="">Усі</option>
    <option value="General">General</option>
    <option value="Petite">Petite</option>
  </select>

  <label for="department">Департамент:</label>
  <select name="department" id="department">
    <option value="">Усі</option>
    <option value="Tops">Tops</option>
    <option value="Dresses">Dresses</option>
    <option value="Bottoms">Bottoms</option>
  </select>

  <label for="category">Категорія:</label>
  <select name="category" id="category">
    <option value="">Усі</option>
    <option value="Dresses">Dresses</option>
    <option value="Knits">Knits</option>
    <option value="Blouses">Blouses</option>
```

```

</select>

<label for="min-rating">Мін. рейтинг:</label>
<input type="range" id="min-rating" name="min_rating" min="1" max="5"
value="1" oninput="updateRange()">
<span id="min-rating-label">1</span>

<label for="max-rating">Макс. рейтинг:</label>
<input type="range" id="max-rating" name="max_rating" min="1" max="5"
value="5" oninput="updateRange()">
<span id="max-rating-label">5</span>

<button type="submit">Застосувати</button>
</form>

```

Наприклад, вибір категорії «Dresses» і рейтингу від 4 до 5 створює запит `/index?category=Dresses&min_rating=4&max_rating=5`. Діапазон рейтингу задається через два повзунки (`input[type="range"]`), які синхронізуються JavaScript-функцією `updateRange` для відображення поточних значень. Ось приклад цієї функції:

```

// Функція для синхронізації повзунків діапазону рейтингу в index.html
function updateRange() {
  const minRating = document.getElementById('min-rating').value;
  const maxRating = document.getElementById('max-rating').value;
  document.getElementById('min-rating-label').textContent = minRating;
  document.getElementById('max-rating-label').textContent = maxRating;
  // Забезпечення, що minRating не більше maxRating
  if (parseInt(minRating) > parseInt(maxRating)) {
    document.getElementById('min-rating').value = maxRating;
    document.getElementById('min-rating-label').textContent = maxRating;
  }
}

```

Дизайн фільтрів відповідає стилю сайту: смарагдові кнопки, заокруглені краї (border-radius: 5px), білий текст (color: #FFFFFF). Повзунки для рейтингу мають смарагдовий колір треку (background: #2E8B57) і білу ручку (background: #FFFFFF). Ось приклад CSS-стилів із style.css:

```
/* Стилізація фільтрів і повзунків у style.css */
```

```
.filter-panel {  
    background-color: #FDF5E6;  
    padding: 20px;  
    border-radius: 10px;  
}  
  
.filter-panel label {  
    color: #2E8B57;  
    font-weight: bold;  
}  
  
.filter-panel select, .filter-panel button {  
    background-color: #2E8B57;  
    color: #FFFFFF;  
    border: none;  
    border-radius: 5px;  
    padding: 8px;  
    margin: 5px 0;  
}  
  
.filter-panel input[type="range"] {  
    -webkit-appearance: none;  
    width: 100%;  
    background: #2E8B57;  
    border-radius: 5px;  
    height: 8px;  
}
```

```
.filter-panel input[type="range"]::-webkit-slider-thumb {
  -webkit-appearance: none;
  width: 20px;
  height: 20px;
  background: #FFFFFF;
  border: 2px solid #2E8B57;
  border-radius: 50%;
  cursor: pointer;
}
```

На сторінці окремого товару (product.html) фільтри дозволяють аналізувати відгуки за віком, рейтингом і статусом рекомендації (recommended). Слайдер віку, побудований за допомогою бібліотеки noUiSlider, дає змогу вибирати діапазон (наприклад, 30–40 років), а чекбокси дозволяють фільтрувати відгуки за рейтингом (1–5 зірок) і статусом рекомендації (рекомендується чи ні). Ось приклад HTML-коду для фільтрів на сторінці товару:

```
<!-- Фільтри на сторінці товару в product.html -->
<div class="review-filters">
  <label>Вік:</label>
  <div id="age-slider"></div>
  <p>Діапазон віку: <span id="age-range">30 - 40</span></p>

  <label>Рейтинг:</label>
  <label><input type="checkbox" name="rating" value="1"> 1
зірка</label>
  <label><input type="checkbox" name="rating" value="2"> 2
зірки</label>
  <label><input type="checkbox" name="rating" value="3"> 3
зірки</label>
  <label><input type="checkbox" name="rating" value="4"> 4
зірки</label>
  <label><input type="checkbox" name="rating" value="5"> 5
зірок</label>
```

```

    <label>Рекомендація:</label>
    <label><input type="checkbox" name="recommended" value="1">
Рекомендується</label>
    <label><input type="checkbox" name="recommended" value="0"> Не
рекомендується</label>

    <button onclick="applyReviewFilters()">Застосувати</button>
</div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/noUiSlider/15.7.1/nouislider.
min.js"></script>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/noUiSlider/15.7.1/nouislider
.min.css">
<script>
    const slider = document.getElementById('age-slider');
    noUiSlider.create(slider, {
        start: [30, 40],
        connect: true,
        range: { 'min': 18, 'max': 80 }
    });
    slider.noUiSlider.on('update', function (values) {
        document.getElementById('age-range').textContent =
values.map(Math.round).join(' - ');
    });
</script>

```

Наприклад, користувач може вибрати відгуки для товару `clothing_id=1078` від жінок віком 35–45 років із рейтингом 5. Ці фільтри обробляються ендпоінтом `/product/{clothing_id}` із параметрами `age_min`, `age_max`, `rating`, `recommended`. Ось приклад обробки:

```

# Ендпоінт для фільтрації відгуків у main.py
@app.get("/index")
def filter_reviews(

```

```

division: str | None = Query(None),
department: str | None = Query(None),
category: str | None = Query(None),
min_rating: float | None = Query(None),
max_rating: float | None = Query(None),
db: Session = Depends(get_db)
):
    """
    Ендпоінт для фільтрації відгуків за параметрами (без сортування та
пагінації).
    Повертає відфільтровані товари для відображення.
    """
    query = db.query(
        Product.clothing_id,
        func.max(Product.class_name).label("class_name"),
        func.avg(Product.rating).label("avg_rating"),
        func.count(Product.clothing_id).label("review_count")
    ).group_by(Product.clothing_id)

    if division:
        query = query.having(func.max(Product.division_name) == division)
    if department:
        query = query.having(func.max(Product.department_name) ==
department)
    if category:
        query = query.having(func.max(Product.class_name) == category)
    if min_rating is not None:
        query = query.having(func.avg(Product.rating) >= min_rating)
    if max_rating is not None:
        query = query.having(func.avg(Product.rating) <= max_rating)

    items = query.all()

```

```
return templates.TemplateResponse("index.html", {"request": {},
"items": items})
```

Сортування товарів доступне на головній сторінці через випадуючий список у блоці `sorting-bar`, який дозволяє вибирати порядок за рейтингом (`rating_asc`, `rating_desc`), кількістю відгуків (`reviews_asc`, `reviews_desc`) або ID (`id_asc`, `id_desc`). Ось приклад HTML-коду для випадуючого списку сортування:

```
<!-- Випадуючий список для сортування в index.html -->
<div class="sorting-bar">
  <label for="sort">Сортування:</label>
  <select name="sort" id="sort" onchange="this.form.submit()">
    <option value="rating_desc">За рейтингом (спадання)</option>
    <option value="rating_asc">За рейтингом (зростання)</option>
    <option value="reviews_desc">За відгуками (спадання)</option>
    <option value="reviews_asc">За відгуками (зростання)</option>
    <option value="id_desc">За ID (спадання)</option>
    <option value="id_asc">За ID (зростання)</option>
  </select>
  <label for="items_per_page">Товарів на сторінці:</label>
  <select name="items_per_page" id="items_per_page"
onchange="this.form.submit()">
    <option value="12">12</option>
    <option value="24">24</option>
    <option value="36">36</option>
    <option value="48">48</option>
    <option value="60">60</option>
    <option value="96">96</option>
  </select>
</div>
```

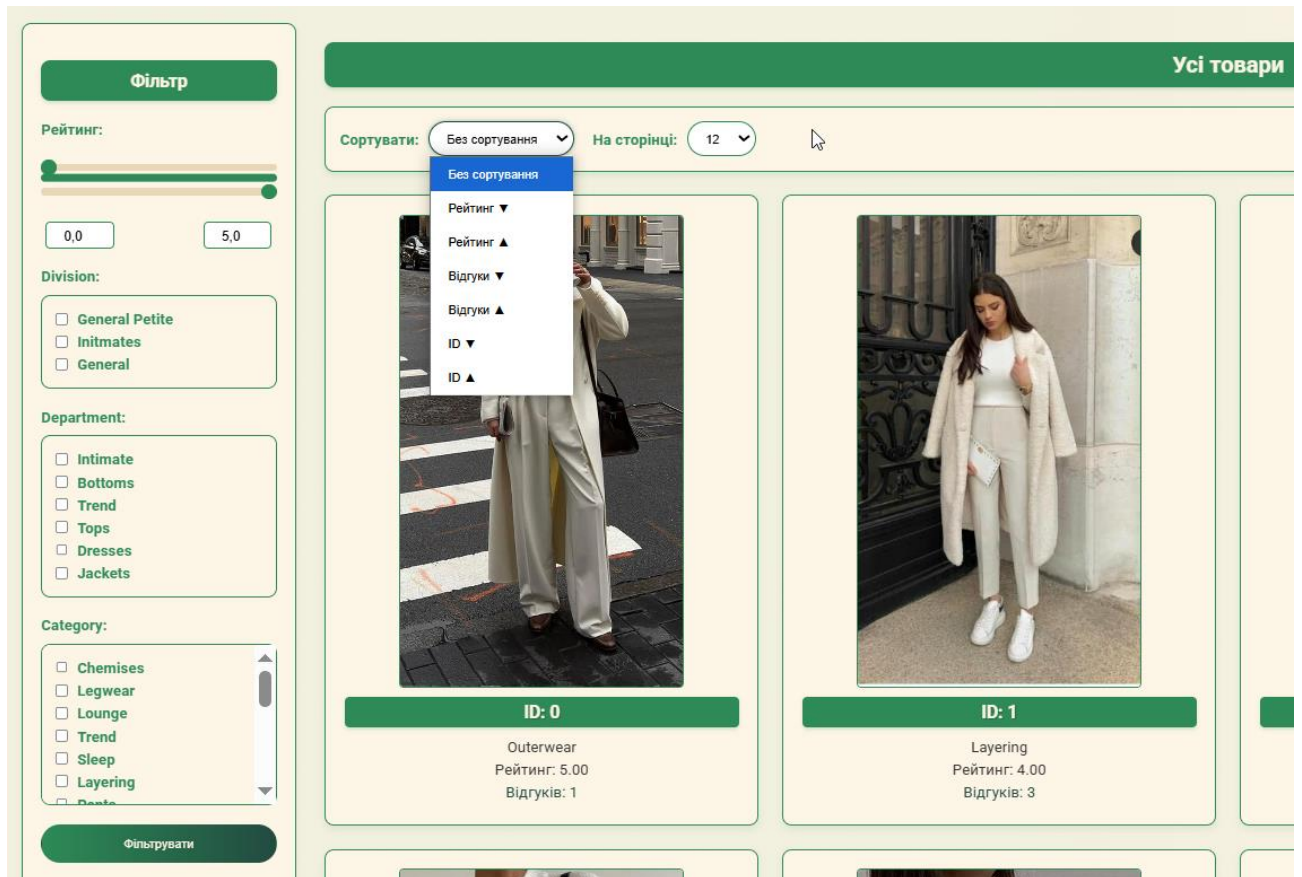


Рисунок 2.8 - Блок фільтрації та сортування на головній сторінці

Сортування реалізовано через SQL-запити з умовою ORDER BY у main.py. Наприклад, сортування за rating_desc повертає товари з найвищим рейтингом першими, як-от clothing_id=1078 із рейтингом 4.5. Кількість товарів на сторінці можна налаштувати (12, 24, 36, 48, 60, 96), що обробляється параметром items_per_page. Пагінація реалізована через параметр page і SQL-запити з LIMIT і OFFSET. Ось приклад ендпоінту / у main.py, який обробляє фільтри, сортування і пагінацію:

```
# Ендпоінт для головної сторінки з фільтрами, сортуванням і пагінацією в main.py
@app.get("/", response_class=HTMLResponse)
async def get_home(
    request: Request,
    division: list[str] | None = Query(None),
    department: list[str] | None = Query(None),
    category: list[str] | None = Query(None),
    min_rating: float | None = Query(None),
    max_rating: float | None = Query(None),
    sort_by: str | None = Query(None),
    items_per_page: str = Query("12"),
    page: int = Query(1, ge=1),
```

```

search: str | None = Query(None),
db: Session = Depends(get_db),
):
    items_per_page_value = get_items_per_page(items_per_page)
    base_query = db.query(
        Product.clothing_id,
        func.max(Product.class_name).label("class_name"),
        func.max(Product.title).label("title"),
        func.max(Product.division_name).label("division_name"),
        func.max(Product.department_name).label("department_name"),
        func.avg(Product.rating).label("rating"),
        func.count(Product.clothing_id).label("positive_feedback_count"),
    ).group_by(Product.clothing_id)
    if division:
        base_query =
base_query.having(func.max(Product.division_name).in_(division))
    if department:
        base_query =
base_query.having(func.max(Product.department_name).in_(department))
    if category:
        base_query =
base_query.having(func.max(Product.class_name).in_(category))
    if min_rating is not None:
        base_query = base_query.having(func.avg(Product.rating) >=
min_rating)
    if max_rating is not None:
        base_query = base_query.having(func.avg(Product.rating) <=
max_rating)
    sort_options = {
        "rating_desc": func.avg(Product.rating).desc(),
        "rating_asc": func.avg(Product.rating).asc(),
        "reviews_desc": func.count(Product.clothing_id).desc(),
        "reviews_asc": func.count(Product.clothing_id).asc(),
        "id_desc": Product.clothing_id.desc(),
        "id_asc": Product.clothing_id.asc(),
    }
    base_query = base_query.order_by(sort_options.get(sort_by,
Product.clothing_id.asc()))
    total_products = base_query.count()
    page, offset = adjust_page_and_offset(page, total_products,
items_per_page_value)
    products =
base_query.offset(offset).limit(items_per_page_value).all()
    template = env.get_template("index.html")
    return HTMLResponse(content=template.render(request=request,
products=products))

```

Адаптивність забезпечує коректне відображення: на екранах до 768 пікселів фільтри ховаються в розкритому меню, а повзунки зменшуються. Обмеження системи включають відсутність фільтрації за текстом відгуків і

необхідність перезавантаження сторінки для застосування фільтрів. Наприклад, фільтрація за категорією «Dresses» із рейтингом ≥ 4 повертає товари, як-от `clothing_id=1078`, але не дозволяє шукати за ключовими словами у відгуках, як-от «comfortable».

Система фільтрів і сортування інтегрована з рекомендаційною системою, яка на сторінці `age_recommendations.html` пропонує товари на основі віку користувача. Наприклад, для жінок 30–40 років система рекомендує товари з високим відсотком позитивних відгуків у цій віковій групі, використовуючи SQL-запит із групуванням за `age` і `recommended`. Загалом, фільтри та сортування забезпечують гнучкий і зручний пошук, хоча асинхронне оновлення могло б покращити користувацький досвід.

2.3.3 Інтерактивні елементи для аналізу даних

Інтерактивні елементи для аналізу даних реалізовані на сторінці /category-stats (category_stats.html), яка дозволяє користувачам досліджувати статистику товарів і відгуків із набору даних «Women's E-Commerce Clothing Reviews». Сторінка містить два типи графіків: інтерактивний графік, побудований за допомогою бібліотеки Chart.js, і статичні графіки, які попередньо підготовлені на основі даних із бази. Інтерактивний графік є основним елементом і дозволяє користувачам налаштовувати тип графіку, дані для осей і сортування, що робить аналіз даних гнучким і зручним.

Інтерактивний графік підтримує п'ять типів: стовпчиковий, лінійний, точковий, круговий і радарний. Користувач може вибирати дані для осі X (categories, avgRating, avgPrice) і осі Y (itemCount, positiveFeedback, totalReviews, positiveFeedbackPercentage, avgRating) через випадаючі списки, а також сортувати дані (без сортування, за зростанням, за спаданням).

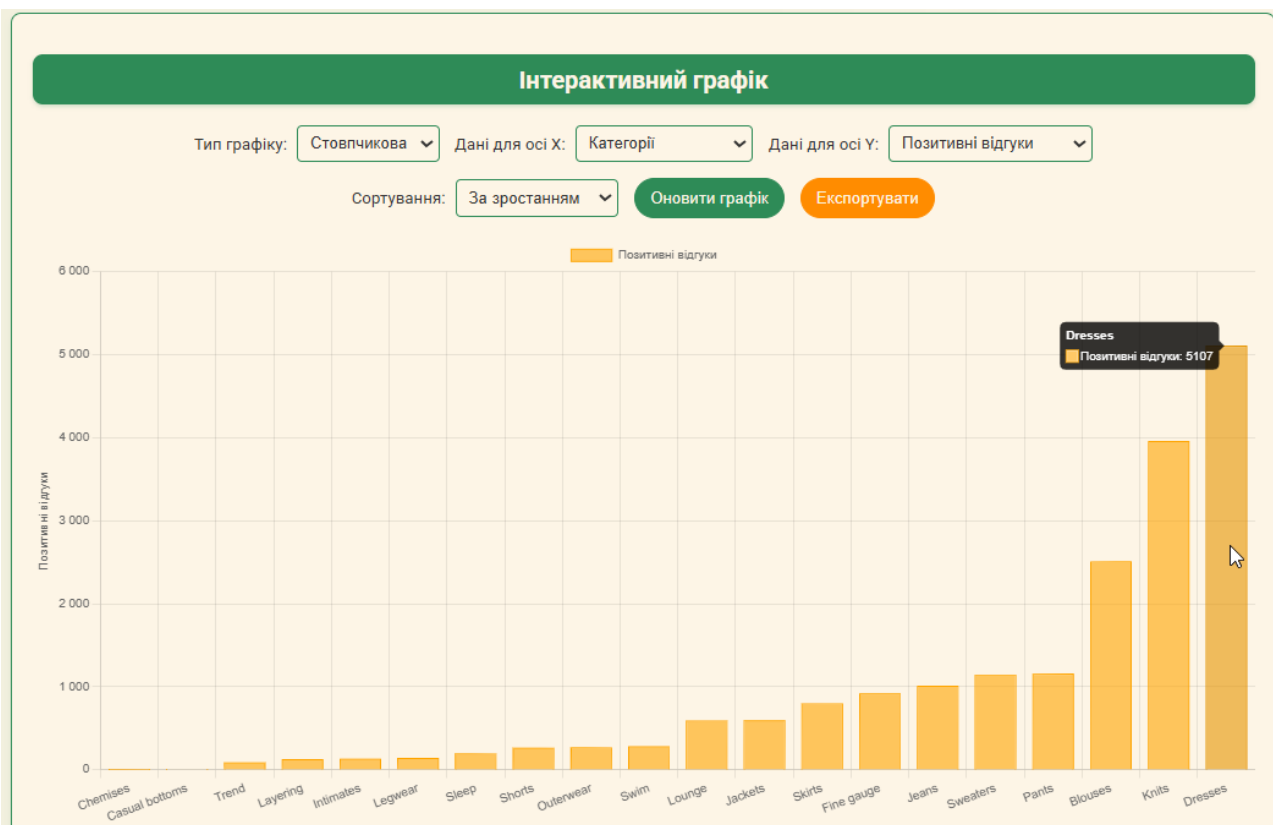


Рисунок 2.9 – Інтерактивна гістограма розподілу позитивних відгуків

Ось приклад HTML-коду для елементів керування графіком із category_stats.html:

```
<!-- Елементи керування інтерактивним графіком у category_stats.html -->
<div class="chart-controls">
  <label for="chart-type">Тип графіку:</label>
  <select id="chart-type" onchange="updateChart()">
    <option value="bar">Стовпчиковий</option>
    <option value="line">Лінійний</option>
    <option value="scatter">Точковий</option>
    <option value="pie">Круговий</option>
    <option value="radar">Радариний</option>
  </select>

  <label for="x-axis">Вісь X:</label>
  <select id="x-axis" onchange="updateChart()">
    <option value="category">Категорії</option>
    <option value="avg_rating">Середній рейтинг</option>
    <option value="avg_price">Середня ціна</option>
  </select>

  <label for="y-axis">Вісь Y:</label>
  <select id="y-axis" onchange="updateChart()">
    <option value="item_count">Кількість товарів</option>
    <option value="positive_feedback">Позитивні відгуки</option>
    <option value="total_reviews">Загальна кількість
відгуків</option>
    <option value="positive_feedback_percentage">Відсоток позитивних
відгуків</option>
    <option value="avg_rating">Середній рейтинг</option>
  </select>

  <label for="sort-order">Сортування:</label>
  <select id="sort-order" onchange="updateChart()">
```

```

    <option value="none">Без сортування</option>
    <option value="asc">За зростанням</option>
    <option value="desc">За спаданням</option>
  </select>

```

```

    <button onclick="exportChart()">Експортувати як PNG</button>
  </div>
  <canvas id="statsChart" style="max-width: 800px;"></canvas>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

```

Наприклад, користувач може побудувати стовпчиковий графік із категоріями (Dresses, Tops тощо) на осі X і кількістю товарів (itemCount) на осі Y. Сортування даних застосовується до значень осі Y. Дані для графіку отримуються асинхронно через ендпоінт /category-stats-data, який повертає JSON із агрегованими даними:

Ендпоінт для даних графіку в main.py

```

@app.get("/category-stats-data")
async def get_category_stats_data(db: Session = Depends(get_db)):
    categories = [c[0] for c in
db.query(Product.class_name).distinct().all() if c[0]]
    categories.sort()
    item_counts_query = db.query(
        Product.class_name,
func.count(func.distinct(Product.clothing_id)).label("item_count")
    ).group_by(Product.class_name).order_by(Product.class_name).all()
    item_counts = [0] * len(categories)
    for cat, count in item_counts_query:
        if cat in categories:
            item_counts[categories.index(cat)] = count
    feedback_counts_query = db.query(
        Product.class_name,
        func.count(Product.clothing_id).label("feedback_count")

```

```

        ).filter(Product.recommended_ind
1).group_by(Product.class_name).order_by(Product.class_name).all()
        positive_feedback_counts = [0] * len(categories)
        for cat, count in feedback_counts_query:
            if cat in categories:
                positive_feedback_counts[categories.index(cat)] = count
        total_reviews_query = db.query(
            Product.class_name,
            func.count(Product.clothing_id).label("total_reviews")
        ).group_by(Product.class_name).order_by(Product.class_name).all()
        total_reviews = [0] * len(categories)
        for cat, count in total_reviews_query:
            if cat in categories:
                total_reviews[categories.index(cat)] = count
        positive_feedback_percentage = [
            round((positive_feedback_counts[i] / total_reviews[i]) * 100, 2)
if total_reviews[i] > 0 else 0.0
            for i in range(len(categories))
        ]
        avg_ratings_query = db.query(
            Product.class_name,
            func.avg(Product.rating).label("avg_rating")
        ).group_by(Product.class_name).order_by(Product.class_name).all()
        avg_ratings = [0.0] * len(categories)
        for cat, avg in avg_ratings_query:
            if cat in categories:
                avg_ratings[categories.index(cat)] = round(float(avg), 2) if
avg is not None else 0.0
        return {
            "categories": categories,
            "itemCount": item_counts,
            "positiveFeedback": positive_feedback_counts,
            "totalReviews": total_reviews,
            "positiveFeedbackPercentage": positive_feedback_percentage,

```

```

    "avgRating": avg_ratings
  }

```

JavaScript-функція `updateChart` оновлює графік після зміни параметрів, а `exportChart` дозволяє завантажити його як PNG-файл. Ось приклад функції `updateChart`:

```

// Функція для оновлення інтерактивного графіку в category_stats.html
let chartInstance = null;

function updateChart() {
  const chartType = document.getElementById('chart-type').value;
  const xAxis = document.getElementById('x-axis').value;
  const yAxis = document.getElementById('y-axis').value;
  const sortOrder = document.getElementById('sort-order').value;

  // Отримання даних через асинхронний запит
  fetch('/category-stats-data')
    .then(response => response.json())
    .then(data => {
      // Сортування даних
      if (sortOrder !== 'none') {
        data.sort((a, b) => {
          if (sortOrder === 'asc') return a[yAxis] - b[yAxis];
          return b[yAxis] - a[yAxis];
        });
      }

      // Підготовка даних для графіку
      const labels = data.map(item => item[xAxis]);
      const values = data.map(item => item[yAxis]);

      // Оновлення графіку
      const ctx =
document.getElementById('statsChart').getContext('2d');
      if (chartInstance) chartInstance.destroy();

```

```

    chartInstance = new Chart(ctx, {
      type: chartType,
      data: {
        labels: labels,
        datasets: [{
          label: yAxis,
          data: values,
          backgroundColor: 'rgba(46, 139, 87, 0.6)',
          borderColor: 'rgba(46, 139, 87, 1)',
          borderWidth: 1
        }]
      },
      options: {
        scales: {
          y: { beginAtZero: true }
        }
      }
    });
  });
}

function exportChart() {
  const link = document.createElement('a');
  link.href = chartInstance.toBase64Image();
  link.download = 'chart.png';
  link.click();
}

```

Кольори графіку відповідають палітрі сайту: стовпчики заповнені смарагдовим (`rgba(46,139,87,0.6)`), а межі — темнішим смарагдовим (`rgba(46,139,87,1)`). Підписи осей автоматично адаптуються до вибраних даних, наприклад, «Кількість товарів» для `itemCount`.

Статичні графіки включають гістограму розподілу відгуків за віком, кругову діаграму рейтингів, діаграму розподілу позитивних відгуків за

категоріями, гістограму частки рекомендацій і графік середнього рейтингу. Ці графіки попередньо підготовлені та зберігаються в папці static (наприклад, age_feedback_histogram.png, rating_distribution.png тощо). Вони відображаються на сторінці /category-stats у вигляді інтерактивних елементів, які можна відкрити у модальному вікні для детального перегляду. Ось приклад коду з category_stats.html, який відповідає за відображення статичних графіків:

```
<!-- Статичні графіки з PDF -->
<div class="chart-section">
  <h2>Статистика з PDF-звіту</h2>
  <div class="pdf-charts">
    {% for chart in charts_info %}
      <div class="pdf-chart" onclick="openModal('{{ chart.filename }}',
'{{ chart.title }}', '{{ chart.description }}')">
        <h3>{{ chart.title }}</h3>
        
        <p>{{ chart.description }}</p>
      </div>
    {% endfor %}
  </div>
</div>

<!-- Модальне вікно -->
<div id="chartModal" class="modal">
  <div class="modal-content">
    <span class="close" onclick="closeModal()">×</span>
    <h2 id="modalTitle"></h2>
    <img id="modalImage" src="" alt="Chart">
    <p id="modalDescription"></p>
  </div>
</div>
```

Наприклад, гістограма віку показує пік відгуків у 39 років, а кругова діаграма рейтингів вказує, що 55.91% товарів мають рейтинг 5. Статичні графіки

зберігаються як PDF і відображаються в модальних вікнах, які відкриваються при натисканні на прев'ю зображення.

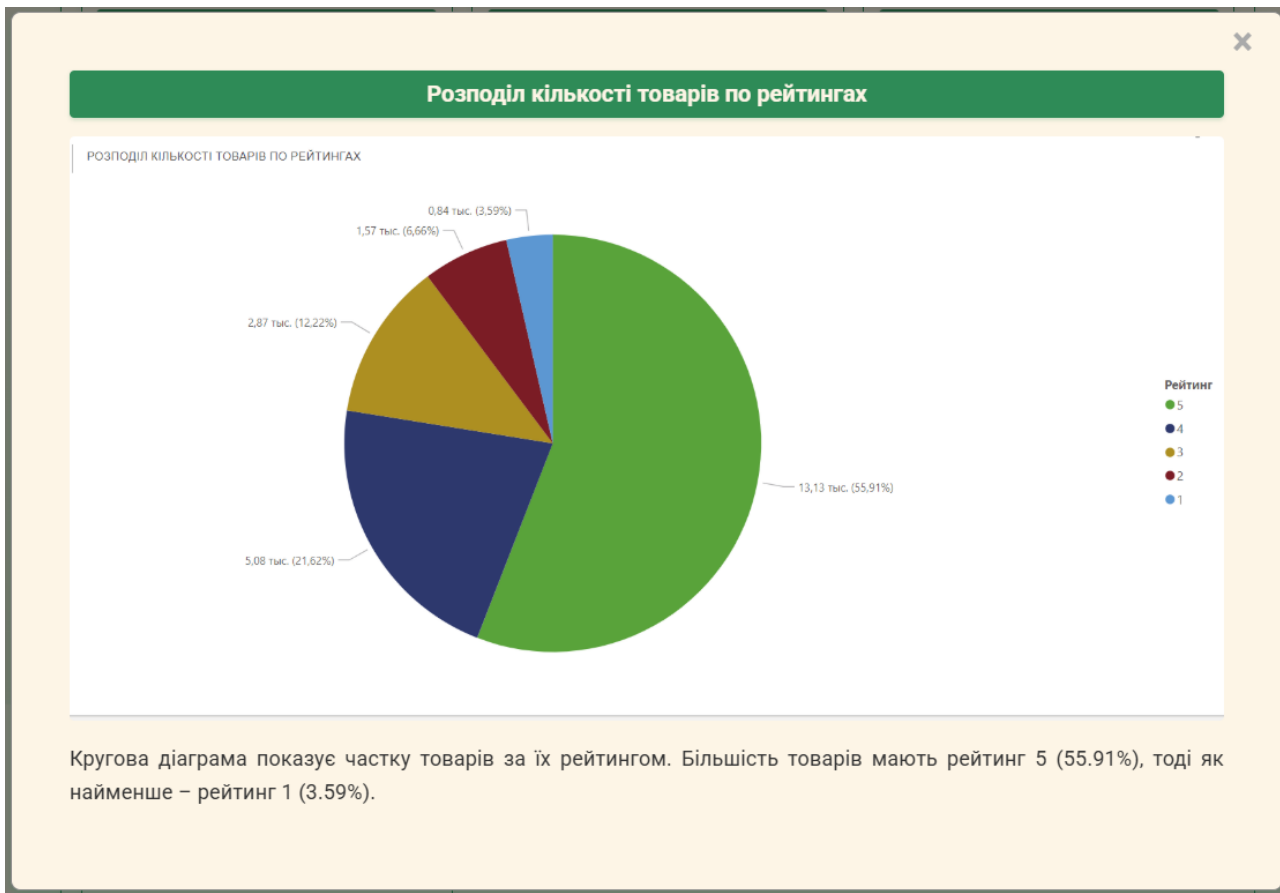


Рисунок 2.10 – Статичні графіки

Модальне вікно реалізовано через JavaScript-функцію `openModal`. Ось приклад цієї функції:

```
// Функція для відображення модального вікна в category_stats.html
function openModal(imageSrc) {
    const modal = document.createElement('div');
    modal.className = 'modal';
    modal.innerHTML = `
        <div class="modal-content">
            <span class="close"
                onclick="this.parentElement.parentElement.remove()">&times;</span>
            
        </div>
    `;
    document.body.appendChild(modal);
}
```

```
}

```

Модальне вікно має бежевий фон (#FDF5E6) зі смарагдовою обводкою (#2E8B57). Ось приклад CSS-стилів із style.css для модального вікна та адаптивності елементів керування:

```
/* Стилі для модального вікна та адаптивності в style.css */
.chart-controls {
  display: flex;
  gap: 20px;
  margin-bottom: 20px;
}

.chart-controls label {
  color: #2E8B57;
  font-weight: bold;
}

.chart-controls select, .chart-controls button {
  padding: 8px;
  background-color: #2E8B57;
  color: #FFFFFF;
  border: none;
  border-radius: 5px;
}

.modal {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.5);
  display: flex;
  justify-content: center;
  align-items: center;
}

.modal-content {
  background: #FDF5E6;
  border: 3px solid #2E8B57;
  padding: 20px;
  position: relative;
  max-width: 90%;
  max-height: 90%;
  overflow: auto;
}

.close {
  position: absolute;
```

```
    top: 10px;
    right: 10px;
    font-size: 24px;
    cursor: pointer;
}

@media (max-width: 768px) {
  .chart-controls {
    flex-direction: column;
    align-items: flex-start;
  }
  #statsChart {
    max-width: 100%;
  }
}
```

Адаптивність інтерактивних елементів забезпечує коректне відображення на різних пристроях: на екранах до 768 пікселів випадаючі списки для вибору типу графіку вирівнюються вертикально, а розмір графіку зменшується. Обмеження включають статичну природу PDF-графіків, які не дозволяють змінювати параметри, і відсутність фільтрації даних у реальному часі (наприклад, за віком чи рейтингом). Інтерактивний графік підтримує масштабування: користувач може наводити курсор на стовпчики, щоб побачити точні значення, наприклад, 150 товарів у категорії «Dresses». Експорт графіку в PNG полегшує збереження результатів аналізу.

Загалом, інтерактивні елементи дозволяють користувачам гнучко досліджувати статистику, аналізувати залежності між категоріями, рейтингами та відгуками, а також порівнювати товари. Інтеграція Chart.js і асинхронних запитів через FastAPI забезпечує швидке оновлення графіків, хоча додавання фільтрів для даних могло б розширити можливості аналізу.

2.4 Висновки до розділу

У спеціальному розділі кваліфікаційної роботи виконано детальний аналіз даних і розроблено рекомендаційну систему для веб-сайту «Elizabeth's Elegance», орієнтованого на продаж жіночого одягу. Аналіз базувався на наборі даних «Women's E-Commerce Clothing Reviews», що містить 23 486 відгуків, і дозволив виявити ключові закономірності у вподобаннях користувачів. Основна увага приділялася категоріям товарів, віковим групам і показникам популярності, що стали основою для створення персоналізованих рекомендацій. Робота охоплює як теоретичні аспекти аналізу даних, так і практичну реалізацію функціональності веб-сайту. Отримані результати мають практичну цінність для підвищення ефективності електронної комерції.

Аналіз даних показав, що категорія «Dresses» є найпопулярнішою, з 6 319 відгуками, а департаменти «Tops» і «Dresses» лідирують за кількістю відгуків. Найвищі середні рейтинги (близько 5) мають категорії «Socks» і «Casual Bottoms», тоді як «Chemises» і «Trend» демонструють нижчі показники. Позитивні оцінки переважають: 55.91% відгуків мають рейтинг 5, а 82.24% супроводжуються рекомендаціями (Recommended IND = 1). Основна аудиторія платформи — користувачі віком 30–50 років, із піком активності у 39 років (1 269 відгуків). Ці дані використані для сегментації користувачів і налаштування рекомендаційної системи.

Ключовими показниками популярності товарів обрано середній рейтинг, кількість позитивних відгуків (Positive Feedback Count) і частку рекомендацій. Ці показники інтегровано в систему через FastAPI-ендпоінт, що дозволяє фільтрувати товари з рейтингом ≥ 4 і позитивними рекомендаціями. Наприклад, категорія «Knits» вирізняється високою кількістю позитивних відгуків, що свідчить про її популярність. Візуалізації, такі як кругові діаграми та гістограми, допомогли виявити найпривабливіші товари й категорії. Обмеженням є

відсутність аналізу текстових відгуків (Review Text), що могло б додати глибину оцінки.

Рекомендаційна функціональність реалізована через два основні ендпоінти: `/age-recommendations` для рекомендацій на основі віку та `/product/{clothing_id}` для рекомендацій схожих товарів. Рекомендації за віком фільтрують відгуки в діапазоні ± 5 років від введеного віку, пропонуючи до 7 товарів із високим рейтингом. Наприклад, для користувачів 30–39 років часто рекомендуються товари категорії «Dresses». Рекомендації за товаром базуються на однаковій категорії, наприклад, пропонуючи інші сукні для товару з категорії «Dresses». Обмеженням є статична природа фільтрації, яка не враховує додаткові параметри, як-от сезонність чи стиль.

Рекомендаційна система реалізована за допомогою FastAPI, шаблонів Jinja2 і SQLAlchemy для ефективного зберігання та обробки даних. Інтерфейс сайту виконано в смарагдово-бежевій палітрі, з адаптивним дизайном для десктопів, планшетів і смартфонів. Сторінки включають слайдери для відображення товарів, фільтри за віком, рейтингом і рекомендаціями, а також інтерактивну статистику на сторінці `/category-stats`. Інтерактивні графіки, створені за допомогою Chart.js, дозволяють користувачам аналізувати дані, обираючи тип графіку та параметри осей. Статичні графіки ілюструють розподіл рейтингів і відгуків, але не підтримують динамічної зміни параметрів.

Система фільтрів і сортування забезпечує гнучкий пошук товарів за підрозділом, департаментом, категорією, рейтингом і віком. Сортування доступне за рейтингом, кількістю відгуків та ID, із пагінацією від 12 до 96 товарів на сторінці. Фільтри на сторінці товару дозволяють аналізувати відгуки за допомогою слайдера віку (noUiSlider) і чекбоксів для рейтингу та рекомендацій. Обмеженням є необхідність перезавантаження сторінки для застосування фільтрів і відсутність пошуку за текстом відгуків. Асинхронне оновлення могло б покращити користувацький досвід.

Практична цінність роботи полягає в персоналізації рекомендацій для цільової аудиторії (жінки 30–50 років), що сприяє підвищенню залученості та продажів. Для вдосконалення системи рекомендується додати аналіз настрою текстових відгуків, врахування контекстних факторів (сезонність, стиль) і асинхронне оновлення фільтрів. Перспективи подальших досліджень включають інтеграцію методів машинного навчання, таких як нейронні мережі, для підвищення точності рекомендацій, а також розширення функціональності сайту, наприклад, додавання автодоповнення в пошуковій системі.

ВИСНОВКИ

У рамках кваліфікаційної роботи проведено аналіз набору даних «Women's E-Commerce Clothing Reviews» (23 486 відгуків) для розробки рекомендаційної системи веб-сайту «Elizabeth's Elegance». За допомогою Power BI встановлено, що категорія «Dresses» є найпопулярнішою (6 319 відгуків), за нею йдуть «Knits» і «Blouses», тоді як «Chemises» має найменшу кількість відгуків (1 одиниця). Департаменти «Tops» і «Dresses» лідирують за кількістю позитивних відгуків, а «Intimate», «Jackets» і «Trend» мають нижчу активність. Категорія «Socks» вирізняється найвищим середнім рейтингом (~5), тоді як «Chemises» і «Trend» мають найнижчі показники. Розподіл рейтингів показав переважання позитивних оцінок: 55.91% відгуків мають оцінку 5, 21.62% — 4, а частка рекомендацій становить 82.24%. Категорії «Casual Bottoms» і «Chemises» досягають майже 100% рекомендацій, тоді як «Trend» має найнижчий відсоток.

Аналіз за віком виявив, що основна аудиторія платформи — користувачі 30–50 років, із піком активності у 39 років (1 269 відгуків). Ключовими показниками популярності товарів обрано середній рейтинг, кількість позитивних відгуків і частку рекомендацій, які інтегровано в рекомендаційну систему через FastAPI-ендпоінти. Реалізовано веб-сайт із адаптивним інтерфейсом, що включає рекомендації за віком (/age-recommendations), схожі товари (/product/{clothing_id}), фільтри, сортування та інтерактивні графіки (Chart.js) для аналізу статистики. Візуалізації (гістограми, кругові діаграми) підтвердили популярність «Dresses» і «Knits» та низьку активність «Chemises» і «Trend».

Для підвищення ефективності рекомендаційної системи пропонується додати аналіз настрою текстових відгуків (Review.Text), що дозволить глибше оцінити вподобання користувачів. Впровадження контекстних факторів, таких як сезонність, стиль чи колір, підвищить релевантність рекомендацій. Рекомендується реалізувати асинхронне оновлення фільтрів і сортування, щоб

уникнути перезавантаження сторінок і покращити користувацький досвід. Додавання автодоповнення в пошукову систему спростить пошук товарів. Для оптимізації асортименту слід збільшити запаси популярних категорій («Dresses», «Knits») і переглянути пропозиції в менш популярних («Chemises», «Trend»). Маркетингові стратегії варто зосередити на цільовій аудиторії 30–50 років, акцентуючи на товарах із високими рейтингами.

Подальші дослідження можуть бути спрямовані на інтеграцію методів машинного навчання, таких як нейронні мережі, для підвищення точності рекомендацій шляхом аналізу складних шаблонів поведінки користувачів. Розширення функціональності веб-сайту передбачає додавання аналізу даних у реальному часі, фільтрації за текстом відгуків і врахування додаткових параметрів (наприклад, геолокація, історія покупок). Перспективним є створення мобільного додатку для «Elizabeth's Elegance», що забезпечить ширший доступ до рекомендаційної системи. Впровадження адаптивних алгоритмів, стійких до маніпуляцій (наприклад, фальшивих відгуків), підвищить надійність системи. Ці напрямки сприятимуть розвитку персоналізованих рішень в електронній комерції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Годовиченко М. А., Горбатенко А. А. Методологічні принципи інформаційних технологій / М. А. Годовиченко, А. А. Горбатенко // Гуманітарний вісник: Наука. Освіта. Технології. – 2023. – Т. 6, № 4. – С. 308–319. – DOI: <https://doi.org/10.15276/hait.06.2023.20>. – URL: <https://hait.od.ua/index.php/journal/article/view/190/224> (дата звернення: 24.04.2025).
2. Рекомендаційні системи. Skalar. URL: <https://skalar.ua/ua/expertise/recommender-systems> (дата звернення: 24.04.2025).
3. Мелешко Є. В. Методи оцінки якості роботи рекомендаційних систем / Є. В. Мелешко // Системи управління, навігації та зв'язку. – 2018. – Вип. 5(51). – С. 92–98. – DOI: 10.26906/SUNZ.2018.5.092. – URL: https://www.researchgate.net/publication/329792214_METODI_OCINKI_AKOSTI_ROBOTI_REKOMENDACIJNIH_SISTEM (дата звернення: 24.04.2025).
4. Product Reviews Update: нові критерії Google, за якими ранжуватимуться відгуки. Web-Promo. 2023. URL: <https://web-promo.ua/ua/blog/product-reviews-update-novyie-kriterii-google-po-kotorym-budut-ranzhirovatsya-otzyvy/> (дата звернення: 24.04.2025).
5. Основні метрики і КРІ в інтернет-маркетингу. Idea Digital Agency. URL: <https://ideadigital.agency/blog/osnovni-metriki-i-kpi-v-internet-marketingu/> (дата звернення: 24.04.2025).
6. Пахольчук Я. І. Інструменти веб-аналітики для аналізу відвідувачів сайтів / Я. І. Пахольчук // Міжнародні відносини, суспільні комунікації та регіональні студії. – 2018. – № 1. – С. 71–77. – DOI: 10.29038/2524-2679-2018-01-71-77. – URL: https://www.researchgate.net/publication/343074081_Instrumenti_veb-analitiki_dla_analizu_vidviduvaciv_sajtiv (дата звернення: 25.04.2025).

7. Горбатенко А. А., Годовиченко М. А. Досягнення консенсусу в групових рекомендаційних системах / А. А. Горбатенко, М. А. Годовиченко // Гуманітарний вісник: Наука. Освіта. Технології. – 2024. – Т. 7. – С. 33–42. – DOI: 10.15276/hait.07.2024.3.URL:

<https://hait.od.ua/index.php/journal/article/view/203/227> (дата звернення: 25.04.2025).

8. Кіберінциденти в Україні: огляд за 2024 рік. Національний координаційний центр кібербезпеки при РНБО України. Київ: НКЦК, 2025. 48 с. URL:

https://www.rnbo.gov.ua/files/2024/NATIONAL_CYBER_SCC/20250109/Year%20in%20review_UKR_upd.pdf (дата звернення: 25.04.2025).

9. Кудріна О. Ю. Вплив упровадження штучного інтелекту та мереж 5G на управління фінансовим забезпеченням інноваційного розвитку телекомунікаційних підприємств / О. Ю. Кудріна // Державне управління: удосконалення та розвиток. – 2023. – № 16. – С. 1–6. – DOI: 10.32782/dees.16-1. – URL:<http://dees.iei.od.ua/index.php/journal/article/view/546/526> (дата звернення: 25.04.2025).

10. Кваліфікаційна робота бакалавра [Електронний ресурс] : метод. рекомендації для здобувачів ступеня бакалавра освіт.-проф. програми «Системний аналіз» зі спеціальності 124 Системний аналіз / Т. А. Желдак, Т. В. Хом'як, А. В. Малієнко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». — Дніпро : НТУ «ДП», 2025. — 32 с.

11. Моделювання та реінжиніринг бізнес-процесів : навч. посіб. / С. В. Козир, В. В. Слесарєв, С. А. Ус, Т. В. Хом'як ; М-во освіти і науки України. — Дніпро : НТУ «ДП», 2022. — 163 с.

12. Хом'як Т. В. Бази даних у професійних задачах аналітики [Електронний ресурс] : навч. наоч. посіб. / Т. В. Хом'як, К. С. Хабарлак, Д. М. Гаранжа ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». — Дніпро : НТУ «ДП», 2024. — 192 с.

13. Хом'як Т. В. Системний аналіз виявлення проблем системи освіти та шляхи їх вирішення / Т. В. Хом'як, О. Прус // Information Technology: Computer Science, Software Engineering and Cyber Security. — 2024. — Вип. 3. — С. 180–188. — DOI: 10.32782/IT/2024-3-1.

14. Хом'як Т. В. Прогнозування причин виявлення цукрового діабету методами машинного навчання / Т. В. Хом'як, К. Сидоренко, А. В. Малієнко, О. С. Мінеєв // Системні технології. — 2025. — Т. 1, № 156. — С. 39–49. — DOI: 10.34185/1562-9945-1-156-2025-05.

15. Хом'як Т. В. Вирішення задачі маршрутизації транспорту на підприємстві / Т. В. Хом'як, Н. С. Коханчик, А. В. Малієнко // Зб. наук. пр. НГУ. — 2020. — № 63. — С. 145–155.

16. Хом'як Т. В. Застосування методів згладжування для прогнозування обсягу виробництва / Т. В. Хом'як, А. В. Малієнко, Г. В. Симонець // Вісник Нац. техн. ун-ту «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. — 2019. — № 1. — С. 8–12.

17. Хом'як Т. В. Розробка системи підтримки прийняття рішень для вибору виду діяльності / Т. В. Хом'як, А. В. Малієнко, К. К. Фішбах // Зб. наук. пр. НГУ. — 2019. — № 59. — С. 132–142. — DOI: 10.33271/crpnmu/59.132.

18. Сидоренко Є. В. Аналіз причин та прогнозування виявлення цукрового діабету методом машинного навчання Decision Tree / Є. В. Сидоренко, Т. В. Хом'як // The 6th International scientific and practical conference “Methodical and practical methods of creating inventions” (October 24–27, 2023), Sofia, Bulgaria. — International Science Group, 2023. — С. 265–271. — DOI: 10.46299/ISG.2023.2.6.

19. Хом'як Т.В., Рябоконт Н. Системний аналіз та вибір стратегій розподілу гуманітарної допомоги // Information Technology: Computer Science, Software Engineering and Cyber Security, Вип. 1, 2025. -С. 226-237. (<https://doi.org/10.32782/IT/2025-1-30>)

ДОДАТКИ

Додаток А. Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення				Найменування	Кількість аркушів	Примітки			
1										
2					Документація					
3										
4	САУ.КР.25.18.ПЗ				Пояснювальна записка	54	Формат А4			
5										
6					Демонстраційний матеріал	17	Презентація на CD-R			
7										
8					Копія роботи	1	Диск CD-R			
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
					САУ.КР.25.18.ДА.ПЗ.					
Змін.	Аркуш	№ докум.	Підпис	Дата						
Розроб.		Жижикіна Є. Д.			Матеріали кваліфікаційної роботи	Літ.	Аркуш	Аркушів		
К. розд.		Хом'як Т. В.								
Керівн.		Хом'як Т. В.				НТУ «ДП», 12; 124-21-1				
Н.контр.		Хом'як Т. В.								
Зав. каф.		Желдак Т. А.								

Додаток Б. Відгук

Відгук на кваліфікаційну роботу бакалавра Здобувача вищої освіти групи 124 – 21 – 1 спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи: Аналіз ключових показників для розробки рекомендаційної системи

Обсяг кваліфікаційної роботи 72 стор.

Мета кваліфікаційної роботи: Аналіз ключових показників для розробки рекомендаційної системи веб-сайту, що базується на даних відгуків користувачів про жіночий одяг, для підвищення персоналізації та ефективності електронної комерції.

Актуальність теми Зростання попиту на персоналізовані рішення в електронній комерції та необхідність аналізу великих обсягів даних роблять тему актуальною для сучасних досліджень у сфері системного аналізу.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 124 Системний аналіз, оскільки передбачає застосування методів системного аналізу, обробки даних (Power BI, SQL) та програмування (FastAPI, SQLAlchemy, JavaScript) для розробки рекомендаційної системи.

Виконані в кваліфікаційній роботі завдання відповідають вимогам ступеня бакалавра. Оригінальність наукових рішень полягає в розробці адаптивного веб-сайту з інтегрованою рекомендаційною системою, що використовує аналіз вікових груп і схожих товарів.

Практичне значення результатів кваліфікаційної роботи полягає в створенні функціональної рекомендаційної системи для веб-сайту, яка підвищує залученість користувачів і сприяє оптимізації асортименту.

Висновки підтверджують можливість використання результатів роботи в створенні функціональної рекомендаційної системи для веб-сайту, яка підвищує залученість користувачів і сприяє оптимізації асортименту.

Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі (*в разі невідповідності – вказати*)

У роботі відзначено такі недоліки: відсутність аналізу текстових відгуків (Review.Text), що могла б додати глибину оцінки.

Кваліфікаційна робота в цілому заслуговує оцінки: “відмінно”.

З урахуванням висловлених зауважень автор заслуговує присвоєння кваліфікації «бакалавр з системного аналізу».

Керівник кваліфікаційної роботи бакалавра,
к.ф.-м.наук,доцент

Хом'як Т.В.

Додаток В. Участь у конференції

Міністерство освіти і науки України
 Національний університет
 «Запорізька політехніка»
 Національний технічний університет
 «Дніпровська політехніка»
 Харківський національний університет
 міського господарства імені О.М. Бекетова
 ГО «Системні дослідження»
 ГО МДЦВЕ
 Esslingen University of Applied Sciences
 University of Koblenz
 Cardiff University
 Kırklareli University
 Universidad Politécnica de Madrid



ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ТЕОРІЯ І ПРАКТИКА

II (VIII) Міжнародна Інтернет-конференція
 здобувачів вищої освіти і молодих учених

2-4 квітня 2025 р

Тези доповідей

Запоріжжя-Харків-Дніпро

Україна

Белова К.О., Новожилова М.В. Методи прийняття рішень у військовій сфері: аналіз ризиків та невизначеностей (ХНУМГ ім. О.М. Бекетова)	111
Василишин Р.А. Порівняння ефективності формату кодування інформації між системами chat gpt та deepseek (НУВІП)	113
Жижикіна Є.Д., Хом'як Т.В. Аналіз текстових відгуків у рекомендаційних системах жіночого одягу (НТУ «Дніпровська політехніка»)	116
Жикул Д.С., Мороз В.В. Алгоритми візуального виявлення місць (ОНУ імені І.І. Мечникова)	119
Казьмірчук О.Г., Садовенко В.С. Методичні засади застосування інструментальних засобів мови python для інтелектуального аналізу та класифікації текстових повідомлень	123
Кіншаков Е.В., Парфененко Ю.В. Проектування діагностичної спр в дерматології (СумДУ)	127
Литвин В.В., Пасемко І.С. Застосування алгоритмів машинного навчання для класифікації важливості повідомлень (НУ «Львівська політехніка»)	131
Луценко О.В., Щербак С.С. Вибір оптимальних параметрів для графових нейронних мереж для завдання оцінки ризику страхування індивідуумів (НУ «Львівська політехніка»)	133
Максюта О.І., Левіков Ю.В. Інноваційні технології створення відеоконтенту: дослідження функціоналу lumen.5 (ХНУМГ імені О.М. Бекетова)	135
Малий А.Ю., Бочаров Б.П. Використання штучного інтелекту в задачах автоматизації та підтримки інформаційних систем (ХНУМГ імені О.М. Бекетова)	138
Марків О.І., Ришковець Ю.В. Інтеграція потокових big data технологій у системи керування зарядними станціями електромобілів (НУ «Львівська Політехніка»)	140
Муковоз Д.А., Воєводіна М.Ю. Інноваційні технології створення вебсайтів: дослідження функціоналу durable ai (ХНУМГ імені О.М. Бекетова)	144
Наум О.М., Угрин Д.І., Литвин В.В. Система підтримки прийняття рішень для туризму на основі ройового інтелекту (НУ «Львівська політехніка», ЧНУ імені Юрія Федьковича)	147
Неласий О.В., Неласа Г.В. Порівняльний аналіз технологій згладжування та масштабування зображення (НУ «Запорізька політехніка»)	150

АНАЛІЗ ТЕКСТОВИХ ВІДГУКІВ У РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ ЖІНОЧОГО ОДЯГУ

Сучасна електронна комерція активно застосовує штучний інтелект (ШІ) для персоналізації рекомендацій, що підвищує залученість клієнтів і продажі [4]. Текстові відгуки користувачів містять цінну інформацію про вподобання, але часто залишаються невикористаними через складність їх обробки [5]. Наприклад, відгуки можуть розкривати причини популярності певних товарів, таких як жіночий одяг, але традиційні системи цього не враховують. Метою дослідження є оцінка потенціалу ШІ для аналізу текстових відгуків у рекомендаційних системах для жіночого одягу. Проблема полягає в тому, що неструктуровані дані, такі як відгуки, потребують спеціальних інструментів і підходів для ефективного використання [5].

Для аналізу застосовуються набори даних, що включають текстові відгуки, рейтинги та категорії жіночого одягу, наприклад, сукні чи трикотаж. ШІ використовує аналіз тональності (sentiment analysis), який класифікує відгуки як позитивні, негативні чи нейтральні, щоб оцінити ставлення клієнтів до товарів [1]. Обробка природної мови (NLP) дозволяє витягувати ключові слова, такі як "зручний" чи "стильний", для створення точніших рекомендацій [2]. Сучасні моделі, зокрема BERT, ефективно аналізують текстові дані, відкриваючи нові можливості для eCommerce [3]. Ці методи можуть бути інтегровані в системи рекомендацій для підвищення функціональності.

Аналіз відгуків показує, що популярність певних категорій, наприклад суконь, може бути зумовлена їх характеристиками, але без тексту ці причини важко визначити. Позитивний відгук, як-от "Ця сукня ідеально сидить", вказує на комфорт, що корисно для підбору схожих товарів [1].

На рис. 1 представлено типовий розподіл позитивних відгуків за категоріями жіночого одягу, де "Dresses" і "Knits" лідирують із кількістю відгуків понад 5 тис., а "Chemises" має найменше - менше 1 тис.

¹ студентка групи 124-21-1, НТУ «Дніпровська політехніка»

² доцент, каф. САУ, НТУ «Дніпровська політехніка», к. ф.-м. н.

Результати свідчать, що аналіз відгуків покращує персоналізацію, як це демонструють кейси Microsoft, де ШІ визначає настрої клієнтів [1]. Точні рекомендації сприяють економічній вигоді, підвищуючи продажі в онлайн-магазинах [4]. Проте обробка великих обсягів тексту потребує значних обчислювальних ресурсів, що може бути викликом для малих платформ. Якість даних також відіграє ключову роль: короткі чи неоднозначні відгуки ускладнюють аналіз. У 2025 році ШІ стане невід'ємною частиною eCommerce, особливо в ніші жіночого одягу, де персоналізація має велике значення [4]. Поєднання текстових даних із демографічною інформацією може надати глибші інсайти про вподобання різних груп клієнтів.

Висновки ШІ, аналізуючи текстові відгуки, значно підвищує ефективність рекомендаційних систем. Його інтеграція в системи eCommerce є перспективною для жіночого одягу, де важлива індивідуальність вибору. Подальші дослідження мають зосередитися на оптимізації обчислювальних ресурсів і підвищенні якості даних для точнішого аналізу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Microsoft Data Science. Improving Product Recommendation Systems Using Sentiment Analysis [Електронний ресурс] / Medium. – 2024. – Режим доступу: <https://medium.com/data-science-at-microsoft/improving-product-recommendation-systems-using-sentiment-analysis-52ead43211dd>
2. Insight7. Text Analytics for E-commerce: Enhancing Product Recommendations [Електронний ресурс]. – Режим доступу: <https://insight7.io/text-analytics-for-e-commerce-enhancing-product-recommendations/>
3. Srivastava A. The Application Impact of Artificial Intelligence (AI) on E-Commerce [Електронний ресурс] / ResearchGate. – 2021. – Режим доступу: https://www.researchgate.net/profile/Ambar-Srivastava/publication/356635263_The_Application_Impact_of_Artificial_Intelligence_AI_on_E-Commerce/links/61a6046a6864311d938a92b1/The-Application-Impact-of-Artificial-Intelligence-AI-on-E-Commerce.pdf
4. The Inweb Media. Тренди штучного інтелекту на 2025 рік [Електронний ресурс]. – 2024. – Режим доступу: <https://theinweb.media/trendy-shtuchnogo-intelektu-na-2025-rik/>
5. Signity Solutions. AI in Retail and eCommerce: Applications, Benefits and Challenges [Електронний ресурс]. – Режим доступу: <https://www.signitysolutions.com/blog/ai-in-retail-and-ecommerce-applications-benefits-and-challenges>

Рецензія
на кваліфікаційну роботу бакалавра
здобувача вищої освіти групи 124 – 21 – 1
спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи:

Аналіз ключових показників для розробки рекомендаційної системи

Обсяг кваліфікаційної роботи: 72 сторінки

Висновок про відповідність кваліфікаційної роботи завданню та освітньо-професійній програмі спеціальності Робота повністю відповідає поставленому завданню, що передбачало аналіз даних і розробку рекомендаційної системи, а також узгоджується з вимогами освітньо-професійної програми «Системний аналіз», де акцент зроблено на обробці даних і програмуванні.

Загальна характеристика кваліфікаційної роботи, ступінь використання нормативно-методичної літератури та передового досвіду Робота демонструє ґрунтовний підхід до аналізу даних і практичну реалізацію, з активним використанням нормативно-методичної літератури (10–18 джерел) та сучасних інструментів (Power BI, FastAPI, Chart.js), що відображає передовий досвід у сфері рекомендаційних систем.

Позитивні сторони кваліфікаційної роботи:

Відзначається глибоким аналізом набору даних, вдалою розробкою адаптивного веб-сайту з інтерактивними елементами та високою практичною цінністю для електронної комерції.

Основні недоліки кваліфікаційної роботи: відсутність функції автодоповнення в пошуковій системі та ігнорування контекстних факторів (наприклад, сезонність) у рекомендаціях зменшують зручність і релевантність системи.

Кваліфікаційна робота в цілому заслуговує оцінки: “відмінно”

З урахуванням висловлених зауважень автор заслуговує присвоєння кваліфікації «бакалавр з системного аналізу».

Рецензент,
д.т.н, професор,
зав. каф. ПЗКС

Алексєєв М.О.