

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:
**«СИСТЕМА ІНТЕГРАЦІЇ ТОВАРНИХ ДАНИХ У CRM ТА ERP-
СИСТЕМАХ ІЗ ЗАСТОСУВАННЯМ МЕТОДІВ МАШИННОГО
НАВЧАННЯ»**

на здобуття освітнього ступеня магістр
за спеціальності 124 Системний аналіз

(код, найменування спеціальності)

освітньо-професійної програми Інтелектуальні системи управління

(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Микола МАНДРА

(ім'я, ПРІЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група САДМ-61

Микола МАНДРА

(ім'я, ПРІЗВИЩЕ)

Керівник

*д.т.н.
професор*

Каміла СТОРЧАК

(ім'я, ПРІЗВИЩЕ)

Рецензент:

(ім'я, ПРІЗВИЩЕ)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти магістр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма Інтелектуальні системи управління

ЗАТВЕРДЖУЮ

Завідувач кафедру ІСТ

Каміла СТОРЧАК

“ ___ ” _____ 2025 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Мандрі Миколі Васильовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Система інтеграції товарних даних у CRM та ERP-системах із застосуванням методів машинного навчання

керівник кваліфікаційної роботи: Каміла СТОРЧАК д.т.н., професор

(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “30” жовтня 2025 р. № 467

2. Строк подання кваліфікаційної роботи «26» грудня 2025 р.

3. Вихідні дані кваліфікаційної роботи: Вихідними даними кваліфікаційної роботи є товарні дані торговельного підприємства, що надходять від постачальників у різних форматах (XML, API, CSV, Excel), інформація з внутрішніх інформаційних систем підприємства (CRM-система KeyCRM, ERP-система BAS, база даних вебсайту інтернет-магазину та облікова система роздрібних магазинів Ultra). Додатковими вихідними даними є результати обробки неструктурованих текстових описів товарів, отримані з використанням API доступу до попередньо навчених моделей машинного навчання, зокрема великих мовних моделей, які застосовуються для автоматизованої класифікації, нормалізації та інтеграції товарних даних.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно

розробити):

1. Аналіз існуючого процесу отримання, обробки та інтеграції товарних даних у CRM та ERP-системах торговельного підприємства.
2. Обґрунтування вибору архітектури системи інтеграції товарних даних та методів машинного навчання для їх автоматизованої обробки.
3. Розробка та опис алгоритму автоматизованого формування товарних атрибутів і характеристик із використанням API доступу до великих мовних моделей.
4. Реалізація інтеграції результатів обробки товарних даних у CRM-систему та пов'язані інформаційні ресурси підприємства.
5. Оцінка ефективності впровадженої системи та аналіз отриманих результатів автоматизації.

5. Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «30» жовтня 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури		Виконано
2.	Дослідження тенденцій розвитку та застосування CRM та ERP-систем у торговельних підприємствах		Виконано
3.	Аналіз методів машинного навчання та NLP для обробки і нормалізації товарних даних		Виконано
4.	Дослідження архітектур систем інтеграції та інтеграційних платформ для багатосистемного середовища		Виконано
5.	Проектування архітектури системи інтеграції товарних даних між CRM, ERP та вебресурсами		Виконано
6.	Розробка та реалізація системи інтеграції товарних даних із використанням OpenAI API		Виконано
7.	Тестування розробленої системи та аналіз результатів машинного навчання		Виконано

8.	Підготовка практичних рекомендацій щодо впровадження системи в діяльність підприємства		Виконано
9.	Розробка демонстраційних матеріалів та підготовка презентації для захисту		Виконано
7.	Оформлення магістерської роботи		Виконано

Здобувач вищої освіти _____ Микола МАНДРА
 (підпис) (ім'я, ПРІЗВИЩЕ)
 Керівник кваліфікаційної роботи _____ Каміла СТОРЧАК
 (підпис) (ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 87 стор., 6 рис., 4 табл., 16 джерел.

Мета роботи – дослідження та розробка системи інтеграції товарних даних між CRM та ERP-системами з використанням методів машинного навчання для підвищення якості, повноти та узгодженості даних у торговому підприємстві.

Об'єкт дослідження – процес інтеграції товарних даних у інформаційних системах торгового підприємства.

Предмет дослідження – методи, моделі та програмні засоби інтеграції, очищення, нормалізації та зіставлення товарних даних у CRM, ERP та суміжних інформаційних системах із застосуванням методів машинного навчання.

Короткий зміст роботи.

У цій дипломній роботі розроблено та впроваджено систему інтеграції товарних даних у CRM та ERP-системах торговельного підприємства із застосуванням методів машинного навчання та API-доступу до великих мовних моделей. У результаті роботи сформовано практичні рекомендації щодо впровадження системи інтеграції товарних даних у торгових підприємствах для підвищення якості інформації, зменшення ручної обробки та оптимізації бізнес-процесів.

КЛЮЧОВІ СЛОВА: CRM, ERP, інтеграція даних, товарні дані, KeyCRM, BAS, API, XML, CSV, Excel, машинне навчання, класифікація, зіставлення даних, дедуплікація, нормалізація даних, автоматизація бізнес-процесів, інформаційні системи, e-commerce.

ABSTRACT

The text part of the qualifying work for obtaining a bachelor's degree: 87 pp., 6 fig., 4 tables., 16 sources.

The purpose of the research is to develop and study a product data integration system between CRM and ERP systems using machine learning methods in order to improve data quality, completeness, and consistency in a trading enterprise.

The object of the research is the process of product data integration in the information systems of a trading enterprise.

The subject of the research is methods, models, and software tools for the integration, cleansing, normalization, and matching of product data in CRM, ERP, and related information systems using machine learning methods.

Brief description of the research.

In this thesis, a product data integration system in CRM and ERP systems of a trading enterprise was developed and implemented using machine learning methods and API access to large language models. As a result of the work, practical recommendations were formed for the implementation of a product data integration system in trading enterprises to improve information quality, reduce manual processing, and optimize business processes.

KEYWORDS: CRM, ERP, data integration, product data, KeyCRM, BAS, API, XML, CSV, Excel, machine learning, classification, data matching, deduplication, data normalization, business process automation, information systems, e-commerce.

ЗМІСТ

ВСТУП.....	12
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ІНТЕГРАЦІЇ ТОВАРНИХ ДАНИХ У CRM ТА ERP-СИСТЕМАХ ІЗ ЗАСТОСУВАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ	15
1.1. CRM та ERP-системи у торговельних підприємствах: призначення та функціональні можливості.....	15
1.1.1. Роль CRM у керуванні товарними даними	15
1.1.2. Призначення ERP-систем	15
1.1.3. Взаємодія CRM та ERP у багатоканальному продажі.....	16
1.2. Проблеми інтеграції товарних даних у багатосистемному середовищі....	17
1.2.1. Різноманітність форматів товарних даних	17
1.2.2. Дублювання та неузгодженість товарної інформації.....	18
1.2.3. Вплив якості даних на бізнес-процеси.....	18
1.3. Методи машинного навчання для обробки та нормалізації товарних даних.....	19
1.3.1. Класифікація та вилучення атрибутів товарів.....	19
1.3.2. Методи обробки природної мови (NLP).....	20
1.3.3. Попередньо навчені моделі та підхід transfer learning.....	20
1.4. Використання великих мовних моделей у системах інтеграції даних.....	21
1.4.1. Архітектура LLM та принципи роботи	21
1.4.2. API-доступ до ML-моделей.....	22
1.4.3. Переваги та обмеження використання LLM у корпоративних системах...22	
ВИСНОВОК ДО РОЗДІЛУ 1	23
РОЗДІЛ 2 АНАЛІЗ СИСТЕМИ УПРАВЛІННЯ ТОВАРНИМИ ДАНИМИ НА ТОРГОВЕЛЬНОМУ ПІДПРИЄМСТВІ	25
2.1 Характеристика торговельного підприємства та його інформаційної інфраструктури.....	25
2.1.1. Загальна характеристика підприємства.....	25

2.1.2. Опис використовуваних систем: KeyCRM, BAS, вебсайт, Ultra.....	25
2.1.3. Структура товарних даних.....	26
2.2. Аналіз процесу отримання та обробки товарних даних від постачальників.....	27
2.2.1. Джерела надходження товарної інформації.....	27
2.2.2. Формати даних та особливості їх обробки.....	27
2.2.3. Поточний рівень автоматизації.....	28
2.3. Проблеми формування товарних характеристик і атрибутів у CRM та на сайті.....	29
2.3.1. Ручне заповнення характеристик.....	29
2.3.2. Помилки та неузгодженість даних.....	30
2.3.3. Вплив на користувацький досвід та продажі.....	30
2.4. Обґрунтування доцільності застосування методів машинного навчання....	30
2.4.1. Порівняння rule-based та ML-підходів.....	30
2.4.2. Очікувані переваги автоматизації.....	31
2.4.3. Вибір технологічного підходу.....	31
ВИСНОВОК ДО РОЗДІЛУ 2	32
РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ ІНТЕГРАЦІЇ ТОВАРНИХ ДАНИХ ІЗ ЗАСТОСУВАННЯМ OPENAI API.....	34
3.1 Архітектура системи інтеграції товарних даних.....	34
3.1.1. Загальна схема взаємодії компонентів.....	34
3.1.2. Модуль збору та попередньої обробки даних.....	36
3.1.2. Інтеграційний шар CRM та ERP.....	38
3.2. Реалізація модуля автоматизованої обробки товарних даних.....	40
3.2.1 Структура вхідних даних.....	40
3.2.2. Нормалізація та підготовка даних.....	42
3.2.3. Формування уніфікованої моделі товару.....	53
3.3 Автоматизоване формування товарних характеристик із використанням OpenAI API.....	58
3.3.1. Постановка задачі класифікації товарних атрибутів.....	59
3.3.2. Логіка взаємодії з мовною моделлю.....	60

3.3.3. Формування структурованого JSON-виводу.....	62
3.3.4. Інтеграція результатів у CRM.....	62
3.4. Валідація та контроль якості результатів машинного навчання.....	71
3.5. Оцінка ефективності впровадженої системи.....	73
3.5.1. Порівняння процесів до та після автоматизації.....	73
3.5.2. Зменшення витрат часу та кількості помилок.....	73
3.5.3. Практичні результати впровадження.....	74
ВИСНОВОК ДО РОЗДІЛУ 3	75
ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)	82

ВСТУП

Актуальність теми пов'язана з реальною потребою торговельних підприємств у наявності єдиної та узгодженої бази товарних даних, яка дозволяє забезпечити коректну роботу CRM та ERP-систем, веб ресурсів та облікових систем. Застосування методів машинного навчання дозволяє зменшити обсяг ручної роботи, підвищити точність обробки даних і забезпечити масштабованість рішень у разі зростання кількості товарів та постачальників.

Метою кваліфікаційної роботи є створення та дослідження системи інтеграції товарних даних між CRM та ERP-системами за допомогою методів машинного навчання для підвищення якості, повноти та узгодженості даних у торговельному підприємстві.

Для досягнення мети необхідно вирішити такі *завдання*:

- проаналізувати сучасні підходи до управління та інтеграції товарних даних у CRM та ERP-системах;
- дослідити проблеми обробки товарних даних, що надходять із різних джерел і форматів;
- проаналізувати існуючі методи ML, придатні для класифікації, нормалізації та зіставлення товарних даних;
- спроектувати архітектуру системи інтеграції товарних даних у багатосистемному середовищі;
- розробити та реалізувати прототип системи інтеграції товарних даних із використанням методів ML;
- оцінити ефективність запропонованого рішення та сформулювати практичні рекомендації щодо його впровадження.

Об'єктом дослідження є процес інтеграції товарних даних в інформаційних системах торговельного підприємства.

Предметом дослідження є методи, моделі та програмні засоби інтеграції, очищення, нормалізації та зіставлення товарних даних у CRM, ERP та суміжних інформаційних системах із застосуванням методів машинного навчання.

Під час виконання роботи було застосовано *методи дослідження*: аналіз і

узагальнення наукових та технічних джерел, системний підхід, методи обробки природної мови й машинного навчання, а також підходи до проектування інформаційних систем і експериментальну оцінку ефективності запропонованих рішень.

Наукова новизна роботи полягає у розробці підходу до інтеграції товарних даних у CRM та ERP-системах із використанням попередньо навчених моделей машинного навчання для автоматизованого формування та нормалізації товарних атрибутів у багато системному середовищі.

Практичне значення роботи полягає в тому, що отримані результати можуть бути використані під час впровадження системи інтеграції товарних даних у діяльності торговельних підприємств для підвищення якості інформації, скорочення часу її обробки та оптимізації бізнес-процесів.

Опис структури роботи

Дана дипломна робота складається з трьох основних розділів:

У першому розділі магістерської роботи проведено аналіз сучасних підходів до управління товарними даними в CRM та ERP-системах. Розглянуто особливості використання систем KeyCRM, BAS, внутрішніх баз даних інтернет-магазину та торгових точок на базі Ultra. Проаналізовано проблеми фрагментації, дублювання та неузгодженості товарних даних, що виникають при роботі з декількома джерелами інформації.

У другому розділі досліджено методи інтеграції даних із зовнішніх джерел, зокрема файлів форматів XML, CSV, Excel та API постачальників. Розглянуто архітектури інтеграційних систем, принципи побудови ETL/ELT-процесів та можливості використання API для синхронізації даних між CRM, ERP та веб-системами. Окрему увагу приділено аналізу методів машинного навчання для автоматичного зіставлення товарів, класифікації, виявлення дублікатів і нормалізації атрибутів.

У третьому розділі розроблено та реалізовано прототип системи інтеграції товарних даних з використанням методів машинного навчання та сервісів на основі API. Описано процес автоматизованої обробки товарних даних, їх очищення, стандартизації та передачі між системами KeyCRM, BAS та іншими компонентами інформаційної інфраструктури підприємства. Проведено тестування системи, виконано аналіз отриманих результатів та оцінено ефективність застосування методів машинного навчання в задачах інтеграції товарних даних.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІНТЕГРАЦІЇ ТОВАРНИХ ДАНИХ У CRM ТА ERP-СИСТЕМАХ ІЗ ЗАСТОСУВАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ

1.1. CRM та ERP-системи у торговельних підприємствах: призначення та функціональні можливості

1.1.1. Роль CRM у керуванні товарними даними

CRM-системи (Customer Relationship Management) у торговельних підприємствах призначені для централізованого управління взаємодією компанії з клієнтами, обробки замовлень та підтримки процесів продажу. В умовах розвитку електронної комерції та багатоканальних продажів CRM-система все частіше виконує роль оперативного центру роботи з товарною інформацією, що використовується у взаємодії з кінцевим споживачем.

У контексті керування товарними даними CRM-система забезпечує зберігання та обробку таких атрибутів, як найменування товару, артикул, бренд, категорія, ціна, наявність, характеристики, зображення та фільтри для пошуку. Дані, що зберігаються в CRM, безпосередньо впливають на формування каталогу товарів на веб сайті інтернет-магазину, коректності оформлення замовлень та якості клієнтського сервісу.

Використання CRM-систем, дозволяє автоматизувати процеси оновлення товарних даних, синхронізації цін і залишків з ERP та сайтом, а також інтеграції з маркетплейсами та зовнішніми каналами продажу. Проте через орієнтацію CRM-систем на операційні процеси продажу вони зазвичай не містять розширених механізмів аналітичної обробки та обліку товарних даних, що зумовлює необхідність їх передачі в ERP-системи.

1.1.2. Призначення ERP-систем

ERP (Enterprise Resource Planning) призначені для комплексного управління ресурсами підприємства, включаючи фінанси, складські запаси, закупівлі, бухгалтерський та управлінський облік. На відміну від CRM, ERP-

системи орієнтовані на внутрішні бізнес-процеси підприємства та забезпечують цілісність і достовірність облікових даних.

Ключова роль ERP-системи у формуванні та підтримці єдиної номенклатури товарів, обліку залишків, контролі руху товарів між складами та магазинами, а також розрахунку собівартості та фінансових показників. ERP-система є основним джерелом достовірної інформації щодо фактичної наявності товарів і фінансових результатів діяльності підприємства.

Разом із тим ERP-системи, як правило, не пристосовані до оперативної роботи з неструктурованими або слабо структурованими даними, такими як описи товарів, маркетингові характеристики чи фільтри для клієнтського пошуку. Це ускладнює використання ERP як єдиного інструменту для управління товарною інформацією в умовах e-commerce.

1.1.3. Взаємодія CRM та ERP у багатоканальному продажі

У багатоканальному середовищі продажів, що включає інтернет-магазин, роздрібні торгові точки, маркетплейси та B2B-канали, ефективна взаємодія між CRM та ERP-системами є критично важливою умовою стабільної роботи підприємства. CRM та ERP виконують взаємодоповнювальні функції, формуючи єдину інформаційну систему.

CRM-система забезпечує оперативну роботу з клієнтами та товарними пропозиціями, тоді як ERP відповідає за точність обліку та контроль ресурсів. Інтеграція між цими системами дозволяє автоматично передавати дані про номенклатуру товарів, ціни, залишки, замовлення та статуси їх виконання. Водночас відсутність єдиних стандартів представлення товарних даних та різні вимоги до їх структури призводять до проблем синхронізації та неузгодженості інформації.

У таких умовах особливої актуальності набуває застосування інтелектуальних методів обробки даних, які дозволяють автоматизувати процеси нормалізації, класифікації та зіставлення товарних даних між CRM та ERP-системами. Це створює передумови для побудови масштабованих інтеграційних

рішень, здатних ефективно працювати з великими обсягами різномірної товарної інформації у багатоканальному середовищі продажів.

CRM та ERP-системи відіграють ключову роль у діяльності торговельних підприємств, забезпечуючи управління як клієнтськими, так і обліковими процесами. Водночас відмінності у їх функціональному призначенні зумовлюють необхідність тісної інтеграції для формування єдиного простору товарних даних. Недоліки традиційних підходів до інтеграції створюють передумови для використання методів машинного навчання як інструменту підвищення якості та узгодженості товарної інформації.

1.2. Проблеми інтеграції товарних даних у багатосистемному середовищі

1.2.1. Різноманітність форматів товарних даних

Однією з ключових проблем інтеграції товарних даних у торговельних підприємствах є різноманітність форматів, у яких виробники чи постачальники надають інформацію про свою продукцію. Найпоширенішими форматами обміну є XML, API, CSV та електронні таблиці Excel, PDF файли кожен з яких має власні особливості структури, обмеження та рівень стандартизації.

Формат XML часто використовується для передачі структурованих каталогів товарів, проте відсутність єдиного стандарту побудови XML-файлів призводить до значних відмінностей у схемах, назвах полів і способах представлення характеристик товарів. У результаті інтеграція таких даних потребує індивідуальної розробки парсерів та постійної адаптації у разі змін з боку постачальника.

API-доступ забезпечує більш гнучкий і оперативний обмін даними, однак кожен постачальник реалізує власну логіку роботи API, формати відповідей, механізми аутентифікації та обмеження на кількості запитів в певних часових рамках. Це ускладнює масштабування інтеграційних рішень і потребує додаткових ресурсів для підтримки стабільної роботи обміну даними.

Формати CSV та Excel є найбільш поширеними серед невеликих і середніх постачальників завдяки простоті їх використання. Водночас ці формати характеризуються відсутністю чіткої структури, неоднорідністю назв стовпців, використанням різних одиниць виміру та помилками введення даних, що значно ускладнює автоматизовану обробку та інтеграцію товарної інформації.

1.2.2. Дублювання та неузгодженість товарної інформації

Іншою суттєвою проблемою інтеграції товарних даних є дублювання та неузгодженість інформації, що виникає під час обробки даних з кількох джерел. Один і той самий товар може бути представлений різними назвами, артикульними кодами або наборами характеристик, що ускладнює його ідентифікацію та коректне зіставлення у внутрішніх системах підприємства.

Дублювання товарних позицій призводить до появи надлишкових записів у CRM та ERP-системах, що, в свою чергу, негативно впливає на якість каталогу товарів, помилки в обліку залишків і формування звітності. Неузгодженість даних між системами може проявлятися у відмінностях цін, статусів наявності та резерву, характеристиках товарів, а це, створює ризики помилок під час обробки замовлень та взаємодії з клієнтами, адже може бути продано інший товар, або товар, якого фактично уже немає в наявності.

Причинами виникнення таких проблем є відсутність єдиного ідентифікатора товару, різні правила ведення номенклатури у CRM та ERP-системах, а також ручні операції під час імпорту та коригування даних. У багатосистемному середовищі ці фактори призводять до накопичення помилок і зниження загальної якості інформації.

1.2.3. Вплив якості даних на бізнес-процеси

Якість товарних даних безпосередньо впливає на ефективність ключових бізнес-процесів торговельного підприємства. Некоректні або неповні дані можуть призводити до помилок у ціноутворенні, неправильного відображення наявності товарів, затримок у виконанні замовлень та зниження рівня клієнтського сервісу.

У сфері електронної комерції якість товарних даних визначає зручність пошуку та фільтрації товарів на сайті, це напряму впливає на конверсію та обсяги продажів. Відсутність, або неправильне заповнення товарних характеристик унеможливорює коректне використання властивостей товару і знижує інформативність товарних карток для кінцевого споживача.

Крім того, низька якість товарних даних ускладнює аналітичну обробку інформації, прогнозування попиту та прийняття управлінських рішень. У довгостроковій перспективі це призводить до втрат в операційній діяльності, зниження конкурентоспроможності та обмеження можливостей масштабування бізнесу.

Проблеми різноманітності форматів товарних даних, дублювання та неузгодженості товарної інформації, а також негативний вплив низькоякісних даних на бізнес-процеси зумовлюють необхідність пошуку більш ефективних підходів до внесення товарних даних. Традиційні методи не забезпечують належного рівня гнучкості та масштабованості, що створює передумови для застосування методів машинного навчання як інструменту підвищення якості та узгодженості товарної інформації у багатосистемному середовищі.

1.3. Методи машинного навчання для обробки та нормалізації товарних даних

1.3.1. Класифікація та вилучення атрибутів товарів

Обробка товарних даних у торговельних підприємствах часто пов'язана з аналізом неструктурованої або слабоструктурованої інформації, такої як назви товарів, текстові описи та переліки характеристик. Для автоматизації цих процесів застосовуються методи машинного навчання, спрямовані на класифікацію товарів та вилучення їх атрибутів.

Задача класифікації товарів полягає у віднесенні кожної товарної позиції до відповідної категорії на основі наявних текстових та числових характеристик. Для її розв'язання можуть використовуватися як традиційні алгоритми машинного навчання, так і сучасні підходи глибокого навчання. Вилучення атрибутів товарів передбачає автоматичне визначення значень таких

характеристик, як виробник, матеріал, колір, тип підключення або технічні параметри, з текстових описів та специфікацій.

Застосування методів машинного навчання дозволяє зменшити залежність від жорстко заданих правил і забезпечує більшу гнучкість у роботі з різноманітними джерелами товарної інформації. Це особливо актуально в умовах частих змін форматів даних постачальників і зростання обсягів оброблюваної інформації.

1.3.2. Методи обробки природної мови (NLP)

Методи обробки природної мови відіграють ключову роль у задачах аналізу текстових товарних даних. Вони дозволяють перетворювати текстову інформацію у формалізований вигляд, придатний для подальшої автоматизованої обробки та аналізу.

На початкових етапах NLP-обробки використовуються методи токенізації, лематизації та нормалізації тексту, які забезпечують усунення шуму, приведення слів до базової форми та уніфікацію написання. Далі можуть застосовуватися методи векторизації тексту, такі як Bag-of-Words або TF-IDF, що дозволяють представити текст у числовому вигляді.

Сучасні підходи до NLP базуються на використанні нейронних мереж і контекстних векторних представлень слів та фраз. Такі методи забезпечують урахування семантичного контексту, що є досить важливим для коректного зіставлення назв товарів, виявлення дублікатів і класифікації атрибутів. Використання NLP-методів дозволяє ефективно працювати з багатомовними даними та варіативністю термінології, характерною для товарних описів.

1.3.3. Попередньо навчені моделі та підхід transfer learning

Одним із нових напрямів застосування машинного навчання у задачах обробки товарних даних є використання попередньо навчених моделей та підходу transfer learning. Суть цього підходу полягає у використанні моделей, попередньо навчених на великих обсягах загальних даних, з подальшою адаптацією до конкретної предметної області.

Попередньо навчені моделі, зокрема мовні моделі глибокого навчання, здатні ефективно аналізувати текстові дані без необхідності створення великих навчальних вибірок для кожної окремої задачі. Адаптація таких моделей до обробки товарних даних здійснюється шляхом налаштування параметрів, використання контекстних підказок або поєднання результатів роботи моделі з бізнес-правилами підприємства.

Перевагами підходу transfer learning є скорочення часу впровадження рішень, зменшення витрат на навчання моделей та підвищення якості результатів у задачах класифікації, нормалізації та зіставлення товарних даних. У контексті інтеграції CRM та ERP-систем цей підхід дозволяє створювати масштабовані та адаптивні системи обробки інформації, здатні ефективно працювати з різнорідними джерелами даних.

Методи машинного навчання, зокрема класифікація, вилучення атрибутів, методи NLP та підхід transfer learning, є ефективним інструментом для обробки та інтеграції товарних даних у багатосистемне середовище. Їх застосування дозволяє автоматизувати процеси інтеграції товарної інформації, підвищити її якість та забезпечити гнучкість і масштабованість.

1.4. Використання великих мовних моделей у системах інтеграції даних

1.4.1. Архітектура LLM та принципи роботи

Великі мовні моделі (Large Language Models) є класом моделей глибокого навчання, побудованих на основі трансформерної архітектури, яка забезпечує ефективну обробку послідовностей текстових даних. Основним принципом роботи LLM є навчання на великих корпусах текстів з метою формування узагальнених мовних закономірностей, що дозволяють моделі виконувати різноманітні завдання обробки природної мови.

Архітектура LLM базується на механізмі самоуваги (self-attention), який забезпечує врахування контексту кожного елемента тексту відносно інших. Це дозволяє моделі аналізувати семантичні зв'язки між словами та фразами, що є критично важливим для задач зіставлення назв товарів, вилучення атрибутів та

класифікації товарної інформації. На відміну від традиційних NLP-методів, LLM не обмежуються локальними ознаками, а враховують контекст у межах усього тексту.

Завдяки багаторівневій структурі та великій кількості параметрів LLM здатні працювати з неструктурованими та слабоструктурованими даними, характерними для товарних описів, технічних характеристик і довільних форматів даних постачальників.

1.4.2. API-доступ до ML-моделей

У корпоративних інформаційних системах LLM зазвичай використовуються не шляхом локального розгортання, а через API-доступ до попередньо навчених моделей, що надаються спеціалізованими сервісами. Такий підхід дозволяє інтегрувати ML-функціональність у CRM, ERP та інші бізнес-системи без необхідності підтримки складної інфраструктури машинного навчання.

API-доступ до LLM забезпечує стандартизований механізм взаємодії, за якого система інтеграції даних передає на вхід моделі текстову інформацію (наприклад, назви товарів або їх описи) та отримує структурований результат у вигляді класифікацій, атрибутів або семантичних відповідностей. Це створює передумови для побудови масштабованих інтеграційних рішень, здатних обробляти великі обсяги товарних даних у режимі, близькому до реального часу.

Використання API також спрощує оновлення моделей, оскільки вдосконалення алгоритмів і параметрів відбувається на стороні постачальника сервісу, без необхідності змін у внутрішній архітектурі корпоративних систем.

1.4.3. Переваги та обмеження використання LLM у корпоративних системах

Застосування великих мовних моделей у системах інтеграції товарних даних має низку суттєвих переваг. До основних з них належать висока гнучкість у роботі з різномірними форматами даних, здатність до семантичного аналізу тексту та швидка адаптація до нових типів інформації без повного перенавчання

моделей. Це особливо важливо в умовах динамічного оновлення асортименту та постійних змін з боку виробників.

Водночас використання LLM у корпоративних системах має певні обмеження. До них належать залежність від зовнішніх сервісів, потенційні затримки обробки даних, а також необхідність контролю якості результатів, оскільки мовні моделі можуть генерувати ймовірнісні, але не завжди коректні відповіді. Крім того, питання інформаційної безпеки та конфіденційності даних потребують додаткової уваги при передачі товарної інформації через зовнішні API.

З огляду на зазначені переваги та обмеження, доцільним є використання LLM у поєднанні з традиційними методами обробки даних та бізнес-правилами підприємства. Такий гібридний метод забезпечує баланс між гнучкістю, точністю та надійністю інтеграційних рішень у корпоративному середовищі.

Великі мовні моделі є ефективним інструментом для обробки та інтеграції товарних даних. Використання API-доступу до попередньо навчених LLM дозволяє реалізувати масштабовані та адаптивні рішення, здатні працювати з неструктурованими даними в багатосистемному середовищі. Водночас для забезпечення надійності корпоративних систем необхідно враховувати обмеження LLM та застосовувати комбіновані підходи до обробки даних.

ВИСНОВОК ДО РОЗДІЛУ 1

У першому розділі магістерської роботи досліджено теоретичні основи інтеграції товарних даних у CRM та ERP-системах торговельних підприємств. Проаналізовано функціональне призначення та роль CRM-систем у керуванні товарною інформацією, а також визначено ключові особливості ERP-систем як основного джерела достовірних облікових даних. Обґрунтовано необхідність їх взаємодії в умовах багатоканального продажу для забезпечення узгодженості товарної інформації та ефективності бізнес-процесів.

У розділі виявлено основні проблеми інтеграції товарних даних у багато системному середовищі, зокрема різноманітність форматів даних різних постачальників, дублювання та неузгодженість товарної інформації, а також

негативний вплив низької якості даних на операційну діяльність підприємства. Показано, що традиційні rule-based підходи до інтеграції даних не забезпечують достатньої гнучкості та масштабованості в умовах зростання обсягів і різноманітності товарної інформації.

Проведено аналіз методів машинного навчання для обробки та нормалізації товарних даних. Розглянуто можливості класифікації товарів, вилучення атрибутів, застосування методів обробки природної мови та використання попередньо навчених моделей у межах підходу transfer learning. Доведено переваги та недоліки використання великих мовних моделей як інструменту семантичного аналізу неструктурованих товарних даних у системах інтеграції.

У результаті проведеного аналізу зроблено висновок, що застосування великих мовних моделей із використанням API-доступу створює передумови для побудови масштабованих і адаптивних систем інтеграції товарних даних у системах підприємств. Водночас визначено необхідність поєднання традиційних методів з методами машинного навчання для забезпечення надійності та контрольованості корпоративних рішень.

Отримані теоретичні результати дозволяють провести аналіз поточного стану управління товарними даними на конкретному торговельному підприємстві та розглянути розробку практичного рішення системи інтеграції товарних даних, що буде представлено в наступному розділі магістерської роботи.

РОЗДІЛ 2. АНАЛІЗ СИСТЕМИ УПРАВЛІННЯ ТОВАРНИМИ ДАНИМИ НА ТОРГОВЕЛЬНОМУ ПІДПРИЄМСТВІ

2.1. Характеристика торговельного підприємства та його інформаційної інфраструктури

2.1.1. Загальна характеристика підприємства

Об'єктом дослідження у даній магістерській роботі є торговельне підприємство, що здійснює діяльність у сфері оптової та роздрібної торгівлі з використанням багатоканальної моделі продажів. Підприємство реалізує продукцію через інтернет-магазин, роздрібні торгові точки, а також B2B-канали, що зумовлює необхідність ефективного управління великим обсягом товарної інформації.

Асортимент підприємства характеризується значною кількістю товарних позицій, які надходять від різних постачальників та регулярно оновлюються. Для забезпечення безперервності бізнес-процесів підприємство використовує декілька інформаційних систем, кожна з яких виконує окремі функції в управлінні товарними, фінансовими та операційними даними. Така багатосистемна архітектура дозволяє забезпечити гнучкість і масштабованість діяльності, проте водночас створює додаткові виклики щодо узгодженості та якості товарних даних.

2.1.2. Опис використовуваних систем: KeyCRM, BAS, вебсайт, Ultra

Для управління процесами продажу та взаємодії з клієнтами підприємство використовує CRM-систему KeyCRM. Дана система забезпечує облік товарних позицій, формування замовлень, управління цінами, знижками та наявністю товарів, а також інтеграцію з вебсайтом інтернет-магазину. У CRM зберігається товарна інформація, орієнтована на представлення продукції кінцевому споживачу.

Облікові та фінансові процеси підприємства реалізуються за допомогою ERP-системи BAS, яка використовується для ведення номенклатури товарів, контролю складських залишків, обліку руху товарів, закупівель та фінансових

операцій. ERP-система виступає основним джерелом достовірних даних щодо фактичної наявності товарів і їх облікової вартості.

Веб Сайт інтернет-магазину є ключовим каналом онлайн-продажів та забезпечує відображення каталогу товарів, пошук, фільтрацію та оформлення замовлень. Дані, що відображаються на вебсайті, формуються на основі інформації, отриманої з CRM-системи, та доповнені даними з бази сайту, вони повинні бути актуальними, повними й узгодженими.

Для управління роздрібними торговими точками підприємство використовує облікову систему Ultra, яка забезпечує фіксацію продажів у фізичних магазинах, облік залишків та формування звітності. Наявність окремої системи для офлайн-продажів додатково ускладнює процес синхронізації товарних даних між усіма інформаційними компонентами підприємства.

2.1.3. Структура товарних даних

Товарні дані підприємства формуються з декількох джерел та мають ієрархічну структуру. Основними складовими товарної інформації є ідентифікаційні дані (артикул, внутрішній код, SKU), описові атрибути (назва, бренд, категорія), цінова інформація, дані про наявність і залишки, а також технічні характеристики та мультимедійні матеріали.

У CRM-системі товарні дані орієнтовані на потреби продажу та маркетингу і включають розширені описи, характеристики, зображення та допоміжні атрибути. В ERP-системі структура товарних даних більш формалізована та спрямована на облік і контроль, що проявляється у жорсткій номенклатурі та регламентованих атрибутах. У системі роздрібної торгівлі та на веб сайті товарні дані можуть мати спрощену або адаптовану структуру, що створює додаткові розбіжності між системами.

Відсутність єдиної централізованої моделі товарних даних призводить до дублювання атрибутів, різних підходів до заповнення характеристик і складнощів у процесі синхронізації. Це обґрунтовує необхідність впровадження системи інтеграції товарних даних, здатної забезпечити їх уніфікацію,

нормалізацію та узгодженість між усіма інформаційними системами підприємства.

2.2. Аналіз процесу отримання та обробки товарних даних від постачальників

2.2.1. Джерела надходження товарної інформації

Товарна інформація на торговельному підприємстві формується на основі даних, що надходять від зовнішніх постачальників, а також з внутрішніх інформаційних систем. Основними джерелами товарних даних є прайс-листи та каталоги постачальників, що містять відомості про асортимент, ціни, технічні характеристики, наявність товарів і мультимедійні матеріали.

Залежно від рівня цифрової зрілості постачальників, товарні дані можуть надходити у вигляді структурованих електронних файлів або через програмні інтерфейси доступу. Частина постачальників забезпечує регулярне оновлення даних, тоді як інші надають інформацію з нерегулярною періодичністю, що ускладнює підтримку актуальності товарної інформації в інформаційних системах підприємства.

Отримані від постачальників дані після первинної обробки використовуються для наповнення каталогу товарів у CRM-системі, подальшої синхронізації з ERP-системою та відображення на вебсайті інтернет-магазину і в роздрібних торгових точках.

2.2.2. Формати даних та особливості їх обробки

Постачальники надають товарні дані у різних форматах, серед яких найпоширенішими є XML, API, CSV та файли Excel. Кожен із зазначених форматів має власні особливості, що впливають на процес їх обробки та інтеграції.

XML-файли зазвичай містять структуровану інформацію про товари, однак через відсутність єдиних стандартів формування XML-схем назви полів, вкладеність елементів та спосіб представлення характеристик суттєво

відрізняються між постачальниками. Це потребує створення індивідуальних механізмів парсингу та регулярної підтримки таких рішень.

API-доступ дозволяє отримувати товарні дані в режимі, наближеному до реального часу, проте супроводжується необхідністю обробки обмежень швидкості запитів, механізмів автентифікації та різних форматів відповідей. Крім того, стабільність обміну даними через API залежить від надійності інфраструктури постачальника.

CSV та Excel-файли є найбільш універсальними форматами, однак часто містять неструктуровані або неповні дані. До типових проблем належать відсутність єдиних назв стовпців, використання різних одиниць виміру, помилки введення та дублювання інформації. Це ускладнює автоматизовану обробку та потребує додаткових етапів очищення й нормалізації даних.

2.2.3. Поточний рівень автоматизації

Процес отримання та обробки товарних даних на підприємстві характеризується частковою автоматизацією. Для окремих постачальників реалізовано автоматизоване завантаження даних, зокрема через API або регулярний імпорт структурованих файлів. Водночас значна частина товарної інформації все ще обробляється з використанням напівавтоматичних або ручних процедур.

Ручне втручання необхідне для перевірки коректності даних, зіставлення товарних позицій, заповнення відсутніх атрибутів і усунення дублікатів. Це призводить до збільшення часових витрат на обробку інформації, підвищення ймовірності помилок та зниження оперативності оновлення товарного каталогу.

Поточний рівень автоматизації не забезпечує повної узгодженості товарних даних між CRM, ERP, вебсайтом та роздрібними системами. Відсутність централізованого механізму інтелектуальної обробки даних обмежує можливості масштабування та ускладнює інтеграцію нових постачальників, що підтверджує доцільність впровадження системи інтеграції товарних даних із використанням методів машинного навчання.

2.3. Проблеми формування товарних характеристик і атрибутів у CRM та на сайті

2.3.1. Ручне заповнення характеристик

У поточній практиці торговельного підприємства формування товарних характеристик і атрибутів у CRM та на вебсайті інтернет-магазину значною мірою здійснюється вручну. Незважаючи на наявність структурованих даних від постачальників, значна частина товарних характеристик подається у вигляді неструктурованого тексту, що потребує додаткового аналізу та інтерпретації з боку співробітників.

Ручне заповнення характеристик передбачає аналіз назв товарів, описів і технічних характеристик з метою визначення відповідних значень атрибутів, таких як виробник, матеріал, колір, тип підключення або функціональні особливості. Цей процес є трудомістким, залежить від суб'єктивних рішень виконавців і не забезпечує стабільної якості результатів, особливо в умовах великого асортименту та частих оновлень каталогу.

Крім того, ручна обробка товарних даних ускладнює масштабування бізнесу, оскільки підключення нових постачальників або розширення асортименту призводить до пропорційного зростання навантаження на персонал, що негативно впливає на оперативність наповнення та актуалізації каталогу товарів.

2.3.2. Помилки та неузгодженість даних

Ручні методи формування товарних характеристик неминуче призводять до виникнення помилок та неузгодженості даних. Одні й ті самі характеристики товарів можуть бути заповнені з використанням різних варіантів написання, скорочень або значень, що ускладнює їх подальшу обробку та аналіз.

Неузгодженість даних проявляється у відмінностях між інформацією, що зберігається у CRM, ERP та на вебсайті інтернет-магазину. Наприклад, відсутність або некоректне заповнення характеристик у CRM може призводити до неправильного відображення товарів у каталозі сайту, а також до

розбіжностей із обліковими даними ERP-системи. Це створює додаткові складнощі при синхронізації інформації між системами та підвищує ризик помилок під час обробки замовлень.

Такі помилки потребують подальшого ручного виправлення, що збільшує витрати часу та ресурсів і знижує загальну ефективність процесів управління товарними даними.

2.3.3. Вплив на користувацький досвід та продажі

Якість формування товарних характеристик безпосередньо впливає на користувацький досвід відвідувачів вебсайту інтернет-магазину. Неповні або некоректно заповнені фільтри ускладнюють пошук товарів, знижують зручність навігації та можуть призводити до втрати потенційних клієнтів.

У ситуації, коли користувач не може швидко знайти товар із потрібними характеристиками або стикається з некоректною інформацією, зростає ймовірність відмови від покупки. Це негативно впливає на показники конверсії та загальний обсяг продажів в умовах високої конкуренції.

Крім того, неякісні товарні дані ускладнюють використання аналітичних інструментів, персоналізованих рекомендацій і маркетингових кампаній, що обмежує можливості підвищення ефективності продажів. Недоліки у формуванні товарних характеристик і атрибутів впливають як на щоденні операційні процеси підприємства, так і на його фінансові результати.

2.4. Обґрунтування доцільності застосування методів машинного навчання

2.4.1. Порівняння rule-based та ML-підходів

Традиційні підходи до інтеграції та обробки товарних даних у торговельних підприємствах здебільшого базуються на rule-based методах, які передбачають використання заздалегідь визначених правил, шаблонів і регулярних виразів. Такі підходи є ефективними для обробки стандартизованих і стабільних форматів даних, однак мають суттєві обмеження в умовах різноманітності джерел і динамічних змін товарної інформації.

Rule-based системи потребують постійної підтримки та оновлення правил у разі зміни структури даних постачальників або появи нових типів товарів. Крім того, такі системи не здатні ефективно працювати з неструктурованими текстовими даними, що знижує їхню точність при формуванні товарних атрибутів і характеристик.

На відміну від цього, методи машинного навчання забезпечують можливість автоматичного виявлення закономірностей у даних і адаптації до нових умов без необхідності жорсткого програмування кожного сценарію. Використання ML-підходів дозволяє працювати з варіативними назвами товарів, описами та характеристиками, забезпечуючи більш високу точність класифікації та зіставлення даних.

2.4.2. Очікувані переваги автоматизації

Впровадження методів машинного навчання в процесі обробки та інтеграції товарних даних дозволить отримати важливі переваги для торговельного підприємства. Передусім автоматизація зменшить обсяг ручної роботи, пов'язаної з аналізом та заповненням товарних атрибутів, що посприє скороченню часових витрат і зниженню операційних витрат.

Застосування ML-методів також підвищить якість товарних даних завдяки більш точному вилученню атрибутів і зменшенню кількості помилок, спричинених людським фактором. Це, у свою чергу, позитивно вплине на користувацький досвід, покращить навігацію та пошук товарів на вебсайті, а також посприє зростанню конверсії та обсягів продажів.

Додатковою перевагою буде масштабованість запропонованого підходу. Машинне навчання дозволить ефективно обробляти зростаючі обсяги товарних даних і спростить інтеграцію нових постачальників без пропорційного збільшення навантаження на персонал.

2.4.3. Вибір технологічного підходу

З урахуванням особливостей інформаційної інфраструктури торговельного підприємства та характеру товарних даних доцільним є

використання гібридного технологічного підходу, що поєднує методи машинного навчання з традиційними rule-based механізмами. Такий підхід дозволяє використовувати переваги ML для роботи з неструктурованими та варіативними даними, водночас забезпечуючи контроль і передбачуваність результатів за рахунок бізнес-правил.

Оптимальним рішенням є застосування попередньо навчених моделей машинного навчання з доступом через API, що дозволяє інтегрувати інтелектуальну обробку даних у CRM та ERP-системи без розгортання складної інфраструктури. Поєднання ML-обробки з механізмами валідації та перевірки результатів забезпечує баланс між гнучкістю, точністю та надійністю системи інтеграції товарних даних.

Обраний технологічний підхід створює передумови для побудови масштабованої системи інтеграції товарних даних, здатної адаптуватись до змін у форматах даних постачальників та зростання асортименту продукції.

ВИСНОВОК ДО РОЗДІЛУ 2

У другому розділі проаналізовано управління товарними даними на торговельному підприємстві, що функціонує в умовах багато системної інформаційної інфраструктури. Досліджено особливості використання CRM, ERP, веб сайту інтернет-магазину та облікових систем роздрібної торгівлі, а також визначено їх роль у формуванні та використанні товарної інформації.

Аналіз показав, що дані надходять із різних джерел постачальників у різноманітних форматах, зокрема XML, API, CSV та Excel, що ускладнює процес їх автоматизованої обробки та інтеграції. Встановлено, що поточний рівень автоматизації є недостатнім для забезпечення повної узгодженості та актуальності товарних даних, а значна частина дій виконується вручну.

Окрему увагу приділено проблемам формування товарних характеристик і атрибутів у CRM та на веб сайті інтернет-магазину. Виявлено, що переважання ручних методів заповнення генерує помилки, дублі та неузгодженості інформації

між системами, і негативно впливає на досвід покупців, ефективність пошуку товарів і показники продажів.

На основі порівняння rule-based та методів машинного навчання обґрунтовано доцільність використання ML-підходів для автоматизації обробки товарних даних. Визначено, що застосування машинного навчання дозволяє підвищити якість товарної інформації, зменшити обсяг ручної роботи та забезпечити масштабованість інтеграційних рішень у разі зростання асортименту та кількості постачальників.

Отримані результати аналізу підтверджують необхідність розробки та впровадження системи інтеграції товарних даних із використанням методів машинного навчання. Виявлені проблеми та сформульовані вимоги слугують підґрунтям для проєктування архітектури та реалізації практичного рішення, що буде представлено в третьому розділі магістерської роботи.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ ІНТЕГРАЦІЇ ТОВАРНИХ ДАНИХ ІЗ ЗАСТОСУВАННЯМ OPENAI API

3.1. Архітектура системи інтеграції товарних даних

3.1.1. Загальна схема взаємодії компонентів

Запропонована система інтеграції товарних даних побудована за модульним принципом та орієнтована на роботу в багатосистемному середовищі торговельного підприємства. Архітектура системи передбачає розділення функціональних компонентів за етапами обробки даних, що забезпечує масштабованість, гнучкість та можливість подальшого розширення.

До основних компонентів системи належать:

- модуль збору товарних даних від постачальників;
- модуль попередньої обробки та нормалізації даних;
- модуль інтелектуальної обробки з використанням API-доступу до великих мовних моделей;
- інтеграційний шар взаємодії з CRM та ERP-системами;
- цільові інформаційні системи підприємства.

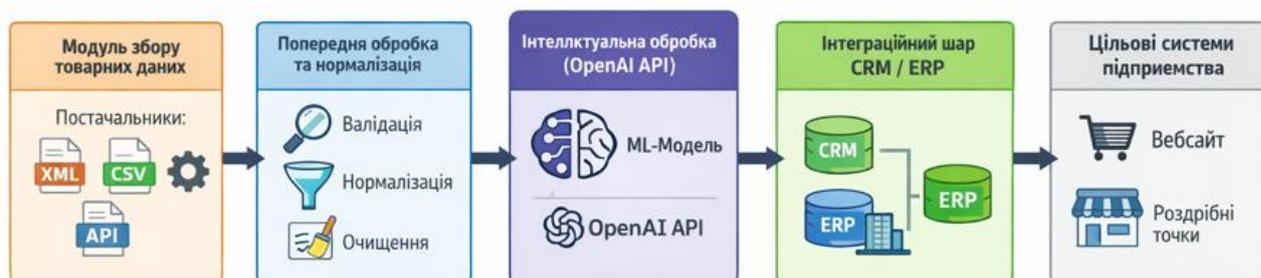


Рис. 3.1. Архітектура системи інтеграції товарних даних

На рисунку 3.1 представлено загальну архітектуру системи інтеграції товарних даних торговельного підприємства з використанням методів машинного навчання та API-доступу до великих мовних моделей. Архітектура є модульною і охоплює всі основні етапи обробки товарної інформації — від

отримання даних від постачальників до їх використання в корпоративних інформаційних системах.

У лівій частині схеми розташовано зовнішні джерела товарних даних, до яких належать постачальники, що надають інформацію у форматах XML, CSV, Excel або через API. Дані з цих джерел мають різну структуру та рівень стандартизації, що зумовлює необхідність їх подальшої уніфікації.

Центральним елементом архітектури є модуль, який збирає та здійснює попередню обробку даних, він виконує отримання товарної інформації, її первинний аналіз, очищення та приведення до уніфікованого формату. На цьому етапі здійснюється усунення структурних відмінностей, нормалізація значень атрибутів і підготовка даних до інтелектуальної обробки.

Наступним компонентом є модуль інтелектуальної обробки, який використовує API-доступ до великих мовних моделей для семантичного аналізу товарних даних. Даний модуль забезпечує автоматизоване вилучення та класифікацію товарних атрибутів, формування характеристик і підготовку структурованих результатів у форматі, придатному для подальшої інтеграції.

Результати роботи модуля машинного навчання передаються до інтеграційного шару, який забезпечує взаємодію між системою інтеграції та внутрішніми інформаційними системами підприємства. Інтеграційний шар відповідає за зіставлення атрибутів, контроль коректності даних і передачу інформації у відповідні цільові системи.

У правій частині схеми розташовано цільові інформаційні системи підприємства, до яких належать CRM-система, ERP-система, вебсайт інтернет-магазину та система обліку роздрібних торгових точок.

Стрілки потоків даних на рисунку відображають послідовний та однонаправлений рух інформації між компонентами системи. Така організація архітектури забезпечує контрольованість процесів, можливість валідації даних на кожному етапі та створює передумови для масштабування системи у разі збільшення кількості постачальників або обсягу товарних даних.

Загальна логіка роботи системи полягає у поетапному проходженні товарних даних від зовнішніх джерел до внутрішніх інформаційних систем

підприємства з автоматизованою перевіркою, очищенням і формуванням структурованих атрибутів. Такий підхід дозволяє централізовано керувати обробкою товарної інформації та зменшити залежність від ручних операцій.

3.1.2. Модуль збору та попередньої обробки даних

Модуль збору та попередньої обробки даних забезпечує отримання товарної інформації від постачальників у різних форматах, зокрема XML, CSV, Excel та через API-доступ та їх обробку. Для кожного типу джерела реалізовано окремі механізми імпорту, що дозволяє уніфікувати процес завантаження даних незалежно від формату.

На етапі попередньої обробки виконується:

- перевірка цілісності та коректності вхідних даних;
- приведення назв полів і значень до єдиного формату;
- очищення текстових даних від зайвих символів і шуму;
- базова нормалізація одиниць виміру та числових значень.

Результатом роботи модуля є уніфікований набір товарних даних, підготовлений для подальшої інтелектуальної обробки. Відокремлення етапу попередньої обробки дозволяє зменшити навантаження на модуль машинного навчання та підвищити загальну надійність системи.

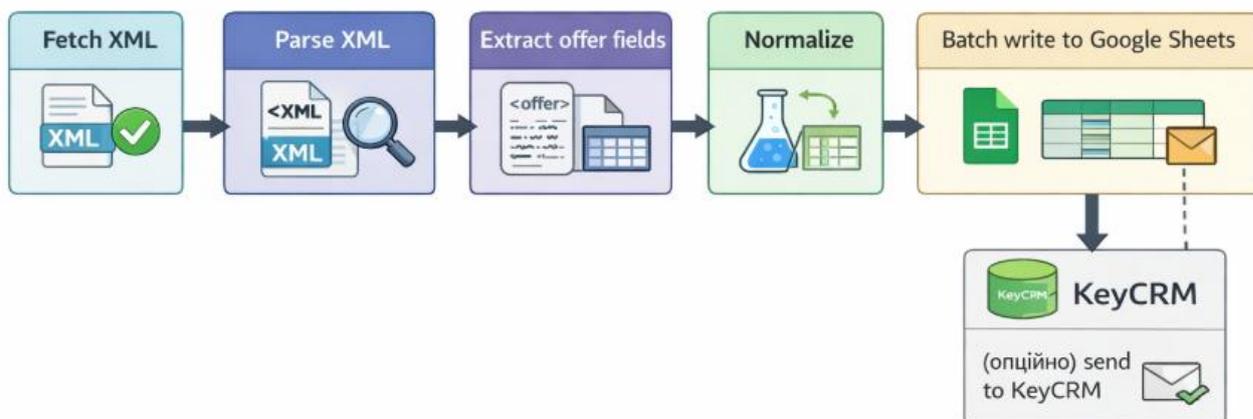


Рис. 3.2. Опис компонентів та потоків даних

На рисунку 3.2 представлено послідовність обробки XML-прайсу постачальника у межах системи інтеграції товарних даних. Процес реалізовано у

вигляді конвеєра (pipeline), що складається з логічно пов'язаних компонентів, між якими здійснюється послідовна передача даних.

- Модуль Fetch XML виконує отримання XML-файлу прайсу з зовнішнього джерела постачальника за допомогою HTTP-запиту. На цьому етапі відбувається завантаження актуальної версії даних без їх модифікації. Потік даних передається у вигляді сирого XML-документа до наступного етапу.
- Модуль Parse XML відповідає за синтаксичний розбір XML-документа та побудову внутрішнього об'єктного представлення даних. На цьому етапі перевіряється коректність структури XML та доступність ключових вузлів, таких як <offers> та <offer>.
- Модуль Extract offer fields виконує вибірку необхідних полів із кожного елемента <offer>, зокрема ідентифікаторів товару, назв, цін, залишків, зображень і довільних параметрів <param>. Результатом роботи модуля є набір записів у структурованому вигляді.
- Модуль Normalize здійснює попередню обробку та нормалізацію даних, включаючи приведення форматів числових значень, очищення текстових полів, уніфікацію одиниць виміру та підготовку атрибутів до подальшої інтеграції. Саме на цьому етапі усувається неоднорідність даних, спричинена різними стандартами постачальників.
- Модуль Batch write to Google Sheets забезпечує пакетний запис оброблених товарних даних у Google Таблиці. Використання пакетного режиму зменшує кількість API-запитів і підвищує продуктивність системи при обробці великих обсягів товарної інформації.
- Опційний модуль send to KeyCRM виконує передачу структурованих товарних даних до CRM-системи для подальшого використання у процесах продажу та управління каталогом. Передача здійснюється лише після успішного завершення попередніх етапів обробки.

3.1.2. Інтеграційний шар CRM та ERP

Інтеграційний шар забезпечує взаємодію системи інтеграції товарних даних із внутрішніми інформаційними системами підприємства, зокрема CRM та ERP. Його основним завданням є передача структурованих і нормалізованих товарних даних у відповідні системи з урахуванням їхніх функціональних особливостей.

У межах запропонованого рішення CRM-система KeyCRM використовується як основний інструмент роботи з товарною інформацією, орієнтованою на продажі та взаємодію з клієнтами. Інтеграційний шар забезпечує автоматичне оновлення товарних атрибутів, характеристик, цін та статусів наявності у CRM на основі результатів обробки даних.

ERP-система BAS використовується для обліку номенклатури, залишків і фінансових показників. Інтеграція з ERP передбачає синхронізацію ключових ідентифікаційних атрибутів товарів, що дозволяє забезпечити узгодженість між обліковими та операційними даними.

Інтелектуальна обробка товарних даних у запропонованій архітектурі реалізується з використанням API-доступу до попередньо навчених моделей машинного навчання, що надаються сервісом OpenAI. Отримані результати використовуються для автоматизованого формування товарних атрибутів, які після валідації передаються до CRM та ERP-систем.

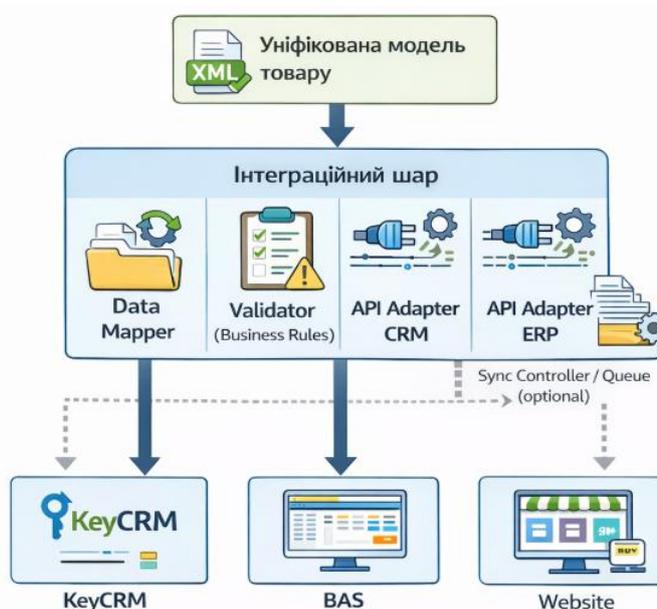


Рис. 3.3 Інтеграційний шар взаємодії CRM та ERP-систем

На рисунку 3.3 зображено структуру інтеграційного шару, який забезпечує узгоджену взаємодію між уніфікованою моделлю товару та корпоративними інформаційними системами торговельного підприємства. Інтеграційний шар виконує функцію проміжного рівня, що відокремлює логіку обробки та нормалізації даних від бізнес-систем CRM та ERP.

Вхідними даними для інтеграційного шару є уніфікована модель товару, сформована на попередніх етапах обробки товарних даних постачальників. Потік даних надходить до інтеграційного шару, де послідовно проходить через модулі зіставлення полів (Data Mapper) та перевірки бізнес-правил (Validator). На цих етапах здійснюється приведення структури товарних атрибутів до вимог цільових систем і контроль коректності значень перед записом.

Подальша передача даних здійснюється через окремі адаптери доступу до API CRM та ERP-систем (API Adapter CRM, API Adapter ERP). Використання окремих адаптерів дозволяє враховувати специфіку кожної системи, зокрема формати даних, обмеження API та вимоги до авторизації, не змінюючи основну логіку інтеграції.

Опційний компонент Sync Controller / Queue забезпечує керування чергами синхронізації та дозволяє виконувати асинхронну або пакетну передачу даних, що підвищує надійність системи у разі великих обсягів товарної інформації або тимчасової недоступності цільових систем.

У результаті роботи інтеграційного шару структуровані та перевірені товарні дані передаються до CRM-системи, ERP-системи та вебсайту інтернет-магазину. Стрілки на рисунку відображають напрямки основних потоків даних, а також допоміжні зворотні потоки, що використовуються для обробки статусів синхронізації та помилок.

Таким чином, представлений на рисунку інтеграційний шар забезпечує централізовану, контрольовану та масштабовану інтеграцію товарних даних у багато системному інформаційному середовищі торговельного підприємства.

3.2. Реалізація модуля автоматизованої обробки товарних даних

3.2.1 Структура вхідних даних

Модуль автоматизованої обробки товарних даних працює з інформацією, що надходить із різних зовнішніх джерел постачальників у форматах XML, CSV, Excel та через API. Незважаючи на відмінності у форматах подання, вхідні дані зазвичай містять схожі за змістом інформаційні елементи, зокрема ідентифікатори товарів, назви, цінові параметри, характеристики та описи.

Для забезпечення уніфікованої обробки всі вхідні дані приводяться до проміжного представлення, яке включає такі основні поля:

- ідентифікаційні дані (артикул постачальника, SKU, внутрішній код);
- текстові атрибути (назва товару, опис, бренд);
- структуровані характеристики (матеріал, колір, розміри, технічні параметри);
- цінова інформація та дані про наявність;
- допоміжні дані (категорія, зображення, службові ознаки).

Наявність такого проміжного шару дозволяє абстрагуватися від специфіки конкретного джерела даних і забезпечує єдиний підхід до подальшої обробки товарної інформації.

Приклад “Сирих” даних в форматі XML:

```
<yml_catalog date="2025-12-12 00:40">
<shop>
<name>www.UA</name>
<currencies>
<currency id="UAH" rate="1"/>
</currencies>
<categories>
...
</categories>
<offers>
<offer id="2658" available="true">
  <url>https://www.ua/catalog/2658</url>
  <old_price/>
  <price>71.00</price>
  <currencyId>UAH</currencyId>
```

```

<categoryId>103</categoryId>
<barcode>5997270790135</barcode>
<name>Ароматизатор Paloma Parfume Tabac жидкий</name>
<name_ua>Ароматизатор Paloma Parfume Tabac рідкий</name_ua>
<vendorCode>79013</vendorCode>
<description>
<![CDATA[ Широко признанный и популярный продукт с отличной ценой.
Производится с широким выбором ароматов и включает в себя жидкий
парфюмерный концентрат, который покупатели могут распределять в
соответствии со своими требованиями в силе аромата. ]]>
</description>
<description_ua>
<![CDATA[ Широко визнаний і популярний продукт з відмінною ціною.
Виробляється з широким вибором ароматів і включає в себе рідкий
парфюмерний концентрат, який покупці можуть розподіляти відповідно до
своїх вимог в силі аромату. ]]>
</description_ua>
<stock_quantity>20</stock_quantity>
<minimum_order_quantity>1</minimum_order_quantity>
<available>true</available>
<picture>https://www.ua/uploads/cache/CatalogItems/CatalogItems2658/0859f
63b69-1_resize_1200x1200.jpeg</picture>
<vendor>Paloma</vendor>
<param name="Тип">Рідкі</param>
<param name="Аромат">Інший</param>
<param name="УКТЗЕД">3307 49 00 00</param>
</offer>
<offers>
</shop>
</yml_catalog>

```

Таблиця 3.1

Структура вхідних даних

XML поле	Приклад	Unified field	Тип	Коментар
offer/@id	2658	external_id	string	Ідентифікатор у постачальника
offer/@available	TRUE	supplier_available	bool	Ознака доступності
url	https://www.ua/catalog/2658	product_url	string	Посилання на товар

price	71.00	price	number	Поточна ціна
old_price	90.00	old_price	number/null	Стара ціна/знижка
currencyId	UAH	currency	string	Валюта
categoryId	103	supplier_category_id	string/int	Категорія постачальника
barcode	5997...	barcode	string	EAN
name	Ароматизатор	name_ru	string	Назва RU
name_ua	Ароматизатор	name_ua	string	Назва UA
vendorCode	79013	vendor_code	string	Артикул/код
vendor	Paloma	brand	string	Виробник/бренд
stock_quantity	20	stock_qty	int	Залишок
picture	...jpeg	images[]	array	Масив URL
param[@name]	“Тип”=“Рідкий”	attributes[Тип]	map	Довільні атрибути

3.2.2. Нормалізація та підготовка даних

На етапі нормалізації та підготовки даних здійснюється комплекс дій, необхідних для підвищення якості та узгодженості вхідної інформації. Основними завданнями цього етапу є усунення неоднорідності даних, зменшення рівня шуму та приведення значень атрибутів до єдиних форматів.

Процес нормалізації включає:

- стандартизацію назв атрибутів і значень (усунення різних варіантів написання);
- очищення текстових полів від спеціальних символів, службових слів і дублюючої інформації;
- уніфікацію одиниць виміру та числових форматів;
- виявлення та усунення очевидних помилок і пропусків у даних.

Підготовлені дані після нормалізації передаються до модуля інтелектуальної обробки, де використовуються методи машинного навчання для семантичного аналізу та автоматичного вилучення атрибутів. Такий підхід дозволяє значно зменшити залежність від ручної обробки та підвищити стабільність результатів у разі змін у форматах даних постачальників.



Рис. 3.4 Процес нормалізації товарних даних

На рисунку 3.4 представлено процес нормалізації та підготовки товарних даних, що отримані із зовнішніх джерел у форматі XML. Метою даного етапу є приведення різномірної вхідної інформації до уніфікованого, структурованого та придатного для подальшої автоматизованої обробки вигляду.

Початковим елементом схеми є блок «Сирі дані XML», який містить необроблену товарну інформацію постачальника, включаючи текстові описи, числові значення та довільні атрибути. Дані з цього блоку передаються до етапу

«Очищення тексту», де виконується видалення службових символів, HTML-розмітки, зайвих пробілів та дублюючої інформації.

Наступним етапом є «Стандартизація атрибутів», на якому здійснюється приведення назв полів і значень до єдиних стандартів шляхом усунення різних варіантів написання та використання довідників відповідностей. Це дозволяє зменшити неоднорідність даних, що виникає через відмінності у форматах постачальників.

Після цього виконується «Уніфікація форматів», яка передбачає нормалізацію числових значень, валют, одиниць виміру та форматів дат. На даному етапі забезпечується коректність представлення даних для подальшого збереження та аналізу.

Завершальним етапом є «Перевірка та заповнення пропусків», під час якого виявляються відсутні або некоректні значення атрибутів і виконуються операції їхнього виправлення або маркування для подальшої обробки. Результатом виконання всіх етапів є «Нормалізовані дані», готові до передачі в модуль інтелектуальної обробки та подальшої інтеграції в CRM та ERP-системи.

Таким чином, наведена на рисунку схема демонструє поетапний і контрольований процес підготовки товарних даних, що забезпечує підвищення якості інформації та стабільність подальших етапів автоматизованої обробки.

```
function normalizeAndPrepare_www_XML() {
  const XML_URL =
  "https://opt.www.ua/data/ff764b8ef412d7090041a00ff8e0b187.xml";
  const SHEET_NAME = "www_Normalized";
  const BATCH_SIZE = 150;
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const sh = ss.getSheetByName(SHEET_NAME) ||
  ss.insertSheet(SHEET_NAME);

  // Заголовки
  const header = [
    "external_id",
    "vendor_code",
    "barcode",
    "brand",
    "name",
    "name_ua",
    "description",
```

```

    "description_ua",
    "category_id",
    "currency",
    "price",
    "old_price",
    "stock_qty",
    "min_order_qty",
    "available_attr",
    "available_tag",
    "product_url",
    "picture",
    "params_json",
    "normalized_at"
];

if (sh.getLastRow() === 0) {
  sh.getRange(1, 1, 1, header.length).setValues([header]);
}

const props = PropertiesService.getScriptProperties();
const lastIndex =
parseInt(props.getProperty("www_NORM_LAST_INDEX") || "0", 10);

const resp = UrlFetchApp.fetch(XML_URL, { muteHttpExceptions: true });
if (resp.getResponseCode() !== 200) {
  throw new Error("Failed to fetch XML. HTTP=" +
resp.getResponseCode());
}

const xml = XmlService.parse(resp.getContentText());
const root = xml.getRootElement(); // <yml_catalog>
const offers = root.getChild("shop").getChild("offers").getChildren("offer");

const endIndex = Math.min(lastIndex + BATCH_SIZE, offers.length);
const rows = [];
const nowIso = new Date().toISOString();

for (let i = lastIndex; i < endIndex; i++) {
  const offer = offers[i];

  // 1) Extract raw fields
  const raw = extractOfferRaw_(offer);

  // 2) Normalize & prepare
  const unified = buildUnifiedProduct_(raw);

```

```

// 3) Convert to row
rows.push([
  unified.external_id,
  unified.vendor_code,
  unified.barcode,
  unified.brand,
  unified.name,
  unified.name_ua,
  unified.description,
  unified.description_ua,
  unified.category_id,
  unified.currency,
  unified.price,
  unified.old_price,
  unified.stock_qty,
  unified.min_order_qty,
  unified.available_attr,
  unified.available_tag,
  unified.product_url,
  unified.picture,
  JSON.stringify(unified.params || {}),
  nowIso
]);
}

// Batch write
if (rows.length) {
  const startRow = sh.getLastRow() + 1;
  sh.getRange(startRow, 1, rows.length, header.length).setValues(rows);
}

// Save progress
props.setProperty("www_NORM_LAST_INDEX", String(endIndex));

// Finish
if (endIndex >= offers.length) {
  props.deleteProperty("www_NORM_LAST_INDEX");
}
}

/** ----- Extraction ----- */

function extractOfferRaw_(offer) {
  const getAttr = (name) => offer.getAttribute(name)?.getValue() || "";

  return {

```

```

external_id: getAttr("id"),
available_attr: getAttr("available"),

product_url: getText_(offer, "url"),
old_price: getText_(offer, "old_price"),
price: getText_(offer, "price"),
currency: getText_(offer, "currencyId"),
category_id: getText_(offer, "categoryId"),
barcode: getText_(offer, "barcode"),
name: getText_(offer, "name"),
name_ua: getText_(offer, "name_ua"),
vendor_code: getText_(offer, "vendorCode"),
description: getCdataText_(offer, "description"),
description_ua: getCdataText_(offer, "description_ua"),
stock_qty: getText_(offer, "stock_quantity"),
min_order_qty: getText_(offer, "minimum_order_quantity"),
available_tag: getText_(offer, "available"),
picture: getText_(offer, "picture"),
params: extractParams_(offer)
};
}

function getText_(parent, tagName) {
  const el = parent.getChild(tagName);
  if (!el) return "";
  return String(el.getText() || "").trim();
}

function getCdataText_(parent, tagName) {
  const el = parent.getChild(tagName);
  if (!el) return "";
  return cleanText_(String(el.getText() || ""));
}

function extractParams_(offerEl) {
  const params = offerEl.getChildren("param");
  const map = {};
  params.forEach(p => {
    const key = p.getAttribute("name") ?
String(p.getAttribute("name").getValue()).trim() : "";
    const val = String(p.getText() || "").trim();
    if (key) map[key] = val;
  });
  return map;
}

```

```
/** ----- Normalization & Preparation ----- **/
```

```
function buildUnifiedProduct_(raw) {
  // Normalize textual fields
  const name = normalizeName_(raw.name);
  const nameUa = normalizeName_(raw.name_ua);

  // Normalize numbers
  const price = normalizeNumber_(raw.price);
  const oldPrice = normalizeNullableNumber_(raw.old_price);
  const stockQty = normalizeInt_(raw.stock_qty);
  const minOrder = normalizeInt_(raw.min_order_qty);

  // Normalize currency
  const currency = normalizeCurrency_(raw.currency);

  // Normalize brand
  const brand = normalizeBrand_(raw.params, raw);

  // Normalize params (standardize keys/values + clean)
  const params = normalizeParams_(raw.params);

  // Clean descriptions
  const description = cleanText_(raw.description);
  const descriptionUa = cleanText_(raw.description_ua);

  // Basic validation / fixes
  const fixedOldPrice = (oldPrice !== null && oldPrice <= price) ? null :
oldPrice;
  const fixedStock = Math.max(0, stockQty);

  return {
    external_id: String(raw.external_id || "").trim(),
    vendor_code: normalizeSku_(raw.vendor_code),
    barcode: normalizeBarcode_(raw.barcode),
    brand: brand,
    name: name,
    name_ua: nameUa,
    description: description,
    description_ua: descriptionUa,
    category_id: String(raw.category_id || "").trim(),
    currency: currency,
    price: price,
    old_price: fixedOldPrice,
    stock_qty: fixedStock,
    min_order_qty: Math.max(1, minOrder || 1),
```

```

    available_attr: normalizeBoolStr_(raw.available_attr),
    available_tag: normalizeBoolStr_(raw.available_tag),
    product_url: normalizeUrl_(raw.product_url),
    picture: normalizeUrl_(raw.picture),
    params: params
  };
}

/** ----- Normalization Helpers ----- */

function cleanText_(text) {
  return String(text || "")
    .replace(/<[^>]*>/g, " ") // remove HTML tags
    .replace(/&nbsp;/g, " ")
    .replace(/\s+/g, " ")
    .trim();
}

function normalizeName_(name) {
  return cleanText_(name)
    .replace(/["']/g, "\"")
    .replace(/'/g, "\"")
    .trim();
}

function normalizeNumber_(v) {
  const s = String(v || "").trim();
  const n = parseFloat(s.replace(",", "."));
  return isNaN(n) ? 0 : n;
}

function normalizeNullableNumber_(v) {
  const s = String(v || "").trim();
  if (!s) return null;
  const n = parseFloat(s.replace(",", "."));
  return isNaN(n) ? null : n;
}

function normalizeInt_(v) {
  const s = String(v || "").trim();
  if (!s) return 0;
  const n = parseInt(s.replace(",", "."), 10);
  return isNaN(n) ? 0 : n;
}

function normalizeCurrency_(cur) {

```

```

const c = String(cur || "").trim().toUpperCase();
const allowed = { "UAH": true, "USD": true, "EUR": true };
return allowed[c] ? c : "UAH";
}

```

```

function normalizeBoolStr_(v) {
  const s = String(v || "").trim().toLowerCase();
  if (s === "true" || s === "1" || s === "yes" || s === "так") return "true";
  if (s === "false" || s === "0" || s === "no" || s === "ні") return "false";
  return s ? "true" : "false"; // якщо поле є, але нестандартне
}

```

```

function normalizeSku_(sku) {
  return String(sku || "").trim();
}

```

```

function normalizeBarcode_(barcode) {
  const b = String(barcode || "").trim();
  const digits = b.replace(/\D/g, "");
  return digits || b;
}

```

```

function normalizeUrl_(url) {
  const u = String(url || "").trim();
  if (!u) return "";
  if (!/^https?:\:\/\/i.test(u)) return "";
  return u;
}

```

```

function normalizeBrand_(params, raw) {
  const vendor = String(raw.vendor || "").trim();
  if (vendor) return vendor;
  if (params && params["Виробник"]) return
String(params["Виробник"]).trim();
  return "";
}

```

```

function normalizeParams_(params) {
  const out = {};
  const KEY_MAP = {
    "Тип": "Тип",
    "Аромат": "Аромат",
    "УКТЗЕД": "УКТЗЕД",
    "Код УКТЗЕД": "УКТЗЕД"
  };
}

```

```

Object.keys(params || {}).forEach(k => {
  const key = String(k).trim();
  const mappedKey = KEY_MAP[key] || key;
  const val = cleanText_(params[key]);

  out[mappedKey] = normalizeParamValue_(mappedKey, val);
});

return out;
}

function normalizeParamValue_(key, val) {
  if (!val) return "";

  if (key === "Тип") {
    const v = val.toLowerCase();
    if (v.includes("рідк")) return "Рідкі";
    if (v.includes("гел")) return "Гелеві";
    return capitalizeFirst_(val);
  }

  if (key === "УКТЗЕД") {
    return val.replace(/s+/g, " ").trim();
  }

  return val;
}

function capitalizeFirst_(s) {
  const str = String(s || "").trim();
  if (!str) return "";
  return str.charAt(0).toUpperCase() + str.slice(1);
}

```

Після виконання отримуємо таблицю :

Таблиця 3.2

Таблиця нормалізованих даних

external_id	vendor_code	barcode	brand	name_u a	descript ion	categor y_id	currenc y	price	old_pri ce	stock_q ty	min_or der_qty	availabl e_attr	availabl e_tag	product _url	picture	params _json	normali zed_at
2658	79013	599727 079013 5	Paloma	Аромат изатор Paloma Parfum e Tabac рідкий	Опис товару	103	UAH	71	95	20	1	TRUE	TRUE	www	www	{"Тип": "Рідкі", "Аром ат": "Ін ший", 2025- УКТЗЕ Д": "33 07 49 00 00"}	12- 23Т00:2 6:30.817 Z
3059	BL4.8x 300	590532 551520 6	Paloma	Хомути пластик ові чорні 4,8x300 мм, 100шт	Опис товару	212	UAH	113	130	20	1	TRUE	TRUE	www	www	{"Довж ина, мм": "3 00", "Ш ирина профіл ю шини, мм": "4, 8", "УК ТЗЕД": "3925 90 80 00"}	2025- 12- 23Т00:2 6:30.817 Z

3.2.3. Формування уніфікованої моделі товару

Кінцевим результатом роботи модуля автоматизованої обробки є формування уніфікованої моделі товару, яка використовується для інтеграції даних у CRM та ERP-системах підприємства. Уніфікована модель товару являє собою структурований набір атрибутів, узгоджений між усіма інформаційними системами.

До складу уніфікованої моделі входять:

- унікальний ідентифікатор товару;
- стандартизована назва та опис;
- класифікаційні ознаки (категорія, підкатегорія);
- нормалізовані атрибути та фільтри для пошуку;
- цінові параметри та дані про наявність;
- службові поля для інтеграції та аудиту змін.

Формування такої моделі забезпечує єдине представлення товарної інформації незалежно від джерела походження даних та каналу продажу. Уніфікована модель товару слугує основою для подальшої синхронізації даних між CRM, ERP, вебсайтом та роздрібними системами, а також створює передумови для аналітичної обробки та масштабування системи.

Таблиця 3.3

Уніфікована модель товару

Поле	Тип	Обов'язков	Джерело	Приклад
external_id	string	так	XML offer/@id	"2658"
vendor_code	string	так	XML vendorCode	"79013"
name_ua	string	так	XML name_ua	"Ароматизатор"
brand	string	ні	XML vendor	"Paloma"
price	number	так	XML price	71.00

old_price	number/null	ні	XML old_price	91.00
currency	string	так	XML currencyId	"UAH"
stock_qty	int	ні	XML stock_quantit y	20
images	array[string]	ні	XML picture	["url1"]
attributes	object(map)	ні	XML param	{"Тип":"Рідкі"}

Формування уніфікованої моделі товару реалізовано у вигляді функції `buildUnifiedProductModel()`, що приймає XML-вузол `<offer>` і повертає структурований об'єкт, який відповідає полям, наведеним у Таблиці 3.3. Основні етапи включають вилучення полів, нормалізацію значень та формування структури атрибутів і службових метаданих.»

```
function buildUnifiedProductModel(offerEl) {
    // 1) Ідентифікаційні поля
    const external_id = getAttr_(offerEl, "id");           // offer/@id
    const vendor_code = getText_(offerEl, "vendorCode"); // <vendorCode>

    // 2) Базові текстові поля
    const name_ua = cleanText_(getText_(offerEl, "name_ua")); // <name_ua>
    const brand = cleanText_(getText_(offerEl, "vendor")); // <vendor>

    // 3) Цінові параметри
    const price = toNumber_(getText_(offerEl, "price")); // <price>
    const old_price = toNumberOrNull_(getText_(offerEl, "old_price")); //
<old_price>
    const currency = normalizeCurrency_(getText_(offerEl, "currencyId")); //
<currencyId>

    // 4) Наявність / залишки
    const stock_qty = toInt_(getText_(offerEl, "stock_quantity")); //
<stock_quantity>

    // 5) Зображення (може бути 0..n)
    const images = extractPictures_(offerEl); // <picture>...</picture>

    // 6) Атрибути з довільних параметрів <param name="...">...</param>
```

```

const attributes = normalizeAttributes_(extractParams_(offerEl));

// 7) Службові поля для інтеграції та аудиту
const meta = {
  source: "www_XML",
  processed_at: new Date().toISOString()
};

// 8) Формування уніфікованої моделі (відповідає Таблиці 3.2)
return {
  external_id: external_id,
  vendor_code: vendor_code,
  name_ua: name_ua,
  brand: brand || null,
  price: price,
  old_price: (old_price !== null && old_price > price) ? old_price : null,
  currency: currency,
  stock_qty: stock_qty,
  images: images,
  attributes: attributes,
  meta: meta
};
}

/* ----- Допоміжні функції ----- */

function getAttr_(el, attrName) {
  return el.getAttribute(attrName) ?
String(el.getAttribute(attrName).getValue()).trim() : "";
}

function getText_(parent, tagName) {
  const el = parent.getChild(tagName);
  if (!el) return "";
  return String(el.getText() || "").trim();
}

function cleanText_(text) {
  return String(text || "")
    .replace(/<[^\>]*>/g, " ")
    .replace(/\s+/g, " ")
    .trim();
}

function toNumber_(v) {
  const n = parseFloat(String(v || "").replace(",", "."));

```

```

    return isNaN(n) ? 0 : n;
}

```

```

function toNumberOrNull_(v) {
    const s = String(v || "").trim();
    if (!s) return null;
    const n = parseFloat(s.replace(",", "."));
    return isNaN(n) ? null : n;
}

```

```

function toInt_(v) {
    const n = parseInt(String(v || "").replace(",", "."), 10);
    return isNaN(n) ? 0 : n;
}

```

```

function normalizeCurrency_(cur) {
    const c = String(cur || "").trim().toUpperCase();
    const allowed = { "UAH": true, "USD": true, "EUR": true };
    return allowed[c] ? c : "UAH";
}

```

```

function extractPictures_(offerEl) {
    const pics = offerEl.getChildren("picture") || [];
    return pics
        .map(p => String(p.getText() || "").trim())
        .filter(u => u && /^https?:\/\//i.test(u));
}

```

```

function extractParams_(offerEl) {
    const params = offerEl.getChildren("param") || [];
    const map = {};
    params.forEach(p => {
        const key = p.getAttribute("name") ?
String(p.getAttribute("name").getValue()).trim() : "";
        const val = cleanText_(String(p.getText() || ""));
        if (key) map[key] = val;
    });
    return map;
}

```

```

function normalizeAttributes_(attrs) {
    const keyMap = {
        "Тип": "Тип",
        "Аромат": "Аромат",

```

```

    "УКТЗЕД": "УКТЗЕД",
    "Код УКТЗЕД": "УКТЗЕД"
  };

  const out = {};
  Object.keys(attrs || {}).forEach(k => {
    const mappedKey = keyMap[k] || k;
    out[mappedKey] = normalizeAttrValue_(mappedKey, attrs[k]);
  });

  return out;
}

function normalizeAttrValue_(key, value) {
  const v = cleanText_(value);

  if (key === "Тип") {
    const low = v.toLowerCase();
    if (low.includes("рідк")) return "Рідкі";
    if (low.includes("гел")) return "Гелеві";
  }
  return v;
}

```

Приклад виклику функції для одного товару:

```

function testBuildUnifiedModel() {
  const xmlText = UrlFetchApp.fetch(
    "https://opt.www.ua/data/ff764b8ef412d7090041a00ff8e0b187.xml"
  ).getContentText();

  const xml = XmlService.parse(xmlText);
  const root = xml.getRootElement();
  const offers = root
    .getChild("shop")
    .getChild("offers")
    .getChildren("offer");

  // БЕРЕМО ОДИН offer
  const unified = buildUnifiedProductModel(offers[0]);

  Logger.log(JSON.stringify(unified, null, 2));
}

```

Отримуємо уніфікований товар в форматі JSON:

```
{ "external_id": "2658",
  "vendor_code": "79013",
  "name_ua": "Ароматизатор Paloma Parfume Tabac рідкий",
  "brand": "Paloma",
  "price": 71,
  "old_price": null,
  "currency": "UAH",
  "stock_qty": 20,
  "images": [
    "https://opt.www.ua/uploads/cache/CatalogItems/CatalogItems2658/0859f63b69-1_resize_1200x1200.jpeg"
  ],
  "attributes": {
    "Тип": "Рідкі",
    "Аромат": "Інший",
    "УКТЗЕД": "3307 49 00 00"
  },
  "meta": {
    "source": "www_XML",
    "processed_at": "2025-12-23T00:50:48.906Z"
  }
}
```

3.3. Автоматизоване формування товарних характеристик із використанням OpenAI API

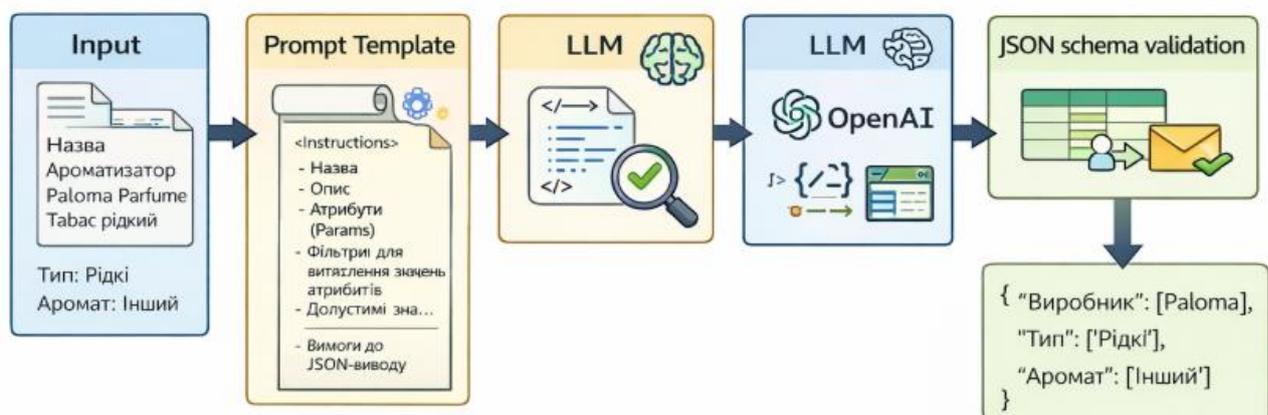


Рис. 3.5. Процес формування товарних характеристик

На рисунку зображено процес формування уніфікованої моделі товару, який є завершальним етапом автоматизованої обробки товарних даних перед їх

інтеграцією у CRM та ERP-системи підприємства. Даний процес базується на поетапному перетворенні нормалізованих даних у структурований набір атрибутів із чітко визначеною семантикою та форматом.

Вхідними даними для процесу є нормалізована товарна інформація, отримана з XML-прайсу постачальника після етапів очищення, стандартизації та уніфікації. Ці дані включають ідентифікатори товару, текстові описи, цінові параметри, інформацію про наявність та довільні атрибути, передані у вигляді параметрів.

На наступному етапі здійснюється формування ключових ідентифікаційних полів уніфікованої моделі, зокрема зовнішнього ідентифікатора товару та артикулу постачальника. Дані поля забезпечують однозначну ідентифікацію товару між різними інформаційними системами та джерелами даних.

Далі виконується формування основних інформаційних атрибутів товару, таких як стандартизована назва, бренд, ціна, валюта та кількість на складі. Одночасно відбувається перевірка коректності значень і застосування базових бізнес-правил, наприклад, перевірка співвідношення поточної та старої ціни або усунення від'ємних значень залишків.

Окремим блоком формується набір нормалізованих атрибутів і фільтрів для пошуку, які представлені у вигляді структурованого набору ключ–значення. Це дозволяє забезпечити гнучке використання товарних характеристик у CRM-системі, та на вебсайті інтернет-магазину.

Завершальним етапом є додавання службових полів, які містять інформацію про джерело даних та час обробки. Такі поля використовуються для інтеграції, аудиту змін та контролю актуальності інформації.

Результатом виконання всіх етапів є уніфікована модель товару — цілісна, структурована та узгоджена форма представлення товарної інформації, незалежна від формату та джерела початкових даних. Запропонований підхід забезпечує єдине представлення товарів у всіх інформаційних системах підприємства та створює основу для подальшої автоматизованої синхронізації, аналітики й масштабування системи.

3.3.1. Постановка задачі класифікації товарних атрибутів

Задача автоматизованого формування товарних характеристик полягає у вилученні та класифікації атрибутів товарів на основі неструктурованих і слабоструктурованих даних, таких як назви, текстові описи та технічні характеристики. Метою є отримання структурованих значень атрибутів, які можуть бути безпосередньо використані в CRM-системі та на вебсайті інтернет-магазину.

У межах даної роботи задача формалізується як задача багатокласової класифікації та інформаційного вилучення, де для кожного товару необхідно визначити значення наперед заданого набору характеристик (наприклад, виробник, матеріал, колір, тип підключення, наявність термостатичного картриджа тощо). Особливістю задачі є те, що значення атрибутів можуть бути подані в довільному текстовому вигляді, а також відсутні або неявно зазначені в описі товару.

Традиційні rule-based підходи виявляються недостатньо ефективними для розв'язання цієї задачі через високу варіативність текстових формулювань. Тому у роботі застосовано методи машинного навчання на основі великих мовних моделей для семантичного аналізу товарної інформації.

3.3.2. Логіка взаємодії з мовною моделлю

Інтелектуальна обробка товарних даних у межах розробленої системи реалізується шляхом взаємодії з великою мовною моделлю через API-доступ до сервісу OpenAI. Мовна модель використовується як інструмент семантичного аналізу, що дозволяє автоматизовано визначати та класифікувати товарні характеристики на основі наявних текстових і структурованих даних.

Логіка взаємодії з мовною моделлю побудована таким чином, щоб максимально обмежити простір можливих відповідей і забезпечити стабільність результатів. Для цього формується структурований запит, який містить чітко

визначені складові: рольову інструкцію, вхідні товарні дані, перелік атрибутів для заповнення, допустимі значення та вимоги до формату відповіді.

Рольова інструкція визначає контекст задачі та закріплює за мовною моделлю функцію експерта з товарних характеристик. Вхідні дані включають повний набір інформації про товар, збережений у внутрішній базі підприємства, зокрема назви, описи, характеристики, бренд, категорію та допоміжні атрибути. На основі цих даних модель повинна сформувати значення визначених фільтрів.

Особливу увагу приділено використанню переліків допустимих значень для кожного атрибуту. Такий підхід дозволяє зменшити кількість помилок класифікації, уніфікувати значення характеристик та забезпечити їхню сумісність із довідниками CRM-системи. Окремою вимогою є обов'язкове повернення результату у форматі коректного JSON без додаткових коментарів або пояснень.

Приклад логіки формування запиту до мовної моделі наведено нижче. У даному прикладі для товару типу “Змішувач” автоматизовано визначаються значення фільтрів на основі наявних характеристик та опису

Приклад запиту до мовної моделі:

Ти експерт з товарних характеристик.

Заповни характеристики для вказаного товару.

Є товар «Змішувач GRANADO Mataro alto Inox». Осць усі доступні дані про товар: [структуровані дані з бази товарів].

Заповни такі фільтри: Виробник, Тип підключення, Матеріал, Колір, Термостатичний картридж.

Для кожного фільтра:

- спочатку обері значення зі списку допустимих;*
- якщо відповідного значення у списку немає, але воно явно зазначене в описі товару — використай його;*
- якщо інформація відсутня — пропусти фільтр;*
- не додавай нові фільтри та коментарі;*
- повертай відповідь виключно у форматі коректного JSON.*

Такий підхід дозволяє використовувати мовну модель не як генеративний інструмент довільних відповідей, а як керований механізм автоматизованої класифікації товарних характеристик. Для підвищення надійності отримані результати додатково проходять етап валідації на відповідність бізнес-правилам і довідникам підприємства.

3.3.3. Формування структурованого JSON-виводу

Результатом роботи мовної моделі є структурований вивід у форматі JSON, який містить лише ті товарні характеристики, для яких у вхідних даних виявлено релевантну інформацію. Використання формату JSON забезпечує простоту подальшої автоматизованої обробки, валідації та інтеграції результатів у CRM та інші інформаційні системи підприємства.

Кожен атрибут у JSON-виводі представлений у вигляді ключа, що відповідає назві фільтра, та масиву значень, що дозволяє у майбутньому розширювати модель без зміни базової структури. Строго визначений формат виводу дозволяє уникнути неоднозначності інтерпретації результатів і спрощує перевірку відповідності отриманих значень допустимим довідникам.

Приклад JSON-виводу для наведеного товару подано нижче:

```
{ "Виробник": [ "GRANADO" ], "Тип підключення": [ "Врізний" ], "Матеріал": [ "Нержавіюча сталь SUS 304" ], "Колір": [ "Нержавіюча сталь" ], "Термостатичний картридж": [ "Так" ] }
```

Отриманий JSON проходить автоматичну перевірку на коректність структури, наявність лише дозволених атрибутів і допустимих значень. Після успішної валідації дані використовуються для автоматичного оновлення товарних характеристик у CRM-системі, а також можуть бути передані на вебсайт інтернет-магазину для покращення навігації та пошуку товарів.

Запропонований формат JSON є універсальним і масштабованим, що дозволяє без суттєвих змін у логіці інтеграції додавати нові характеристики або адаптувати систему до інших товарних категорій. Таким чином, структурований JSON-вивід виступає ключовою ланкою між інтелектуальною обробкою

товарних даних та практичним використанням результатів у бізнес-процесах підприємства.

3.3.4. Інтеграція результатів у CRM

Отримані структуровані дані інтегруються в CRM-систему KeyCRM за допомогою API-інтерфейсу. Інтеграційний модуль забезпечує зіставлення атрибутів, сформованих мовною моделлю, з відповідними полями товарної картки у CRM.

Інтеграція результатів класифікації, отриманих від LLM у форматі JSON, виконується через API KeyCRM шляхом відображення атрибутів на відповідні `custom_fields`. Для масштабованості використано двофазний підхід: формування черги завдань (оновлення/створення) та їх пакетне виконання. Це дозволяє зменшити кількість API-запитів, забезпечити відновлення процесу після зупинки та реалізувати контроль якості через валідацію структури й значень перед записом у CRM.

Зараз все реалізовано через Google Sheets для тестування та відлагодження. Після цього відбудеться інтеграція для виконання на сервері.

```

/***** CONFIG *****/
const KEYCRM = {
  TOKEN: 'YOUR_KEYCRM_TOKEN',
  BASE: 'https://openapi.keycrm.app/v1/products',

  // custom field "Посилання на сайт"
  CUSTOM_LINK: { id: 60, uuid: 'СТ_1060' },

  /**
   * Мапінг “Фільтр (LLM)” -> “custom field KeyCRM”
   */
  FILTER_FIELD_MAP: {
    "Виробник": { id: 101, uuid: "СТ_1101" },
    "Тип підключення": { id: 102, uuid: "СТ_1102" },
    "Матеріал": { id: 103, uuid: "СТ_1103" },
    "Колір": { id: 104, uuid: "СТ_1104" },
    "Термостатичний картридж": { id: 105, uuid: "СТ_1105" }
  }
}

```

```

};

const SHEET_NEW    = 'Товари для CRM';
const SHEET_CRM    = 'Товари з CRM';
const SHEET_TASKS  = 'Дані для оновлення';
const SHEET_CREATE = 'Дані для створення';

const BATCH_SIZE = 200;
const SLEEP_MS   = 400;
const NAME_MAX   = 240;

/***** HELPERS *****/
function _headers() {
  return { Authorization: `Bearer ${KEYCRM.TOKEN}`, 'Content-Type':
'application/json' };
}
function _sleep(ms) { Utilities.sleep(ms); }
function _normName(s) { return String(s || "").replace(/\\s+/g, ' ').trim(); }
function _toNum(v) {
  const s = String(v || "").trim();
  if (!s) return null;
  const n = parseFloat(s.replace(',', '.'));
  return isNaN(n) ? null : n;
}
function _jsonSafeParse(s, fallback) {
  try { return JSON.parse(s); } catch(e) { return fallback; }
}
function _makeUniqueName(baseName, sku, existingNames) {
  let base = _normName(baseName);
  if (!existingNames.has(base)) { existingNames.add(base); return base; }

  const suffix = `[${sku}]`;
  let candidate = base;
  if (candidate.length + suffix.length > NAME_MAX) candidate =
candidate.slice(0, NAME_MAX - suffix.length);
  candidate += suffix;

  if (!existingNames.has(candidate)) { existingNames.add(candidate); return
candidate; }

  for (let i = 2; i <= 99; i++) {
    let c = candidate; const add = `-${i}`;
    if (c.length + add.length > NAME_MAX) c = c.slice(0, NAME_MAX -
add.length);
    c += add;
    if (!existingNames.has(c)) { existingNames.add(c); return c; }
  }
}

```

```

    }
    const ts = `${Date.now()}`;
    let c = candidate;
    if (c.length + ts.length > NAME_MAX) c = c.slice(0, NAME_MAX -
ts.length);
    c += ts; existingNames.add(c); return c;
  }

/**
 * Формує custom_fields для KeyCRM на основі:
 * - посилання (Link)
 * - JSON фільтрів/характеристик (Filters JSON), який дав LLM
 */
function buildCustomFields_(link, filtersJsonText) {
  const custom = [];

  // Link -> custom field
  const linkVal = String(link || "").trim();
  if (linkVal) {
    custom.push({ id: KEYCRM.CUSTOM_LINK.id, uuid:
KEYCRM.CUSTOM_LINK.uuid, value: linkVal });
  }

  // Filters JSON -> mapped custom fields
  const filters = _jsonSafeParse(filtersJsonText || "{}", {});
  if (filters && typeof filters === "object") {
    Object.keys(filters).forEach(filterName => {
      const map = KEYCRM.FILTER_FIELD_MAP[filterName];
      if (!map) return; // якщо в KeyCRM не заведений цей фільтр —
пропускаємо

      let val = filters[filterName];

      if (Array.isArray(val)) val = val.filter(Boolean).join("; ");
      if (val === null || val === undefined) return;

      const valueText = String(val).trim();
      if (!valueText) return;

      custom.push({ id: map.id, uuid: map.uuid, value: valueText });
    });
  }

  return custom.length ? custom : null;
}

```

```

/***** STEP 1: BUILD QUEUE *****/
function prepareKeyCRMSyncQueue() {
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  const sheetNew = ss.getSheetByName(SHEET_NEW);
  const sheetCRM = ss.getSheetByName(SHEET_CRM);

  if (!sheetNew || !sheetCRM) throw new Error('Не знайдено потрібних
аркушів.');
```

```

  let sheetTasks = ss.getSheetByName(SHEET_TASKS);
  if (!sheetTasks) sheetTasks = ss.insertSheet(SHEET_TASKS);
  sheetTasks.clear();

  let sheetCreate = ss.getSheetByName(SHEET_CREATE);
  if (!sheetCreate) sheetCreate = ss.insertSheet(SHEET_CREATE);
  sheetCreate.clear();

  const newRows = sheetNew.getDataRange().getValues();
  const crmRows = sheetCRM.getDataRange().getValues();
  if (newRows.length < 2) throw new Error('Аркуш "Товари для CRM"
порожній.');
```

```

  if (crmRows.length < 2) Logger.log('Увага: "Товари з CRM" порожній.
Буде тільки create_batch.');
```

```

  // indexes new
  const hNew = newRows[0];
  const idxNew = {
    sku: hNew.indexOf('ID'),
    title: hNew.indexOf('Title'),
    link: hNew.indexOf('Link'),
    image: hNew.indexOf('Image'),
    price: hNew.indexOf('Price'),
    sale: hNew.indexOf('Sale Price'),
    filters: hNew.indexOf('Filters JSON') // ВАЖЛИВО: колонка з LLM JSON
  };
  Object.keys(idxNew).forEach(k => { if (idxNew[k] === -1 && k !== "sale")
Logger.log(⚠ Missing column: ${k}`); });

  // indexes crm
  const hCRM = crmRows[0];
  const idxCRM = {
    id: hCRM.indexOf('ID'),
    sku: hCRM.indexOf('SKU'),
    price: hCRM.indexOf('Price'),
    custom: hCRM.indexOf('Custom Fields')
  };

```

```

// map existing CRM products by SKU
const crmMap = {};
const existingNames = new Set();

crmRows.slice(1).forEach(r => {
  const sku = String(r[idxCRM.sku] || "").trim();
  if (!sku) return;

  const id = r[idxCRM.id];
  const price = _toNum(r[idxCRM.price]);

  // дістаємо link з custom_fields (якщо є)
  let linkValue = "";
  const cf = _jsonSafeParse(r[idxCRM.custom] || '[]', []);
  if (Array.isArray(cf)) {
    const linkField = cf.find(f => f.id === KEYCRM.CUSTOM_LINK.id ||
f.uuid === KEYCRM.CUSTOM_LINK.uuid);
    if (linkField) linkValue = String(linkField.value || "").trim();
  }

  crmMap[sku] = { id, price, link: linkValue };
});

// sheet create header
const createHeader = ['Name','SKU','Price','Image','Link','Filters JSON'];
sheetCreate.getRange(1, 1, 1,
createHeader.length).setValues([createHeader]);

// tasks header
const tasksHeader = ['Action','KeyCRM_ID','SKU','Payload','Status'];
const tasks = [tasksHeader];

const createRows = [];
const updates = [];

newRows.slice(1).forEach(r => {
  const sku = String(r[idxNew.sku] || "").trim();
  if (!sku) return;

  const title = _normName(r[idxNew.title]);
  const link = String(r[idxNew.link] || "").trim();
  const image = String(r[idxNew.image] || "").trim();
  const filtersJson = String(r[idxNew.filters] || "").trim();

  // price priority: Sale Price > Price

```

```

let p = String(r[idxNew.sale] || "").trim();
if (!p) p = String(r[idxNew.price] || "").trim();
const price = _toNum(p);

const crmData = crmMap[sku];

// UPDATE if exists in CRM
if (crmData) {
  const payload = {};
  let need = false;

  // price
  if (price !== null && price !== crmData.price) {
    payload.price = price;
    payload.currency_code = 'UAH';
    need = true;
  }

  // custom fields (link + LLM filters)
  const cf = buildCustomFields_(link, filtersJson);
  if (cf) {
    payload.custom_fields = cf;
    need = true;
  }

  if (need) {
    updates.push(['update', crmData.id, sku, JSON.stringify(payload), ""]);
  }
  return;
}

// CREATE if not exists
const finalName = _makeUniqueName(title || sku, sku, existingNames);
createRows.push([finalName, sku, price || 0, image, link, filtersJson]);
});

// write create rows
if (createRows.length) {
  sheetCreate.getRange(2, 1, createRows.length,
createHeader.length).setValues(createRows);
}

// build create_batch tasks with refs to sheetCreate ranges
for (let i = 0; i < createRows.length; i += 100) {
  const start = 2 + i;
  const count = Math.min(100, createRows.length - i);

```

```

    tasks.push(['create_batch', '', `rows ${start}-${start + count - 1}`,
JSON.stringify({ ref: { start, count } }), '']);
    }

    // add updates
    updates.forEach(u => tasks.push(u));

    // write tasks
    sheetTasks.getRange(1, 1, tasks.length, tasks[0].length).setValues(tasks);

    Logger.log(` 📄 Tasks: ${tasks.length - 1},
create_batch=${Math.ceil(createRows.length/100)}, update=${updates.length}`);
    }

    /***** STEP 2: EXECUTE QUEUE *****/
    function continueKeyCRMSyncQueue() {
        const ss = SpreadsheetApp.getActiveSpreadsheet();
        const sheetTasks = ss.getSheetByName(SHEET_TASKS);
        const sheetCreate = ss.getSheetByName(SHEET_CREATE);

        if (!sheetTasks) throw new Error(` Не найдено аркуш
"${SHEET_TASKS}"`);
        const rows = sheetTasks.getDataRange().getValues();
        if (rows.length < 2) { Logger.log('Немає завдань. '); return; }

        const h = rows[0];
        const col = name => h.indexOf(name) + 1;

        const cAction = col('Action');
        const cID = col('KeyCRM_ID');
        const cSKU = col('SKU');
        const cPayload = col('Payload');
        const cStatus = col('Status');

        let processed = 0;

        for (let r = 2; r <= rows.length; r++) {
            if (processed >= BATCH_SIZE) break;

            const status = sheetTasks.getRange(r, cStatus).getValue();
            if (status === 'DONE') continue;

            const action = sheetTasks.getRange(r, cAction).getValue();
            const id = sheetTasks.getRange(r, cID).getValue();
            const skuRef = sheetTasks.getRange(r, cSKU).getValue();
            const payloadText = sheetTasks.getRange(r, cPayload).getValue();

```

```

try {
  let resp, code, body;

  if (action === 'update') {
    resp = UrlFetchApp.fetch(`${KEYCRM.BASE}/${id}`, {
      method: 'put',
      contentType: 'application/json',
      headers: _headers(),
      payload: payloadText,
      muteHttpExceptions: true
    });

    code = resp.getResponseCode();
    body = resp.getContentText();
    if (code >= 200 && code < 300) {
      sheetTasks.getRange(r, cStatus).setValue('DONE');
    } else {
      sheetTasks.getRange(r, cStatus).setValue(`ERR ${code}`);
      Logger.log(body);
    }
  }

  if (action === 'create_batch') {
    if (!sheetCreate) throw new Error('Missing sheet "Дані для створення"');

    const meta = _jsonSafeParse(payloadText || "{}", {});
    const ref = meta.ref;
    if (!ref) throw new Error('Invalid ref payload');

    const rowsData = sheetCreate.getRange(ref.start, 1, ref.count,
6).getValues();
    const products = rowsData.map(row => {
      const [name, sku, price, image, link, filtersJson] = row;

      const p = {
        name: _normName(name || sku || ""),
        sku: String(sku || "").trim(),
        price: Number(price) || 0,
        currency_code: 'UAH'
      };

      if (image) p.pictures = [String(image).trim()];

      const cf = buildCustomFields_(link, filtersJson);
      if (cf) p.custom_fields = cf;

```

```

    return p;
  }).filter(p => p.sku);

  if (!products.length) {
    sheetTasks.getRange(r, cStatus).setValue('DONE (empty)');
    processed++;
    continue;
  }

  resp = UrlFetchApp.fetch(`${KEYCRM.BASE}/import`, {
    method: 'post',
    contentType: 'application/json',
    headers: _headers(),
    payload: JSON.stringify({ products }),
    muteHttpExceptions: true
  });

  code = resp.getResponseCode();
  body = resp.getContentText();

  if (code >= 200 && code < 300) {
    sheetTasks.getRange(r, cStatus).setValue(`DONE
    (${products.length})`);
  } else {
    sheetTasks.getRange(r, cStatus).setValue(`ERR ${code}`);
    Logger.log(body);
  }
}

} catch (e) {
  sheetTasks.getRange(r, cStatus).setValue(`ERR ${e.message}`);
  Logger.log(` ❌ ${action} ${id || skuRef || ""} → ${e.message}`);
}

processed++;
_sleep(SLEEP_MS);
}

Logger.log(` ✅ Обработано: ${processed}`);
}

```

3.4. Валідація та контроль якості результатів машинного навчання

Результати автоматизованого формування товарних атрибутів із використанням великої мовної моделі мають ймовірнісний характер, тому потребують системного контролю якості. У межах тестування та впровадження було оброблено та створено 6967 товарних позицій, для яких мовна модель сформувала структуровані характеристики. В результаті перевірки встановлено, що середня похибка в заповненні характеристик становить близько 5%.

Основні типи помилок, виявлені під час аналізу:

- неправильне визначення значення атрибуту (наприклад, колір/матеріал через неоднозначні формулювання у характеристиках);
- часткове вилучення параметрів (частина характеристик не потрапляє у JSON через відсутність явного згадування);
- надлишкова генерація (модель пропонує атрибут, який логічно можливий, але не підтверджений вхідними даними);
- помилки зіставлення з довідниками (значення коректне семантично, але відсутнє у списку допустимих варіантів, в цьому випадку створювався дубль характеристики).

Аналіз помилок здійснювався шляхом порівняння результатів автоматичного заповнення з еталонними значеннями, сформованими за допомогою вибіркової ручної перевірки. Основний акцент робився на атрибутах, що напряму впливають на коректність фільтрації та відображення товару в каталозі (виробник, матеріал, тип підключення, колір тощо). Отримані результати використовуються для вдосконалення вхідних даних (очищення/нормалізація), уточнення шаблону промпта та підсилення правил валідації.

3.5. Оцінка ефективності впровадженої системи

3.5.1. Порівняння процесів до та після автоматизації

До впровадження системи основні операції з формування характеристик виконувались вручну: оператор аналізував опис, заповнював характеристики та узгоджував значення з довідниками. Такий процес був повільним, залежним від людського фактору та погано масштабувався при збільшенні кількості постачальників і товарних груп.

Після впровадження автоматизованого модуля формування характеристик із використанням OpenAI API більшість операцій перейшла в автоматичний режим: система отримує дані з внутрішньої бази товарів, формує запит до мовної моделі, отримує структурований JSON та виконує інтеграцію в KeyCRM. Ручне втручання зведено до контролю винятків та вибіркової перевірки. Це забезпечило суттєве прискорення наповнення каталогу та зменшення залежності від людського фактору.

3.5.2. Зменшення витрат часу та кількості помилок

Ефективність впровадження оцінювалася за показниками продуктивності та якості. На підставі практичних результатів встановлено, що витрати часу на додавання нових товарів зменшилися в 10–15 разів (залежно від товарної групи та повноти вхідних даних). Ключовим фактором є автоматизоване формування характеристик, яке раніше вимагало значного часу на аналіз описів і ручне заповнення полів.

Також відзначено контрольований рівень якості: похибка в характеристиках становить близько 5%, при цьому помилки виявляються і локалізуються завдяки бізнес-правилам та вибіркового контролю. У порівнянні з ручною обробкою це дозволяє знизити частку системних помилок (наприклад, через неуважність або різне трактування атрибутів різними операторами) та забезпечити єдині стандарти заповнення.

Окремо оцінено витрати на використання OpenAI API. За період обробки було виконано 6 967 запитів і використано 17 624 051 токен, а загальна вартість склала \$9.63, що є економічно доцільним з огляду на досягнуте зростання продуктивності та скорочення ручних витрат. Враховуючи, що до впровадження оператор отримував за кожен товар 0,23\$, вигода складає 170%

3.5.3. Практичні результати впровадження

Впровадження системи інтеграції товарних даних із використанням OpenAI API дало вимірюваний бізнес-ефект. Якщо раніше оператор фізично міг додати до 500 товарів на місяць, то після автоматизації підприємство отримало можливість додавати і 10 000 товарів на місяць без пропорційного збільшення штату або витрат часу на рутинні операції.

Серед практичних результатів:

- різке зростання пропускної здатності процесу наповнення каталогу;
- скорочення часу виходу товарів у продаж (time-to-market);
- підвищення повноти та стандартизованості характеристик;
- покращення пошуку та фільтрації на сайті через уніфіковані атрибути;
- створення бази для масштабування: підключення нових постачальників і розширення асортименту без “вузького місця” у вигляді ручної обробки.

Таким чином, система продемонструвала ефективність як з технічної, так і з економічної точки зору, підтвердивши доцільність подальшого розвитку ML-компонентів у процесах управління товарними даними.

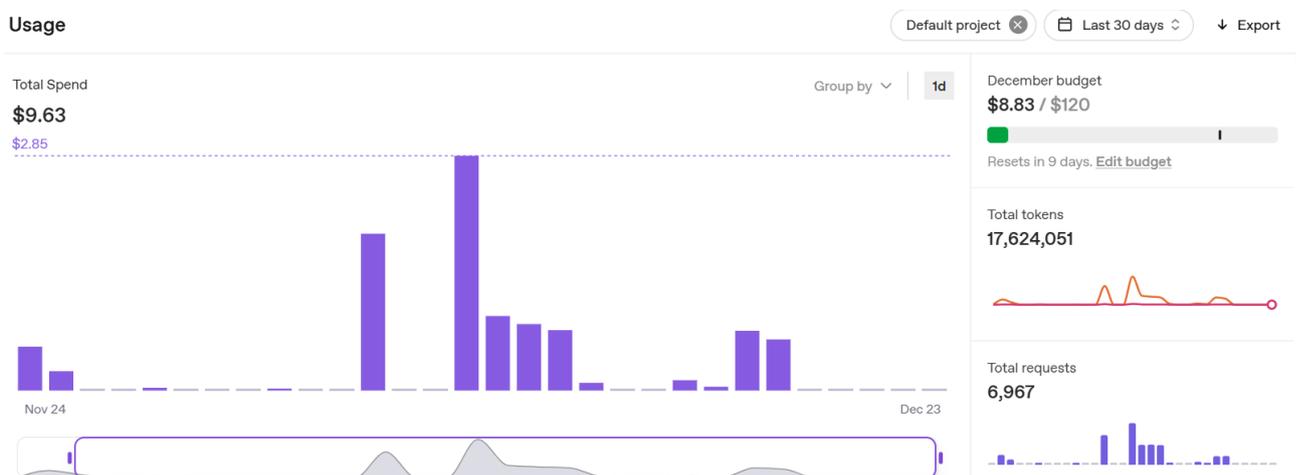


Рис. 3.6 Результати впровадження

ВИСНОВОК ДО РОЗДІЛУ 3

У третьому розділі розроблено та реалізовано систему інтеграції товарних даних у CRM та ERP-системах торговельного підприємства із застосуванням методів машинного навчання з API-доступом до великих мовних моделей. Запропоноване рішення ґрунтується на модульній архітектурі, що забезпечує поетапну обробку товарної інформації — від збору та нормалізації даних постачальників до формування уніфікованої моделі товару й автоматизованої інтеграції результатів у CRM-систему KeyCRM.

У межах розділу реалізовано модуль автоматизованої підготовки товарних даних, який виконує очищення, стандартизацію та уніфікацію атрибутів незалежно від формату вхідної інформації. На основі нормалізованих даних формується уніфікована модель товару, що забезпечує єдине представлення товарної інформації в CRM, вебсайті та суміжних інформаційних системах підприємства.

Особливу увагу приділено автоматизованому формуванню товарних характеристик і фільтрів із використанням API-доступу до великих мовних моделей. Запропонована логіка взаємодії з мовною моделлю, що включає рольові інструкції, обмеження допустимих значень і формування структурованого JSON-виводу, дозволила досягти стабільних і відтворюваних результатів класифікації товарних атрибутів. Практична апробація системи підтвердила прийнятний рівень якості — середня похибка у визначенні характеристик становить близько 5%, що компенсується застосуванням бізнес-правил і процедур валідації.

Для забезпечення надійності та контрольованості результатів машинного навчання в системі реалізовано механізми аналізу похибок, перевірки результатів за допомогою бізнес-правил і контролю життєвого циклу даних. Це дозволило інтегрувати ML-компонент у корпоративне середовище без пом'якшення вимог до якості товарної інформації та забезпечити можливість масштабування рішення.

Оцінка ефективності впровадженої системи показала суттєве зростання продуктивності процесу наповнення товарного каталогу. Якщо до автоматизації оператор міг обробляти не більше 500 товарів на місяць, то після впровадження

системи підприємство отримало можливість додавати 10 000 і більше товарів на місяць. Час обробки однієї товарної позиції зменшився у 10–15 разів, а ручне втручання обмежується лише контролем виняткових випадків.

Важливим результатом є економічний ефект від автоматизації. До впровадження системи вартість додавання одного товару складала 10 грн у вигляді оплати праці оператора. Після автоматизації основні витрати пов'язані з використанням OpenAI API: за обробку 6 967 товарів загальна вартість склала \$9.63, що є значно нижчим за витрати на ручну обробку аналогічного обсягу даних. Це підтверджує економічну доцільність запропонованого підходу та його ефективність у реальних умовах експлуатації.

Отже, результати, отримані у третьому розділі, підтверджують, що використання методів машинного навчання та великих мовних моделей у задачах інтеграції товарних даних дозволяє не лише автоматизувати рутинні процеси, а й істотно підвищити масштабованість, якість та економічну ефективність управління товарною інформацією на торговельних підприємствах.

Таблиця 3.4

Порівняння показників процесу наповнення товарного каталогу до та після впровадження системи

Показник	До впровадження	Після впровадження
Спосіб формування характеристик	Ручне заповнення оператором	Автоматизоване формування з використанням ML та OpenAI API
Середній час обробки одного товару	Високий (ручний аналіз опису та характеристик)	Зменшено у 10–15 разів
Максимальна кількість товарів на місяць	До 500 товарів	10 000+ товарів

Масштабованість	Низька, обмежена людським ресурсом	Висока, обмежена лише обчислювальним і ресурсами
Середній рівень помилок	Залежить від оператора, нестабільний	≈5%, контрольований і прогнозований
Контроль якості	Вибірковий, вручну	Автоматизований (бізнес-правила + валідація)
Узгодженість атрибутів	Нерівномірна	Стандартизована
Витрати на оплату праці	10 грн за 1 товар	Мінімальні (контроль і винятки)
Вартість обробки 1 товару	≈ 10 грн	≈ 0,04 грн (OpenAI API)
Загальні витрати за 6 967 товарів	≈ 69 670 грн	≈ \$9.63 (~380 грн)
Інтеграція з CRM	Часткова, з ручними правками	Повністю автоматизована (API KeyCRM)

ВИСНОВКИ

У магістерській роботі розроблено та впроваджено систему інтеграції товарних даних у CRM та ERP-системах торговельного підприємства із застосуванням методів машинного навчання та API-доступу до великих мовних моделей. Запропоноване рішення забезпечує автоматизований повний цикл обробки товарної інформації — від отримання даних постачальників у різних форматах до формування уніфікованої моделі товару та інтеграції результатів у CRM-систему KeyCRM.

У ході практичної апробації система була застосована для автоматизованого формування характеристик 6 967 товарних позицій, при цьому середній рівень похибок у визначенні атрибутів склав близько 5%, що є прийнятним показником для задач семантичної класифікації. Впроваджені механізми бізнес-правил і валідації дозволили забезпечити контрольованість результатів та зменшити вплив помилок машинного навчання на кінцеві дані.

Одним із ключових результатів роботи є суттєве підвищення продуктивності процесу наповнення товарного каталогу. Якщо до автоматизації оператор міг обробити не більше 500 товарів на місяць, то після впровадження системи підприємство отримало можливість додавати 10 000 і більше товарів на місяць без пропорційного збільшення витрат часу або персоналу. Час обробки одного товару зменшився у 10–15 разів залежно від товарної групи та повноти вхідних даних.

Важливим аспектом є економічна ефективність запропонованого підходу. До впровадження системи оплата праці оператора складала 10 грн за кожен доданий товар, що при масштабуванні призводило до значних постійних витрат. Після автоматизації витрати на формування характеристик зведено переважно до вартості використання OpenAI API. За обробку 6 967 товарів загальні витрати склали \$9.63, що еквівалентно приблизно 400 грн, тоді як ручна обробка аналогічного обсягу коштувала б близько 69 670 грн. Таким чином, вартість обробки одного товару зменшилася більш ніж у 180 разів.

Отримані результати підтверджують, що використання великих мовних моделей у задачах інтеграції та нормалізації товарних даних є не лише технічно

доцільним, але й економічно обґрунтованим. Запропонована система забезпечує високу масштабованість, зменшення залежності від ручної праці та створює надійну основу для подальшого розвитку інформаційної інфраструктури торговельного підприємства, зокрема підключення нових постачальників і розширення асортименту без суттєвого зростання витрат.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. — Cambridge : MIT Press, 2016. — 775 p.
2. Mitchell T. *Machine Learning*. — New York : McGraw-Hill, 1997. — 432 p.
3. Bishop C. M. *Pattern Recognition and Machine Learning*. — New York : Springer, 2006. — 738 p.
4. Hastie T., Tibshirani R., Friedman J. *The Elements of Statistical Learning*. — New York : Springer, 2009. — 745 p.
5. Quinlan J. R. *Induction of Decision Trees // Machine Learning*. — 1986. — Vol. 1. — P. 81–106.
6. Breiman L. *Random Forests // Machine Learning*. — 2001. — Vol. 45. — P. 5–32.
7. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. — 4th ed. — Pearson, 2021. — 1136 p.
8. Ковальчук В. В. *Методи та засоби штучного інтелекту : навч. посіб.* — Київ : КПІ ім. Ігоря Сікорського, 2020. — 256 с.
9. Бахрушин В. Є. *Машинне навчання та аналіз даних : навч. посіб.* — Запоріжжя : ЗНУ, 2019. — 312 с.
10. Литвин В. В., Пасічник В. В. *Інтелектуальні системи : підручник*. — Львів : Новий Світ–2000, 2018. — 406 с.
11. Семенюк О. М. *Системи підтримки прийняття рішень : навч. посіб.* — Київ : КНЕУ, 2017. — 284 с.
12. ДСТУ ISO/IEC 2382:2015. *Інформаційні технології. Словник термінів*. — Київ : ДП «УкрНДНЦ», 2015.
13. OpenAI. *OpenAI API Documentation [Електронний ресурс]*. — Режим доступу: <https://platform.openai.com/docs> (дата звернення: 20.11.2025).
14. Google. *Google Developers. Machine Learning Guides [Електронний ресурс]*. — Режим доступу: <https://developers.google.com/machine-learning> (дата звернення: 18.11.2025).

15.Scikit-learn. Scikit-learn Documentation [Электронный ресурс]. — Режим доступа: <https://scikit-learn.org/stable/> (дата звернения: 15.11.2025).

16.TensorFlow. TensorFlow Documentation [Электронный ресурс]. — Режим доступа: <https://www.tensorflow.org> (дата звернения: 15.11.2025).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

**«СИСТЕМА ІНТЕГРАЦІЇ ТОВАРНИХ ДАНИХ У CRM
ТА ERP-СИСТЕМАХ ІЗ ЗАСТОСУВАННЯМ МЕТОДІВ
МАШИННОГО НАВЧАННЯ»**

Виконав: студент групи САДМ-61 Микола МАНДРА

Науковий керівник: Доктор технічних наук, професор СТОРЧАК КАМІЛА ПАВЛІВНА

**Актуальність теми****Зростання обсягів товарних даних**

Експоненційне збільшення асортименту та кількості характеристик в сучасній електронній комерції.

Високі витрати ручної обробки

Значні часові та фінансові ресурси, необхідні для формування та підтримки актуальності каталогів вручну.

Ризик помилок та неузгодженості

Висока ймовірність людських помилок при формуванні характеристик, що погіршує якість даних та впливає на клієнтський досвід.

Необхідність автоматизації з ІІ

Критична потреба у впровадженні інтелектуальних систем на базі машинного навчання для автоматизації процесів.

Мета та Завдання Дослідження

Глобальна Мета

Дослідження та розробка системи інтеграції товарних даних між CRM та ERP-системами з використанням методів машинного навчання (ML) для підвищення якості, повноти та узгодженості даних у торговому підприємстві.

- 1 **Аналіз сучасних підходів**
Проаналізувати існуючі рішення для управління та інтеграції товарних даних у CRM та ERP-системах.
- 2 **Дослідження проблем обробки**
Вивчити типові проблеми, пов'язані з різними джерелами та форматами вхідних товарних даних.
- 3 **Аналіз методів ML**
Визначити та проаналізувати методи машинного навчання, придатні для класифікації, нормалізації та зіставлення товарних даних.
- 4 **Проектування архітектури**
Сформулювати архітектуру системи інтеграції товарних даних для багатосистемного середовища.
- 5 **Реалізація прототипу**
Розробити та впровадити функціональний прототип системи на основі обраних методів ML.
- 6 **Оцінка ефективності**
Оцінити результативність запропонованого рішення та надати практичні рекомендації щодо його масштабування.



Об'єкт та Предмет Дослідження



1

Об'єкт Дослідження

Торговельне підприємство, яке функціонує у сфері оптової та роздрібно торгівлі та використовує багатоканальну (omnichannel) модель продажів.

2

Ключові Канали

- Інтернет-магазин (e-commerce)
- Роздрібні торгові точки (offline retail)
- B2B-канали (оптові продажі)

Це зумовлює наявність великого та складного обсягу товарної інформації, що потребує інтеграції.

Методи, моделі та програмні засоби інтеграції, очищення, нормалізації та зіставлення товарних даних у CRM, ERP та суміжних інформаційних системах із застосуванням методів машинного навчання.



Аналіз Існуючих Підходів до Управління Товарними Даними

Традиційні Методи:

Ручне Введення Товарів

Створення та редагування карток товарів операторами вручну. Надзвичайно повільний процес, що вимагає високої концентрації уваги.

Шаблонні Правила та Фільтри

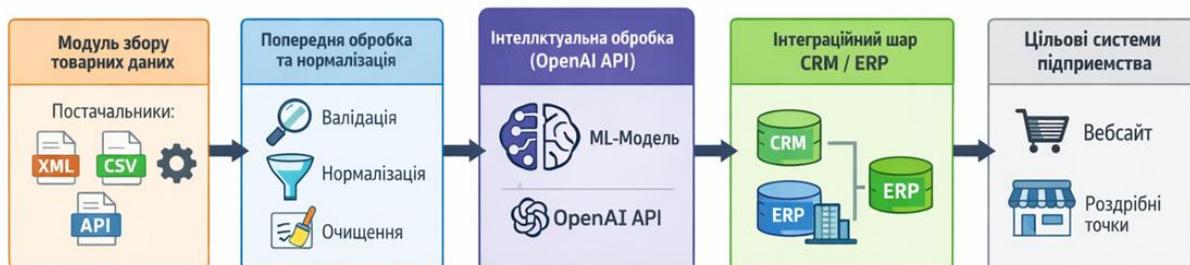
Використання простих регулярних виразів або фіксованих правил для вилучення характеристик. Неefективно для неструктурованих даних.

Ключові Недоліки:

- 1 **Обмежена Масштабованість**
Зі зростанням асортименту пропорційно збільшуються часові та кадрові витрати.
- 2 **Високий Рівень Похибок**
Суб'єктивність та людський фактор призводять до неточностей у характеристиках і дублюваннях даних.
- 3 **Висока Вартість**
Значні операційні витрати на утримання великого штату контент-менеджерів.



Архітектура Розробленої Системи Інтеграції



Практична Реалізація та Функціонал



Система дозволила автоматизувати критично важливі та трудомісткі процеси управління товарними даними, мінімізуючи людське втручання.

Автоматизоване Створення Товарів

Миттєве формування нових карток товарів у системі на основі вхідних файлів постачальників.

Інтелектуальна Генерація Характеристик

Точне вилучення та нормалізація технічних параметрів, необхідних для повноцінної картки товару.

Масова Обробка Даних

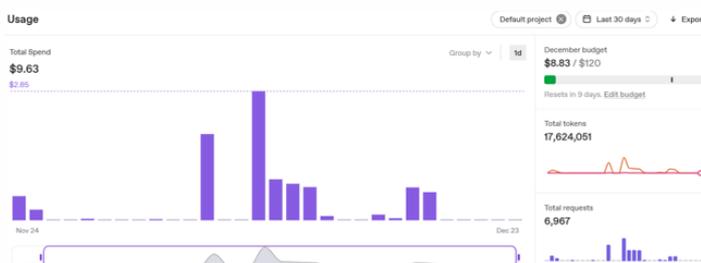
Здатність обробляти великі обсяги даних (тисячі позицій) за лічені хвилини без участі оператора.

- Прототип системи забезпечує високу швидкість відгуку та дозволяє інтегрувати товарні дані з різних джерел у єдиний узгоджений формат.



Результати Експериментальної Перевірки

Експериментальне впровадження системи продемонструвало значний ріст продуктивності та точності обробки даних.



6967

Створених Товарів

Кількість товарних позицій, автоматично інтегрованих у систему за час тестування.

5%

Середня Похибка

Мінімальний рівень неточностей при генерації характеристик, що значно нижче, ніж при ручному введенні.

10

Скорочення Часу

Час на обробку однієї позиції скорочено у 10–15 разів у порівнянні з ручною працею.

Продуктивність

До впровадження: до 500 товарів/місяць. Після впровадження: 10 000+ товарів/місяць.



Порівняльний Аналіз: "До" та "Після" Впровадження ML-Системи

Систематичне порівняння демонструє ключові переваги автоматизованого підходу над традиційним ручним методом.

Час обробки	Високий, залежний від обсягу даних та досвіду оператора.	Мінімальний, автоматизований процес, не залежить від обсягу.
Людський фактор (помилки)	Значний ризик неточностей, неузгодженості та дублювання.	Мінімізований, точність підтримується ML-моделями.
Масштабованість	Низька, вимагає постійного найму персоналу при зростанні каталогу.	Висока, система легко справляється зі зростанням обсягів даних.
Вартість	≈10 грн за обробку одного товару (прямі операційні витрати).	Значно нижча завдяки автоматизації та оптимізації процесів.



Економічний Ефект та Перспективи

1. Скорочення Операційних Витрат

Значне скорочення витрат на оплату праці контент-менеджерів та операторів завдяки автоматизації рутинних завдань.

2. Зменшення Кількості Помилки

Підвищення якості та надійності даних, що позитивно впливає на показники конверсії та зменшує кількість повернень.

3. Швидке Масштабування

Можливість оперативно виводити на ринок нові товарні категорії та розширювати асортимент без збільшення штату.

4. Підвищення Ефективності Бізнес-Процесів

Оптимізація часу виведення товару на ринок (Time-to-Market) та підвищення загальної швидкості бізнес-операцій.

Впровадження системи інтеграції товарних даних на базі машинного навчання є стратегічною інвестицією, що забезпечує конкурентну перевагу на ринку e-commerce.



Висновки та Досягнення

Розроблена система інтеграції товарних даних продемонструвала значні переваги та підтвердила свою ефективність у практичному застосуванні.

Повний Цикл Автоматизації

Система забезпечує автоматизовану обробку товарної інформації: від збору даних постачальників до інтеграції у CRM та ERP-системи.

Висока Точність Класифікації

Середній рівень похибок у визначенні атрибутів склав близько **5%**, що є прийнятним для семантичної класифікації.

Підвищення Продуктивності

Збільшення обробки товарів з 500 до **10 000+** на місяць, скорочення часу обробки одного товару у **10–15 разів**.

Економічне Обґрунтування та Масштабованість

Впровадження системи забезпечило значну економію, що робить її не тільки технічно доцільною, але й економічно вигідною інвестицією.

- **Значне Зменшення Витрат**
- **Низькі Операційні Витрати**
- **Висока Масштабованість**

Використання великих мовних моделей у задачах інтеграції даних є стратегічним кроком до оптимізації бізнес-процесів та забезпечення конкурентної переваги на ринку.

