МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА

Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Системи штучного інтелекту»

галузь знань	12 Інформаційні технології
спеціальність	122 Комп'ютерні науки

Форма навчання: очна (денна) *або* заочна

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему: «Проєктування інтелектуальної системи для підтримки процесів дропшипінгу на платформі Amazon»

здобувача Павленка Артема Віталійовича
(ПІБ, підпис)

Науковий керівник: к.т.н., доцент Галузинський Георгій Петрович
(науковий ступінь, учене звання, ПІБ)

_____
(підпис)

Робота допущена до захисту перед екзаменаційною комісією з атестації здобувачів вищої освіти (ЕК)

Завідувач кафедри: к.е.н., доцент Тішков Б.О. _____
(підпис)

Київ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Системи штучного інтелекту»

галузь знань 12 Інформаційні технології
спеціальність 122 Комп'ютерні науки

ПОГОДЖЕНО:
Керівник проєктної групи (гарант)
освітньо-професійної програми

_____Васильєва Л.В.
«____» _____202_ р.

ЗАТВЕРДЖУЮ:
Завідувач кафедри інформаційних
систем в економіці

_____ Тішков Б.О.
«____» _____ 202_ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Павленка Артема Віталійовича
(прізвище, ім'я, по батькові)
очної(денної) форми навчання
очної (денної), заочної, дистанційної

на підготовку кваліфікаційної магістерської роботи
на тему: *«Проєктування інтелектуальної системи для підтримки процесів дропшипінгу на платформі Amazon»*

Тему затверджено наказом ректора Університету від «07» березня 2025 р. № 464-ст

Кваліфікаційна магістерська робота виконується на матеріалах написаних тез на тему «Проєктування інтелектуальної системи для підтримки процесів дропшипінгу на платформі Amazon»

План кваліфікаційної магістерської роботи

Розділ І ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ СШІ
(назва розділу)
Розділ ІІ ХАРАКТЕРИСТИКА СШІ ТА ПОСТАНОВКА ЗАДАЧІ
(назва розділу)
Розділ ІІІ РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ
(назва розділу)
Об'єкт дослідження: процеси збору, обробки та аналізу інформації про продавців на платформах електронної комерції, зокрема eBay.
Предмет дослідження: методи та програмні засоби автоматизованого збору даних з веб-ресурсів і класифікації продавців на основі ознак за допомогою алгоритмів машинного навчання.

Мета кваліфікаційної магістерської роботи: розробити інтелектуальну систему збору та класифікації даних про продавців на платформі eBay з використанням методів машинного навчання, яка дозволить автоматизувати процес оцінки надійності постачальників для потреб електронної комерції, зокрема дропшипінгу

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I здобувач має проаналізувати сучасні підходи до використання методів штучного інтелекту у сфері електронної комерції. Заплановано дослідження архітектур систем, що забезпечують автоматизовану обробку онлайн-даних, зокрема на маркетплейсах. Буде охарактеризовано наявні алгоритми класифікації (Random Forest, SVM, дерева рішень тощо) з позиції їхньої застосовності до класифікації продавців. Особливу увагу приділено методам збору інформації з веб-ресурсів (веб-скрапінг) та обробки слабко структурованих даних. За результатами аналізу буде сформульовано вимоги до функціоналу майбутньої системи.

У розділі II передбачено розроблення логічної структури інтелектуальної системи, яка включатиме модулі парсингу, обробки та класифікації даних. Заплановано обґрунтувати вибір технологій та інструментів (Python, бібліотеки pandas, selenium, scikit-learn) для реалізації кожного з етапів. У цьому розділі здобувач має здійснити проєктування системи збору даних із платформи eBay за допомогою Selenium, а також визначити підхід до побудови ознак на основі зібраної інформації. Буде запропоновано загальну схему взаємодії між модулями та описано сценарії використання системи у вигляді алгоритмічного представлення.

У розділі III здобувач має реалізувати повну програмну частину проєкту у середовищі Jupyter Notebook, використовуючи інтерактивні блоки для кожного етапу. Заплановано імплементацію модулів збору даних із сайту eBay, обробки таблиць, підготовки ознак та класифікації продавців за допомогою навченої моделі Random Forest. Особливу увагу буде приділено перевірці коректності зібраної інформації, тестуванню моделі на різних прикладах і збереженню результатів у зручному форматі CSV. Під час експериментального етапу здобувач проаналізує точність класифікації, стабільність моделі при зміні даних, а також сформулює рекомендації щодо її подальшого вдосконалення та можливості масштабування.

Завдання підготував
науковий керівник _____ Галузинський Г.П.
(підпис) *(ініціали, прізвище)*

«____» _____ 2025 р.

Завдання одержав
здобувач _____ Павленко А.В.
(підпис) *(ініціали, прізвище)*

«____» _____ 2025 р.

Реферат

Кваліфікаційна бакалаврська робота містить 98 сторінок, 6 таблиць, 27 рисунків, список використаних джерел із 39 найменувань, додатки.

«Проєктування інтелектуальної системи збору та класифікації даних про продавців eBay на основі машинного навчання»

Об'єктом дослідження кваліфікаційної роботи є процеси аналізу інформації про онлайн-продавців на платформі електронної комерції eBay.

Предметом дослідження є сукупність теоретичних, методичних і практичних підходів до створення програмного засобу для автоматизованого збору, обробки та класифікації даних про продавців за допомогою методів машинного навчання.

Мета і завдання дослідження. Основною метою кваліфікаційної роботи є розробка інтелектуальної системи, що дозволяє автоматизувати процес збору даних з eBay, проводити обробку та класифікацію продавців як надійних або потенційно ризикованих.

Відповідно до поставленої мети визначені такі *завдання*:

− дослідити предметну область та провести аналіз підходів до автоматизації оцінки продавців;

− обґрунтувати вибір інструментів для реалізації системи;

− реалізувати систему збору даних за допомогою бібліотеки Selenium;

− описати структуру та логіку обробки зібраних даних;

− реалізувати модель класифікації на основі алгоритму Random Forest;

− реалізувати інтерфейс у середовищі Jupyter Notebook;

− описати програмне, технічне, інформаційне й організаційне забезпечення проєкту;

− провести тестування системи та аналіз результатів.

Теоретична, методична та практична значущість отриманих результатів. У межах роботи проаналізовано сучасні підходи до оцінки продавців в електронній комерції, зокрема в моделі дропшипінгу. Реалізовано універсальну систему, яка здатна самостійно збирати інформацію про продавців з eBay, проводити попередню обробку даних, перетворювати їх у формат, придатний для машинного навчання, та виконувати класифікацію. Практична цінність полягає в можливості використання системи для відбору надійних постачальників у дропшипінгу, автоматизації процесу прийняття рішень та зниження ризиків закупівель.

Рік виконання кваліфікаційної магістерської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: eBay, дропшипінг, класифікація, рейтинг, машинне навчання, Random Forest, інтелектуальна система.

Summary

The bachelor's qualification thesis consists of 98 pages, 6 tables, 27 figures, a list of 39 references, and appendices.

«Design of an Intelligent System for Data Collection and Classification of eBay Sellers Based on Machine Learning»

The object of the study is the process of analyzing information about online sellers on the eBay e-commerce platform.

The subject of the research is the set of theoretical, methodological, and practical approaches to developing a software solution for automated collection, processing, and classification of seller data using machine learning methods.

*Purpose and tasks of the study.*The main goal of the thesis is to develop an intelligent system that automates the process of data collection from eBay, processes it, and classifies sellers as either reliable or potentially risky.

The following tasks were defined to achieve this goal:

- study the domain and analyze approaches to seller evaluation automation;
- justify the choice of tools for system implementation;
- implement a data collection system using the Selenium library;
- describe the structure and logic of collected data processing;
- implement a classification model based on the Random Forest algorithm;
- build a user interface using Jupyter Notebook;
- describe the software, technical, informational, and organizational components of the project;
- conduct system testing and analyze the results

Theoretical, methodological, and practical significance of the results. The work analyzes modern approaches to seller evaluation in e-commerce, especially within the dropshipping model. A universal system was implemented that autonomously collects seller data from eBay, processes it, transforms it into a machine learning-compatible format, and performs classification. The practical value lies in the ability to use the system for reliable supplier selection in dropshipping, automate decision-making, and reduce procurement risks.

Year of completion of the bachelor's qualification thesis – 2025.

Year of defense – 2025.

Keywords: eBay, dropshipping, classification, rating, machine learning, Random Forest, intelligent system.

В І Д Г У К

на кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту «Інститут інформаційних технологій в економіці»
освітньо-професійної програми «Інформаційні управляючі системи і технології»

Павленка Артема Віталійовича
на тему «Проєктування інтелектуальної системи для підтримки процесів дропшипінгу на платформі Amazon»

1. *Актуальність теми*: Розвиток електронної комерції зумовлює стрімке зростання популярності дропшипінгу як бізнес-моделі, що дозволяє здійснювати торгівлю без необхідності зберігання товару або управління логістикою. Підготовлена здобувачем кваліфікаційна магістерська робота є комплексним дослідженням направленим на автоматизацію робіт в цій сфері.

2. *Позитивні риси кваліфікаційної магістерської роботи*: Здобувач дослідив проблемні питання, що виникають у дропшінг-постачальників та в достатній мірі висвітив проектні рішення, пов'язані з розробкою інтелектуальної системи для підтримці процесів дропшипінгу на платформі Amazon.

3. *Наявність самостійних розробок автора*: За результатами проведеного комплексу заходів, що знаходяться в площині теоретичних, аналітичних і конструктивних досліджень, виконаних здобувачем під час підготовки кваліфікаційної магістерської роботи, були розроблені основні програмні модулі інтелектуальної системи для підтримці процесів дропшипінгу на платформі Amazon.

4. *Цінність теоретичних висновків та практичних рекомендацій*: Висновки та рекомендації можуть бути застосовані в практичній діяльності підприємств при впровадженні сучасних інформаційних технологій в сфері електронної комерції.

5. *Наявність недоліків*: Недоліки відсутні. Система, створена в ході дослідження, може бути використана як основа для подальшого розвитку систем для підтримці процесів дропшипінгу.

6. *Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК*: Кваліфікаційна магістерська робота здобувача Ярцева С.Д. відповідає встановленим вимогам, рекомендується до захисту і заслуговує відмінної оцінки.

Науковий керівник доц., к.т.н. Галузинський Г.П.

Рецензія

на кваліфікаційну магістерську роботу
здобувача вищої освіти
Павленко Артема Віталійовича

(прізвище, ім'я, по батькові)

Тема «Проєктування інтелектуальної системи для підтримки процесів дропшипінгу на платформі Amazon»

Актуальність теми кваліфікаційної магістерської роботи і доцільність її розроблення Тема є надзвичайно актуальною з огляду на стрімкий розвиток електронної комерції та дропшипінгу, де якість постачальника безпосередньо впливає на репутацію і прибутковість бізнесу. Розробка інтелектуальної системи, що дозволяє автоматизувати процес збору та оцінювання надійності продавців на платформі eBay, є доцільною і своєчасною,

Якість проведеного дослідження Дослідження виконано на високому рівні. Автор здійснив всебічний аналіз предметної області, обґрунтував вибір алгоритмів машинного навчання та реалізував працездатну систему у вигляді інтерактивного програмного модуля. Робота містить чітку структуру, логічно викладені теоретичні та практичні результати.

Позитивні риси кваліфікаційної магістерської роботи Серед сильних сторін роботи варто відзначити, що здобувач самостійно реалізував повне програмне рішення, яке охоплює етапи від збору даних до класифікації. Робота відзначається ефективним використанням мови програмування Python та відповідних бібліотек, зокрема Selenium, scikit-learn та pandas. Архітектура системи є обґрунтованою та логічно побудованою. Проєкт добре документований у вигляді Jupyter Notebook, що полегшує розуміння логіки виконання.

Зауваження Немає

Практична значимість висновків і рекомендацій Розроблена система має реальну практичну цінність: вона може бути використана у дропшипінгу, маркетинговому аналізі, закупівельній аналітиці, а також адаптована до інших торгових платформ. Запропонований підхід сприяє автоматизації рутинних процесів та підвищенню якості прийняття рішень на основі даних.

Місце роботи та посада рецензента

Науковий ступінь, учене звання (*за наявності*) _____

(підпис, ПІБ)

Підпис засвідчую: _____

(посада, підпис)

ЗМІСТ

ВСТУП 4
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ СШІ 5
1.1 Дослідження предметної області. Збір інформації та вивчення матеріалів з теми кваліфікаційної магістерської роботи. 5
1.2 Аналіз існуючих СШІ й інтелектуальних систем предметної області 6
1.3 Постановка проблеми та формування задач. 7
1.4 Обґрунтування вибору підходів і технологій для проектування та створення СШІ 7
Висновки до розділу 1 7
РОЗДІЛ 2 ХАРАКТЕРИСТИКА СШІ ТА ПОСТАНОВКА ЗАДАЧІ 9
2.1 Характеристика об’єкта дослідження 9
2.2 Структура і характеристика СШІ та її компонентів. 9
2.3 Методи, моделі і моделювання процесів і елементів складних систем 9
2.3.1 Методи дослідження й синтезу компонент систем штучного інтелекту 9
2.3.2 Моделі та методи оптимізації в системах штучного інтелекту 10
2.3.3 Методи та моделі управління в системах штучного інтелекту 10
Висновки до розділу 2 10
РОЗДІЛ 3 РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ 11
3.1 Моделювання та проектування бази знань прийняття інтелектуальних рішень і управління 11
3.2 Розроблення користувацького інтерфейсу. Елементи та структура. 11
3.3 Проектування забезпечувальних підсистем СШІ. Реалізація системи. 11
3.3.1 Інформаційне забезпечення. 11
3.3.2 Програмне забезпечення. 12
3.3.3 Технічне забезпечення 12
3.3.4 Організаційно- економічне забезпечення 12
Висновки до розділу 3 12
ВИСНОВКИ 13
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 13

ВСТУП

Актуальність проблеми.

Сучасні тенденції розвитку електронної комерції вимагають постійного вдосконалення процесів аналітики, автоматизації та прийняття рішень. Одним із найперспективніших напрямів у цій сфері є дропшипінг — бізнес-модель, яка дозволяє реалізовувати товари без необхідності фізичного зберігання чи логістики. Попри переваги, дропшипінг пов'язаний із рядом викликів: вибір прибуткових товарів, перевірка постачальників, моніторинг цін і попиту. Ручне виконання цих процесів є малоефективним та трудомістким.

Саме тому актуальною є розробка інтелектуальних систем, які дозволяють автоматизувати аналіз товарів і формувати обґрунтовані рекомендації щодо вибору для подальшого перепродажу. Це особливо важливо у контексті роботи з глобальними платформами, такими як Amazon та eBay. Проєктування такої системи сприяє підвищенню точності бізнес-рішень, зменшенню ризиків і оптимізації часу.

Аналіз останніх досліджень і публікацій

Проблематика дропшипінгу досліджується в працях українських та закордонних науковців, зокрема у контексті маркетингових стратегій, цифрової логістики та оптимізації асортименту. Інтелектуальні системи рекомендацій, методи скорингу, rule-based системи, а також базові підходи до аналізу великих обсягів даних розглядаються в працях таких авторів, як G. Box, S. Hochreiter, A. Vaswani та інших. Водночас прикладна реалізація простих, гнучких прототипів, орієнтованих на малий і середній e-commerce-бізнес, потребує подальшого практичного дослідження.

Мета дослідження.

- Провести аналіз предметної області та дослідити практичні аспекти дропшипінгу;
- Визначити вимоги до функціональності інтелектуальної системи;

- Розробити скорингову модель оцінки товарів;
- Реалізувати структуру системи у вигляді прототипу на основі Google Sheets;
- Провести тестування системи на реальних товарних даних;
- Оцінити ефективність використання системи в умовах електронної торгівлі.

Об’єкт дослідження

Об’єктом дослідження є процеси дропшипінгу в електронній комерції, а саме діяльність з продажу товарів на платформі Amazon без власного складу, із залученням сторонніх постачальників для виконання замовлень. До об’єкта входять бізнес-процеси, що супроводжують модель дропшипінгу: пошук та відбір товарів для перепродажу, взаємодія з постачальниками, розміщення товарних позицій на маркетплейсі, управління цінами, виконання і доставлення замовлень, а також забезпечення відповідності політикам платформи і очікуванням клієнтів. Таким чином, об’єкт охоплює всю сукупність операцій та рішень, що приймаються дропшипером у ході ведення торговельної діяльності на Amazon.

Предмет дослідження

Предметом дослідження є методи та інформаційно-технологічні засоби підтримки прийняття рішень у процесах дропшипінгу на Amazon, зокрема методи інтелектуального аналізу і скорингового оцінювання товарів, доступних на зовнішніх торговельних майданчиках (на прикладі eBay), для визначення їх доцільності та привабливості для продажу на Amazon. Іншими словами, предметом виступає розроблений підхід до проєктування інтелектуальної системи, що включає: модель представлення даних про потенційні товари (атрибути товарів, показники постачальників, ринкові метрики тощо); алгоритми обробки та агрегування цих даних (скоринг, фільтрація, ранжування); програмну реалізацію даного підходу у вигляді прототипного інструменту на основі хмарних сервісів. Предметом дослідження охоплено як теоретичні аспекти (методологія побудови системи, вибір і налаштування алгоритмів), так і прикладні (особливості інтеграції з платформами Amazon та eBay, реалізація у середовищі Google Sheets, інтерфейс користувача прототипу).

Методи дослідження

У процесі виконання дослідження використовувалися методи:

- інтелектуального аналізу даних (Data Mining),
- скоринг,
- структурно-логічного моделювання,
- табличного аналізу,
- евристичного класифікування,
- візуалізації та інтерпретації результатів.

Теоретична, методична та практична значущість отриманих результатів.

Теоретичне значення полягає у розвитку підходів до використання інтелектуальних систем у сфері дропшипінгу. Методична цінність — у побудові моделі прийняття рішень на основі скорингу. Практичне значення — у можливості застосування прототипу системи в реальному електронному бізнесі для зниження ризиків і підвищення прибутковості.

Інформаційна база дослідження

Інформаційну базу склали публічні дані з eBay та Amazon, аналітичні платформи (Keepa, Grabley), наукові праці, статті, документи стандартів, власні аналітичні таблиці та дані, зібрані під час проходження практики.

Структура роботи. Загальний обсяг роботи становить ____ сторінок друкованого тексту, 26 рисунків на 14 сторінках, ___ таблиць на ___ сторінках, 1 додаток на 2 сторінках. Список використаних джерел налічує 52 найменування.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ СШІ

1.1 Дослідження предметної області. Збір інформації та вивчення матеріалів з теми кваліфікаційної магістерської роботи.

Сутність дропшипінгу та його роль в електронній комерції. Дропшипінг (пряма поставка) – це модель електронної комерції, у якій онлайн-продавець виступає посередником між покупцем та постачальником. Продавець не тримає товарних запасів на власному складі, а приймає замовлення від клієнтів і передає інформацію про замовлення безпосередньо постачальнику, який здійснює відправлення товару покупцю напряму.[1]

Весь процес виглядає так: клієнт замовляє товар у інтернет-магазині посередника, посередник (дропшиппер) отримує оплату і лише після цього закуповує товар у постачальника за оптовою ціною, а постачальник відправляє товар кінцевому покупцю. Прибуток посередника формується за рахунок різниці між роздрібною ціною продажу і оптовою ціною постачальника.[2]

Таким чином, дропшипінг дозволяє підприємцю започаткувати онлайн-бізнес з мінімальним стартовим капіталом, без витрат на складські приміщення та логістику.[1] За рахунок передачі складського зберігання і виконання замовлень постачальнику, дропшиппер може зосередитися на маркетингу та пошуку клієнтів.

Ця модель сьогодні відіграє значну роль в електронній комерції: за оцінками, близько 27% інтернет-рітейлерів обрали дропшипінг як основний метод виконання замовлень[3], а світовий ринок дропшипінгу у 2024 році досяг обсягу ~$344 млрд і прогнозовано зростатиме на ~23% щорічно до 2029 року.[4] Отже, дропшипінг став популярним шляхом швидкого старту онлайн-продажів, оскільки знижує бар’єри входу в бізнес і дозволяє масштабувати асортимент без інвестицій у запас товарів.

Особливості дропшипінгу на платформі Amazon. Платформа Amazon є одним з найбільших майданчиків, де застосовується модель дропшипінгу. Водночас Amazon встановив суворі правила для продавців-дропшипперів, щоб підтримувати високу якість обслуговування покупців.[5] Згідно з офіційною політикою Amazon, продавець, який працює за моделлю дропшипінгу, повинен залишатися «продавцем-рекордсменом» для товару: він має бути зазначений як продавець на всіх рахунках, пакувальних листах та накладних, а будь-які згадки сторонніх постачальників заборонені.[5] Amazon категорично забороняє відправлення замовлень прямо від іншого рітейлера, якщо на упаковці або вкладених документах видно назву чи логотип сторонньої компанії. Наприклад, ситуація, коли клієнт Amazon замовив товар, а отримав посилку з логотипом Walmart, розцінюється як порушення, що вводить споживача в оману.[6] У 2021–2022 роках Amazon ще більше посилив контроль: були масові блокування акаунтів продавців, які порушували політику дропшипінгу, оскільки така практика без відповідного контролю може негативно впливати на якість товарів, ефективність доставки та користувацький досвід на платформі

Попри ці обмеження, Amazon дозволяє дропшипінг за умови дотримання правил. Продавець повинен забезпечити, щоб покупець отримав товар без жодних ознак участі стороннього постачальника. Для цього багато продавців використовують так звані prep-центри (центри підготовки відправлень). Prep-центр – це сторонній сервіс, який отримує товар від постачальника від імені продавця, перевіряє та перепаковує його згідно вимог Amazon, видаляючи будь-які вкладення чи маркування від стороннього продавця, і відправляє посилку кінцевому клієнту від імені продавця.[7]

Така схема дозволяє формально виконати всі умови Amazon: товар надходить покупцеві з належним пакуванням і документами від продавця маркетплейсу, а не від постачальника. Схема процесу дропшипінгу на Amazon з преп-центром виглядає наступним чином: Amazon (платформа) → оформлення замовлення покупцем → Постачальник → Преп-центр → Клієнт.

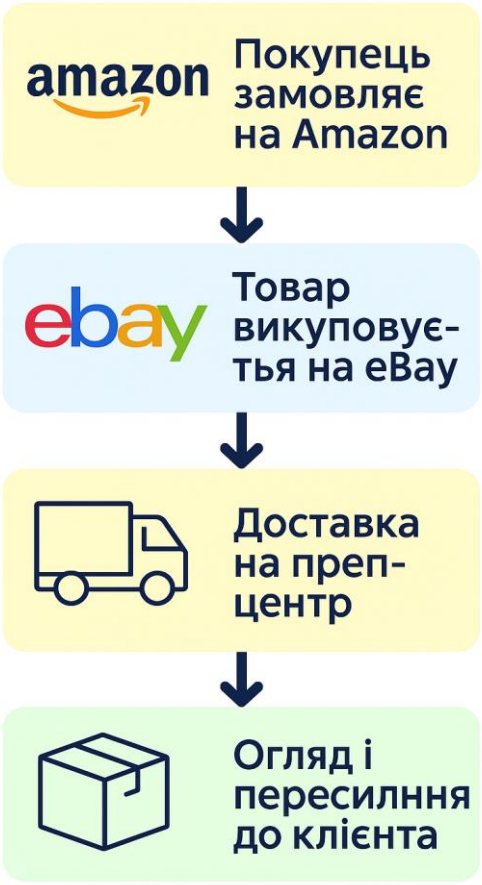


Рисунок 1.1 – Базовий процес дропшипінгу Amazon–eBay з використанням преп-центру

Джерело: розроблено автором

У цій схемі Amazon виступає як платформа для замовлення, постачальник виконує роль відправника товару, преп-центр забезпечує відповідність відправлення вимогам Amazon, а клієнт отримує замовлення безпосередньо. Таким чином, дропшипер на Amazon фактично передає постачальнику і 3PL-препцентру всі операції з товаром (зберігання, упаковка, доставка), зосереджуючись на виборі товарів, виставленні лістингів та обслуговуванні замовлень.

Основна увага – пошук надійних постачальників. У моделі дропшипінгу критично важливо налагодити співпрацю з надійними постачальниками, адже від них залежить своєчасність доставки і якість товарів. За результатами галузевих опитувань, 84% онлайн-продавців вважають, що встановлення стабільних відносин з постачальником є вирішальним фактором успіху в дропшипінгу.

Проблема пошуку постачальників полягає в тому, що підприємець фактично довіряє сторонній компанії виконання значної частини своїх зобов'язань перед покупцем. Типові ризики при виборі ненадійного постачальника включають затримки або зриви відправлень, відправку товарів неналежної якості, помилки в комплектації замовлень, відсутність актуальної інформації про наявність товарів на складі, а також недотримання особливих вимог (наприклад, вкладення рахунків чи рекламних матеріалів, що заборонено Amazon). Крім того, існує ризик потрапити на недоброчесного постачальника, який може використати контактні дані ваших клієнтів у власних цілях.

Рисунок 1.2 – Типові проблеми у ручному дропшипінгу без автоматизації

Джерело: розроблено автором

Усі ці проблеми несуть негативні наслідки для продавця-дропшиппера: погіршення рейтингу магазину через негативні відгуки та скарги, фінансові втрати внаслідок повернень і компенсацій, а в контексті Amazon – ще й ризик блокування аккаунту за незадовільні показники виконання замовлень.[8] Тому процес пошуку та відбору постачальників вимагає особливої ретельності.

Критерії надійності постачальника включають: дотримання обумовлених термінів відправлення, стабільна наявність товарного запасу, висока якість продукції (підтверджена сертифікатами або відгуками), готовність працювати за моделлю дропшипінгу (без власної реклами для кінцевого клієнта), прозорі умови оплати та повернення, гарна комунікація. Процес оцінки та відбору постачальників зазвичай виглядає так: підприємець формує список потенційних постачальників (за допомогою пошуку в інтернеті, на B2B платформах, маркетплейсах або через рекомендації). Далі – встановлює контакт з кожним кандидатом, уточнює ключові умови співпраці: доступний асортимент, оптові ціни, мінімальний обсяг замовлення, умови та вартість доставки, способи оплати тощо.

На основі зібраної інформації проводиться відбір декількох найбільш привабливих і надійних кандидатів. Рекомендується замовити тестові зразки товарів у цих постачальників, щоб перевірити якість продукції та реальні терміни доставки перед початком повноцінної співпраці.[9]

За результатами тестування підприємець обирає одного або кількох постачальників для роботи та укладає з ними договір. Бажано мати резервного постачальника на випадок, якщо основний не зможе виконати замовлення вчасно (наприклад, через вичерпання запасів чи форс-мажор).

Нижче узагальнено найбільш типові проблеми, що виникають при пошуку та роботі з постачальниками-дропшиперами, а також наведено їх можливі наслідки та способи вирішення (мінімізації ризиків).

Таблиця 1.1 – Типові проблеми при пошуку постачальників дропшипінгу, їх наслідки та шляхи вирішення.

Типова проблема при пошуку постачальника	Наслідки для продавця
Ненадійний або фіктивний постачальник. Постачальник може виявитися недобросовісним: затримує відправлення без пояснень, не виконує зобов'язання або виявляється шахраєм.	Зрив термінів доставки, фінансові втрати (оплачені, але не відправлені товари), ризик отримати бан на Amazon за невиконання замовлень, втрата репутації через невдоволених клієнтів.
Низька якість товарів. Постачальник відправляє товари, що не відповідають опису або мають дефекти.	Збільшення повернень і скарг, негативні відгуки від клієнтів, погіршення рейтингу магазину. Продавець змушений нести витрати на повернення/заміну товарів.
Довгий час обробки та доставки замовлень. Постачальник повільно відправляє товар або використовує надто тривалі способи доставки.	Порушення обіцяних строків доставки клієнтам, що призводить до скарг і ануляції замовлень. На Amazon це погіршує метрики продавця (Late Shipment Rate) і може стати причиною санкцій з боку платформи.
Невідповідність вимогам Amazon (пакування, документи). Постачальник вкладає у посилки рахунки-фактури, логотипи або рекламні матеріали, що розкривають його особу, або не дотримується інших інструкцій (наприклад, відсутність потрібного штрих-коду).	Порушення правил дропшипінгу Amazon. Замовлення можуть бути затримані або повернені, акаунт продавця піддається ризику попереджень чи блокування. Також виникає плутанина у клієнта, який отримує «чужу» накладну.
Невизначеність запасів і цін. Постачальник не надає актуального статусу наявності товарів; ціни можуть раптово зрости.	Ризик продавати товар, якого немає в наявності (доведеться скасовувати замовлення, що шкодить показникам сервісу). Раптове підвищення закупівельної ціни «з'їдає» маржу або приводить до збитків, якщо ціну продажу фіксовано.

Джерело: сформовано автором на основі виконаного дослідження

Як видно з таблиці, успіх дропшипінг-бізнесу значною мірою залежить від належного вибору постачальників. Дропшипер фактично делегує зовнішньому партнеру критично важливі процеси – зберігання товарів, комплектацію і доставку замовлень. Тому мінімізація ризиків при виборі постачальника – пріоритетне завдання на старті бізнесу. Рекомендується віддавати перевагу постачальникам, які

вже мають позитивний досвід роботи на ринку (наприклад, перевірені виробники або дистриб'ютори), та вибудовувати з ними довгострокові партнерські відносини на основі взаємної вигоди і довіри. При стабільній співпраці постачальник також зацікавлений виконувати замовлення якісно, адже він отримує середній на 18% більший прибуток, продаючи товари через дропшиперів, ніж напряму через власні канали збуту

Автоматизація процесів дропшипінгу та потреба в інтелектуальній підтримці. Управління дропшипінг-бізнесом, особливо на масштабах великого інтернет-магазину або на багатьох майданчиках, пов'язане з обробкою значних обсягів даних і виконанням однотипних рутинних операцій. До таких операцій належать: моніторинг наявності товарів у постачальників і оновлення товарних позицій в магазині, коригування цін з урахуванням зміни закупівельної вартості та конкуренції, розміщення замовлень постачальникам і відстеження відправлень, обробка трек-номерів і сповіщення клієнтів, управління поверненнями тощо. Автоматизація цих процесів є необхідною умовою ефективності: без використання спеціалізованих програмних рішень дропшипер фізично не зможе в реальному часі відслідковувати сотні позицій і замовлень, що може призвести до помилок (наприклад, продаж неіснуючого товару чи запізніле оформлення замовлення постачальнику). Сучасні інструменти для дропшипінгу – такі як ERP-системи електронної комерції, додатки маркетплейсів, платформи управління замовленнями – дозволяють автоматично синхронізувати каталоги і залишки, розміщувати замовлення постачальнику натисканням кнопки, отримувати трекінг дані і навіть змінювати ціни за заданими правилами.[10] Наприклад, використання автоматизованих засобів моніторингу запасів та цін може запобігти ситуації, коли магазин пропонує товар, відсутній у постачальника, або продає з мінусовою маржею через непомічене підвищення оптової ціни.[11]

Окрім стандартної автоматизації, все більшої ваги набуває інтелектуальна підтримка процесів – впровадження елементів штучного інтелекту (ШІ) та машинного навчання для вдосконалення управління дропшипінг-бізнесом. Інтелектуальна система може аналізувати великі масиви даних (історію продажів,

поведінку клієнтів, ринкові тренди, рейтинги постачальників) і на основі цього підтримувати прийняття рішень. Зокрема, ШІ здатен прогнозувати попит на товари та допомагати обирати найбільш перспективні продукти для продажу; виявляти трендові товари, відслідковуючи популярність запитів і продажів у реальному часі, що дозволяє дропшиперу вийти на ринок раніше за конкурентів.

Такі пошуково-рекомендаційні алгоритми можуть стати в пригоді при формуванні асортименту. Інше завдання – оптимізація цін: системи з підтримкою ШІ можуть динамічно змінювати ціну товару залежно від конкурентного оточення та статистики продажів, щоб забезпечити максимальний прибуток, зберігаючи привабливість пропозиції.[12]

ШІ також успішно застосовується у покращенні клієнтського сервісу: чат-боти на основі штучного інтелекту здатні автоматично відповідати на типові запитання клієнтів щодо статусу замовлення, умов доставки, тим самим знижуючи навантаження на менеджерів і забезпечуючи 24/7 підтримку.

В цілому, інтеграція AI-інструментів у дропшипінг можна уявити як наявність «невидимих помічників», які беруть на себе рутинні процеси керування магазином – від аналізу ринку до обробки замовлень. За рахунок цього підприємець може зосередитися на стратегічних аспектах розвитку бізнесу: розширенні аудиторії, переговорів з новими постачальниками, покращенні маркетингу тощо.

Не менш важлива сфера – моніторинг і оцінка постачальників за допомогою інтелектуальних систем. Уявна «розумна» система підтримки дропшипінгу могла б автоматично збирати дані про виконання замовлень кожним постачальником (вчасність відправок, відсоток бракованих товарів, кількість звернень щодо гарантій), формувати рейтинг надійності постачальників і навіть передбачати ризики (наприклад, повідомити продавця, якщо якийсь постачальник почав затримувати відправки або отримує багато негативних відгуків). Така аналітика на основі даних допомогла б своєчасно вживати заходів – перейти на іншого постачальника або перемовитись про покращення сервісу з поточним.

Загалом, потреба в інтелектуальній підтримці продиктована зростаючою складністю управління дропшипінг-процесами в умовах сучасного електронного бізнесу. Конкуренція серед дропшиперів висока, маржинальність часто обмежена (середня націнка в дропшипінгу становить лише 10–30%, з яких на Amazon частина йде на комісії платформи). Тому підвищення ефективності через автоматизацію та оптимізацію – ключ до успішності. Інтелектуальна система, що підтримує процеси дропшипінгу, може виступати як система підтримки прийняття рішень для підприємця. Вона об’єднує дані про товари, постачальників, ринок і клієнтів, і на основі заданих правил чи навченої моделі пропонує найкращі дії: які товари додати в каталог, у якого постачальника їх замовити, яку ціну встановити, як пріоритизувати обробку замовлень тощо. Це дозволить масштабувати бізнес без втрати контролю над якістю сервісу.

1.2 Аналіз існуючих СШІ й інтелектуальних систем предметної області

Класифікація СШІ для e-commerce та дропшипінгу

Існуючі системи штучного інтелекту (СШІ) у сфері електронної комерції та дропшипінгу можна умовно поділити на декілька категорій за їх функціональним призначенням.[4] Першою категорією є системи підтримки прийняття рішень – програмні інструменти, що допомагають продавцям аналізувати ринок і обирати стратегії. Вони забезпечують аналітику щодо вибору товарів, оцінки їх продажного потенціалу, аналізу конкурентів тощо. Такі системи виступають своєрідними «помічниками» підприємця, надаючи необхідні дані для обґрунтованого вибору товарів і постачальників.

Друга важлива категорія – системи динамічного ціноутворення. Вони автоматично коригують ціни на товари в режимі реального часу на основі заданих критеріїв та аналізу даних.[6] Зазвичай такі інструменти використовують алгоритми й AI-моделі для визначення оптимальної ціни залежно від попиту, цін

конкурентів, залишків складу та інших факторів. Перехід від статичних цін до динамічних дозволяє продавцям залишатися конкурентними й максимізувати прибуток, швидко реагуючи на зміну ринкових умов.

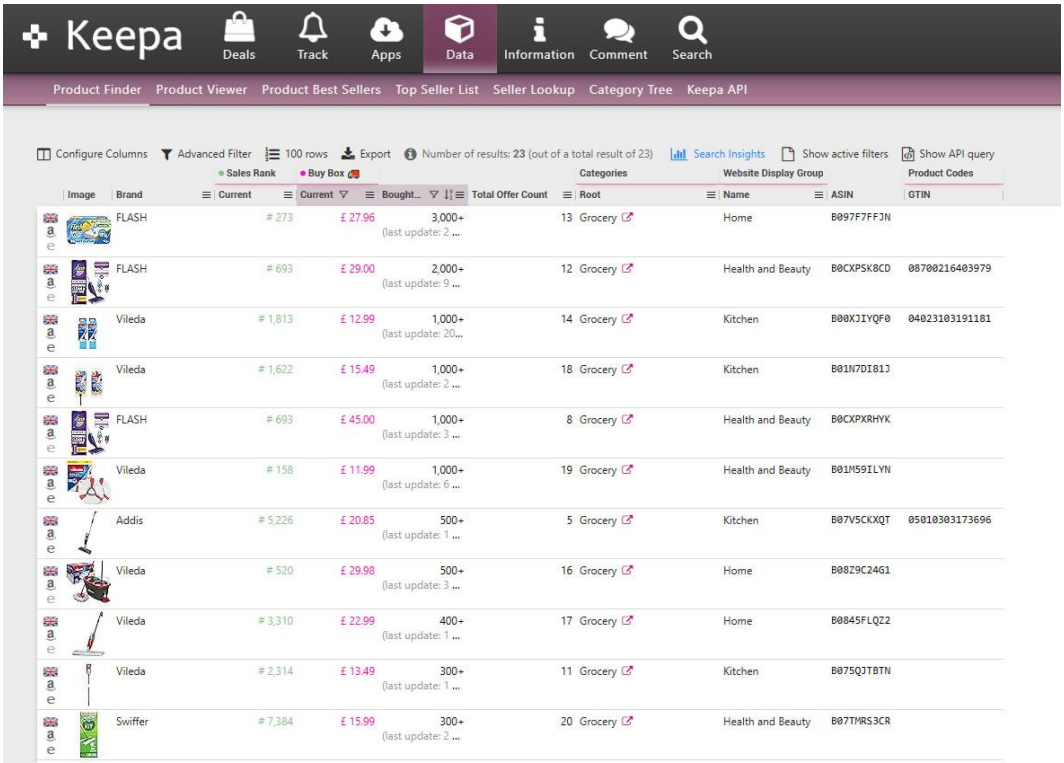
Третьою категорією є прогнозні моделі попиту.[27] Вони призначені для прогнозування обсягу продажів і тенденцій ринку на основі історичних даних і поточних трендів. Сучасні рішення з використанням AI здатні аналізувати великі масиви інформації (історію продажів, поведінку споживачів, сезонність, зовнішні фактори) та виявляти приховані закономірності для точнішого прогнозу майбутнього попиту. Це дає змогу ефективніше управляти запасами і логістикою, запобігати дефіциту чи надлишку товарів та своєчасно реагувати на зміну споживчих вподобань. Таким чином, інтелектуальне прогнозування попиту підвищує адаптивність і точність планування в e-commerce.

Четверта категорія – **інтелектуальні інструменти підбору постачальників та товарів.** До них належать системи, що спрощують пошук вигідних товарів і надійних постачальників для дропшипінгу за допомогою AI-алгоритмів. Такі інструменти аналізують різні джерела (онлайн-маркетплейси, каталоги постачальників, рейтинги) та можуть рекомендувати популярні або “трендові” товари, а також оцінювати постачальників за низкою критеріїв – ціни, рейтинг, швидкість доставки, наявність запасів тощо. Фактично, ці системи знижують роль випадковості при виборі ніші чи постачальника: рішення приймаються на основі великих даних і встановлених метрик, що підвищує шанс успішних продажів. В результаті використання таких інструментів власник магазину швидше знаходить “winning products” та оптимальних партнерів, економлячи час на ручний пошук і аналіз.

Огляд і характеристика ключових систем

На ринку існує ряд програмних рішень, які реалізують зазначені функції для підтримки процесів дропшипінгу, з різним ступенем автоматизації та «інтелектуальності». Розглянемо ключові з них: Keepa, Helium 10, Grabley, OAGenius, AutoDS та DSers, – їх можливості та особливості.

Keepa – сервіс моніторингу даних Amazon, що належить до категорії систем підтримки рішень. Keepa відстежує історичну інформацію про ціни та продажі для понад 5 млрд товарів на Amazon. Користувачі можуть переглядати графіки зміни ціни і рейтингу продажів (sales rank) за часом, налаштовувати сповіщення про зниження ціни, отримувати статистику по товару тощо. Таким чином, Keepa надає продавцю цінову аналітику і допомагає визначити оптимальні моменти для купівлі або змін цінової стратегії. Базові функції доступні безкоштовно через вебінтерфейс чи браузерне розширення, а розширені можливості (наприклад, повнотекстовий пошук товарів, API для доступу до даних) вимагають преміум-підписки (~€19 на місяць). Keepa підтримує API: розробники можуть інтегрувати його дані в власні системи (API доступ надається на платній основі).[15]



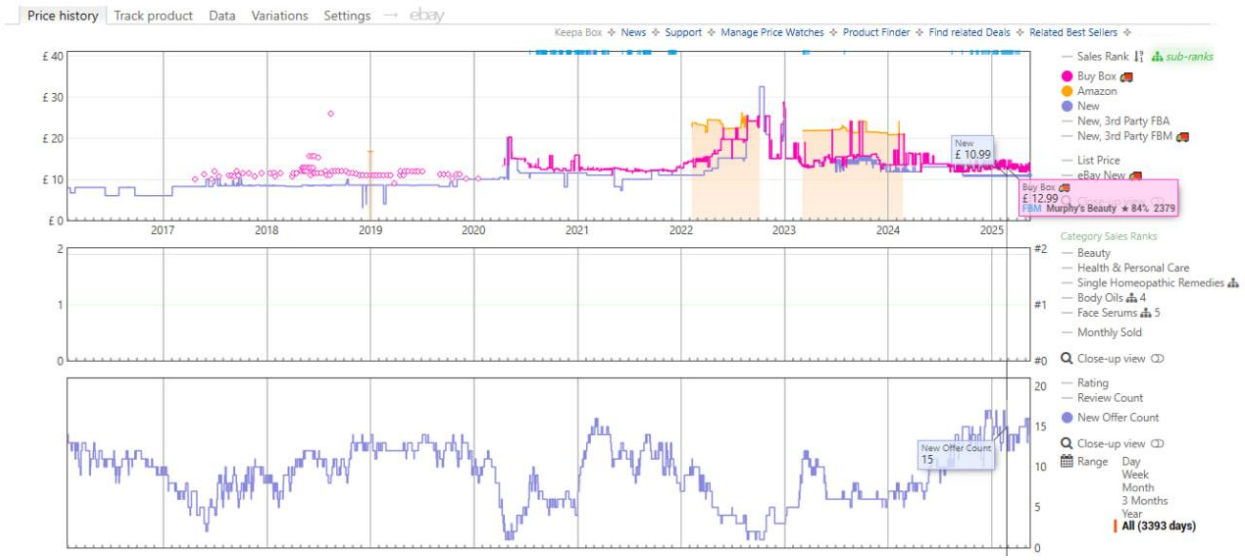


Рисунок 1.3-1.4 – Огляд функціоналу Keepa

Джерело: інтерфейс Keepa

Helium 10 – потужний все-в-одному інструмент для продавців на Amazon, що виконує роль системи підтримки рішень із елементами прогнозної аналітики. Helium 10 являє собою набір із 30+ інтегрованих утиліт для дослідження ринку та оптимізації продажів. Зокрема, платформа дозволяє шукати перспективні продукти (модуль Black Box для генерації ідей товарів, Trendster для аналізу трендів), підбирати ключові слова для лістингів (Magnet, Cerebro), відстежувати конкурентів і динаміку цін, оптимізувати контент товарних сторінок (функції на кшталт Scribbles, Index Checker) і навіть управляти рекламними кампаніями (Adtomic). Іншими словами, Helium 10 надає продавцю інтелектуальну підтримку на всіх етапах – від вибору ніші до аналізу ефективності бізнесу – через зручну єдину панель. Важливо, що цей сервіс орієнтований передусім на професійних продавців: він є комерційним (передбачено кілька тарифних планів вартістю від ~$29–39 на місяць і вище залежно від функціоналу). Прямого відкритого API для експорту даних Helium 10 не пропонує – доступ здійснюється через власний вебінтерфейс та інструменти інтеграції, надані розробником.[14]

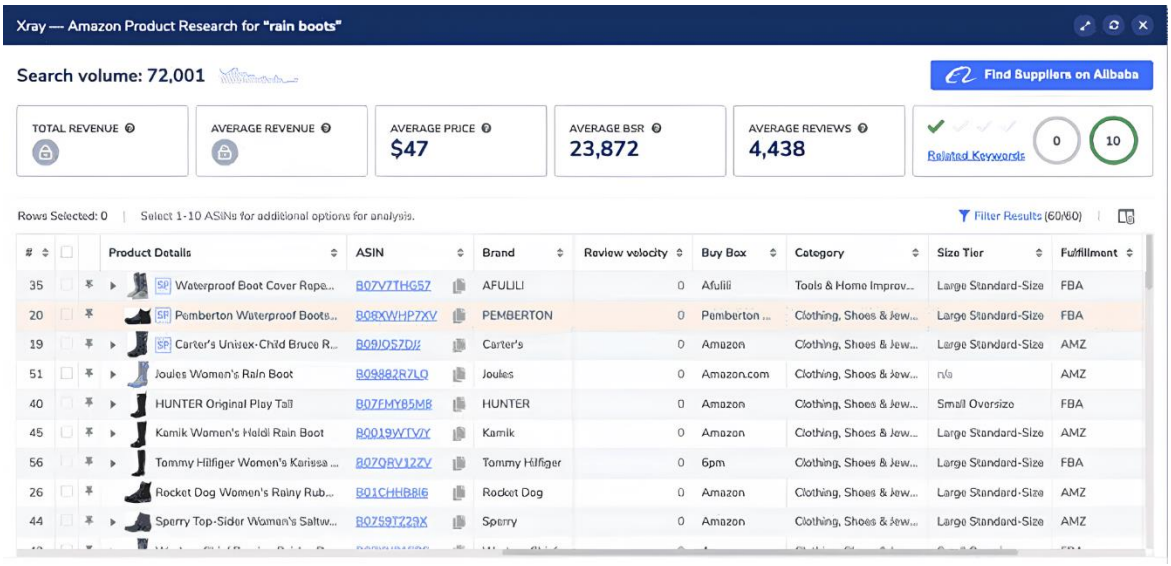


Рисунок 1.5 – Огляд функціоналу Helium 10

Джерело: інтерфейс Helium 10

Grabley – спеціалізований інструмент для пошуку товарів та моніторингу цін від постачальників, корисний у моделі дропшипінгу і онлайн-арбітражу. Grabley існує у вигляді розширення браузера Chrome і дозволяє швидко шукати товари одночасно на кількох маркетплейсах (Amazon, eBay, Walmart, AliExpress тощо) з розширеними фільтрами. Зокрема, користувач може відсіяти результати, які не підходять: приховати товари з позначкою Amazon Prime, виключити продукти, що продаються самим Amazon, встановити діапазон рейтингу бестселера (BSR) та інші критерії для звуження вибірки. Інструмент також відображає корисні деталі для кожного товару – наприклад, ASIN, наявність Buy Box, бренд, кількість продавців – і навіть може показувати таблиці маржинальності для оцінки потенційного прибутку. Grabley суттєво спрощує порівняння цін між різними майданчиками і пошук «нижчих» цін постачальників: за допомогою гарячих клавіш можна швидко копіювати та відкривати посилання товару на різних платформах. Ця система є прикладом інструмента підтримки рішень, орієнтованого на фазу підбору товарів і постачальників. Grabley умовно безкоштовна (розширення доступне для встановлення без оплати), хоча може мати обмежені платні функції; власного API не передбачено, адже всі операції виконуються інтерактивно в браузері.[13]

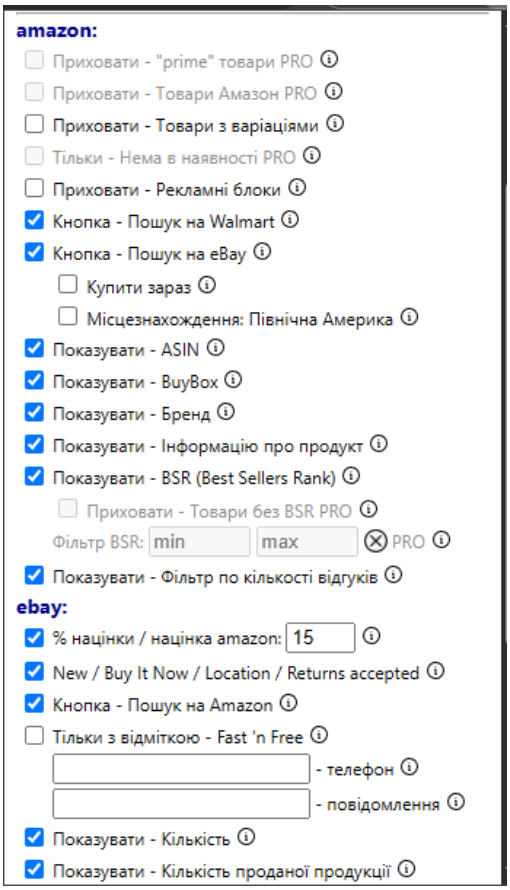


Рисунок 1.6 – Огляд функціоналу Grabley

Джерело: інтерфейс Grabley

OAGenius – комплексна платформа для автоматизації процесів дропшипінгу та онлайн-арбітражу, з акцентом на багатоканальне управління. OAGenius позиціонується як набір інструментів для продавців, що допомагає знаходити прибуткові товари, розміщувати лістинги, керувати запасами, відстежувати ціни і виконувати замовлення. Система складається з декількох модулів: OAGenius Gravity – веб-застосунок і розширення для пошуку товарів, яке в рази прискорює сорсинг (шляхом автоматизованого сканування сторінок постачальників і порівняння їх цін з цінами на Amazon); OAGenius Sage – інструмент для управління товарними позиціями та замовленнями (імпорт товарів на Amazon/Walmart, синхронізація наявності, оновлення статусів замовлень, облік прибутковості по SKU тощо). Важливою можливістю OAGenius є моніторинг цін постачальників: платформа відстежує зміну собівартості на сайтах-постачальниках і через API передає оновлену ціну на підключений сервіс репрайсингу (наприклад, інтеграція з інструментом Informed.co). Це дозволяє автоматично коригувати ціни продажу на

Amazon, щойно змінилася закупівельна ціна, та утримувати конкурентоспроможність в Buy Box.[17]

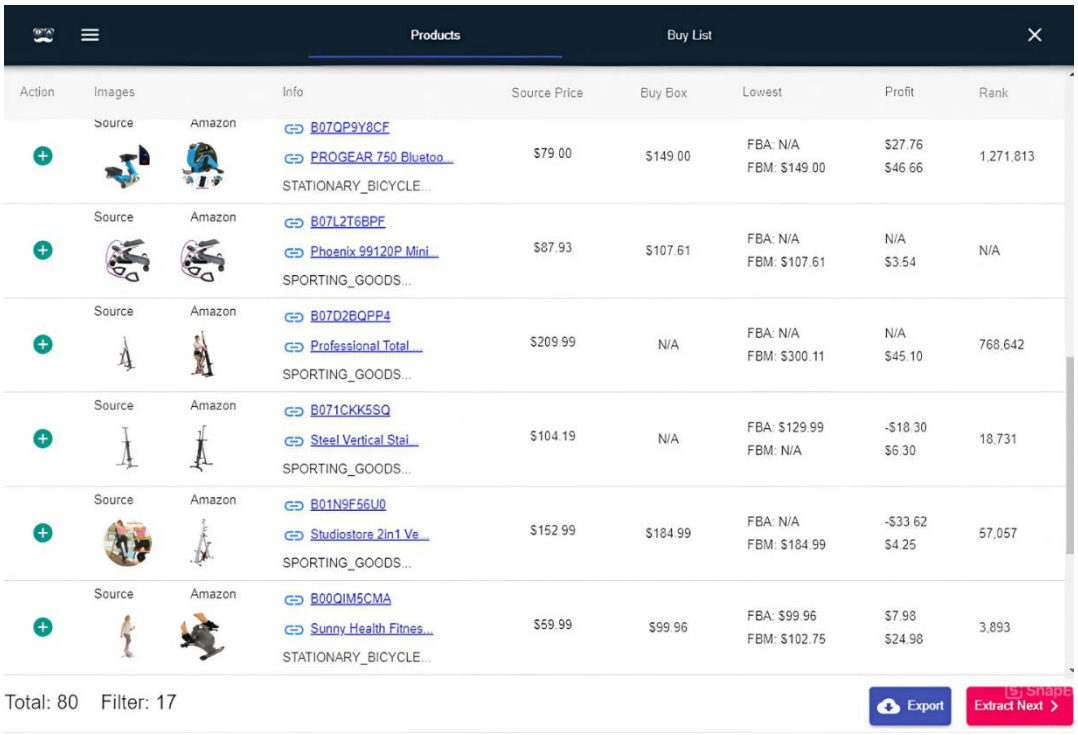


Рисунок 1.7 – Огляд функціоналу OAGenius

Джерело: інтерфейс OAGenius

AutoDS – одна з найпопулярніших хмарних платформ для повної автоматизації дропшипінгу. AutoDS пропонує «єдине вікно» для ведення магазину: система знаходить і імпортує товари з каталогів постачальників, моніторить ціни та складські залишки 24/7, і автоматично обробляє замовлення покупців. Платформа підтримує інтеграцію з багатьма торговельними майданчиками і постачальниками: з AutoDS можна керувати продажами на eBay, Shopify, Wix, Facebook, Etsy, Amazon тощо, імпортуючи товари від десятків популярних постачальників (Walmart, AliExpress, Home Depot та ін.). Серед інтелектуальних функцій AutoDS – інструменти для дослідження ринку: вбудований маркетплейс “winning products” та аналіз трендів (зокрема, модулі для відстеження популярних товарів у рекламі TikTok, Facebook – так звані Ads Spy Tools). Крім того, платформа застосовує AI для рутинних задач: генерації назви і опису товару на основі

ключових слів (AI Title & Description generator) та автоматичної оптимізації цін під час листинга (встановлення націнки за заданими правилами). Таким чином, AutoDS не лише економить час продавця на управлінні магазином, а й допомагає підбирати асортимент і оптимізувати цінову політику. Весь процес – від додавання товару до його доставки – може бути налаштований як напівавтономний. Сервіс AutoDS є комерційним (передбачена підписка, залежно від кількості товарів і каналів продажу; нові користувачі можуть скористатися пробним періодом). API для прямого керування магазинами через AutoDS наразі обмежений – система розрахована на взаємодію через веб-інтерфейс, хоча для партнерських постачальників і розробників можуть надаватися окремі інтеграційні можливості.[18]

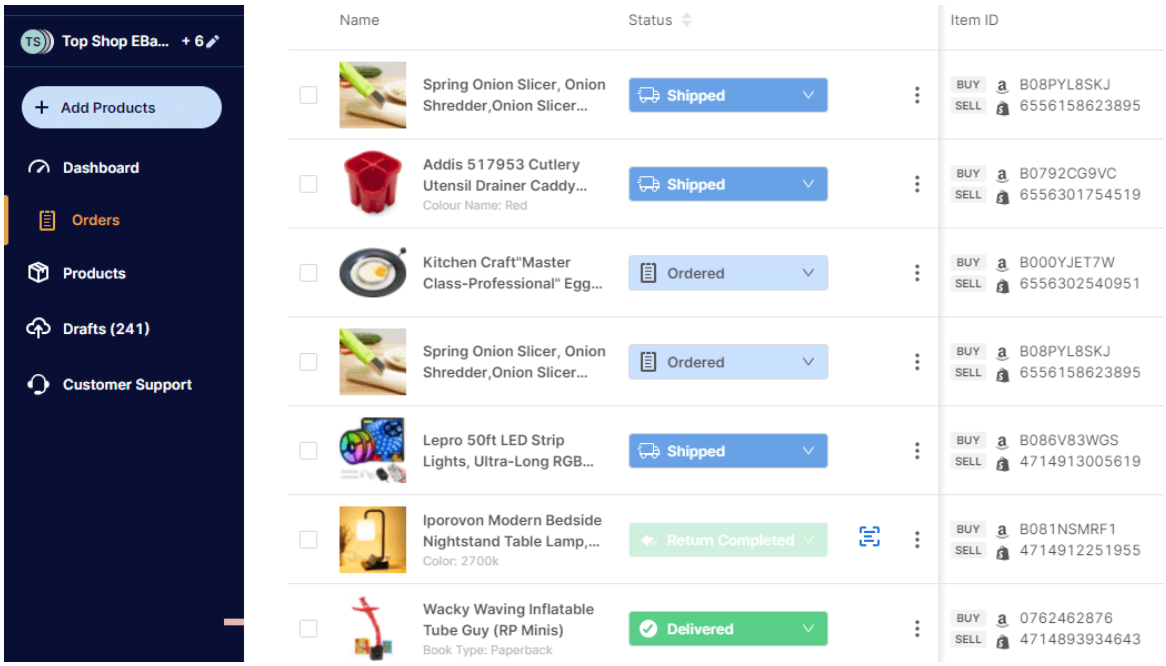


Рисунок 1.8 – Огляд функціоналу AutoDS

Джерело: інтерфейс AutoDS

1.3 Постановка проблеми та формування задач.

Дропшипінг на платформі Amazon пов’язаний з низкою суттєвих проблем, які ускладнюють успішне ведення бізнесу. До основних проблем належать:

1. Надійність постачальників. Постачальники не завжди виконують свої зобов’язання якісно та вчасно. Ненадійна робота постачальника (затримки відправлення, помилки в замовленнях тощо) прямо впливає на репутацію продавця та рівень обслуговування клієнтів. Підприємець-дропшипер залежить від стороннього постачальника у питаннях якості товару, дотримання строків доставки і актуальності запасів на складі. Якщо постачальник припускається помилок або не оновлює вчасно інформацію про наявність товарів, це може призвести до ситуації, коли клієнт замовляє товар, якого насправді немає в наявності, що шкодить довірі до магазину.[20]
2. Динаміка цін. Постійна зміна цін на товари з боку постачальників є ще однією значною проблемою. У моделі дропшипінгу продавець має невелику маржу, тому раптове підвищення закупівельної ціни постачальником може зробити продаж збитковим. За відсутності автоматизованого контролю дропшипер ризикує не встигнути скоригувати ціну у своєму магазині, що призводить або до втрати прибутку, або до необхідності скасування замовлення. До того ж, при роботі з багатьма постачальниками складно відстежувати актуальність цін на весь асортимент. За таких умов підтримання конкурентоспроможної ціни потребує значних зусиль і постійного моніторингу ринку[11]
3. Вибір товарів. Правильний вибір товарів для продажу визначає успіх дропшипінг-бізнесу, проте цей процес є доволі складним. Підприємцю необхідно проаналізувати попит на ринку, рівень конкуренції та прогнозовану маржинальність кожної потенційної позиції.

Дропшипінг зазвичай працює з невисокою націнкою, тому важливо обирати товари з достатньо високою маржею, щоб забезпечити прибуток. Продукти з низькою маржинальністю вимагають великих обсягів продажу для одержання істотного доходу, що не завжди можливо. Таким чином, підприємець стикається з труднощами у визначенні “вигідних” товарів – тих, що поєднують високий попит, невисоку конкуренцію та достатній рівень прибутковості.[31]

Обґрунтування необхідності системи. Перелічені проблеми зумовлюють потребу в інструментах, які допоможуть автоматизувати та оптимізувати процеси дропшипінгу. Зважаючи на ненадійність постачальників, коливання цін та складність вибору товарів, постає питання розробки інтелектуальної системи підтримки процесів дропшипінгу. Така система повинна слугувати засобом підтримки прийняття рішень для продавця, підвищуючи оперативність і обґрунтованість його дій. Зокрема, інтелектуальна система здатна в режимі реального часу відслідковувати ключові показники (рейтинги постачальників, відгуки, ціни, умови доставки тощо) та надавати рекомендації або автоматично позначати оптимальні варіанти для продажу. Створення подібної системи є виправданим, оскільки воно дозволить мінімізувати людський фактор при аналізі постачальників і товарних позицій, своєчасно реагувати на зміни ринкових умов та уникати збитків, пов’язаних з помилковим вибором партнерів чи товарів.[33]

Мета дослідження. Основною метою даної магістерської роботи є розроблення інтелектуальної системи підтримки рішень для дропшипінгу на платформі Amazon. Така система має забезпечити вибір надійних постачальників та ідентифікацію найбільш вигідних товарних позицій у робочій таблиці (Google Sheets). Передбачається, що система буде враховувати рейтинги продавців на платформі, кількість відгуків покупців, рівень маржі (рентабельності товару) та умови доставки при оцінюванні кожної товарної позиції. Іншими словами, метою є створення рішення, яке автоматично проаналізує зазначені параметри і підсвітить або відмітить у таблиці ті позиції, які відповідають заданим критеріям надійності та прибутковості. Це дозволить підприємцю швидко зосередитися на

найперспективніших товарах і постачальниках, підвищуючи ефективність бізнес-процесів дропшипінгу.[36]

Запропонована інтелектуальна система працюватиме за чітко визначеною логікою правил. Алгоритм функціонування матиме такий вигляд:

Збір даних про товари: користувач формує список посилань на потенційні товарні позиції (наприклад, товари з Amazon чи з веб-сайтів постачальників), які планується проаналізувати. Ці посилання заносяться в таблицю Google Sheets або інший вхідний список для обробки системою.

Автоматичне отримання параметрів: система автоматично збирає ключові параметри для кожного товару за вказаними посиланнями. До таких параметрів належать: рейтинг продавця (постачальника) та надійність його роботи, кількість відгуків і середня оцінка товару, поточна ціна і витрати на доставку, а також орієнтовна маржа (різниця між ціною постачальника і роздрібною ціною на Amazon, з урахуванням комісій платформи). Збір даних може здійснюватися через API Amazon (якщо доступно) або парсинг веб-сторінок постачальників та використання інших відкритих джерел інформації.[22]

Скоринг за заданими правилами: на основі отриманих параметрів система застосовує набір правил для скорингу – оцінювання кожної товарної позиції. Правила скорингу розроблені експертним шляхом і відображають критерії вигідності та надійності. Наприклад, правилом може бути: «Якщо рейтинг постачальника вище визначеного порогу та кількість позитивних відгуків перевищує N, а очікувана маржа становить не менше X%, то присвоїти товару високий пріоритет». Подібні правила враховують всі ключові фактори (репутацію постачальника, відгуки, маржу, умови доставки) і виводять інтегральну оцінку або категорію (наприклад, «Не рекомендований», «Нейтральний», «Не рекомендований»).[17]

Маркування в таблиці: відповідно до результатів скорингу система здійснює маркування позицій у Google Sheets. Найбільш вигідні та надійні для дропшипінгу товари можуть бути виділені певним кольором або позначкою в таблиці (наприклад, зеленим кольором для топ-позицій), тоді як позиції з низьким

рейтингом або недостатньою маржинальністю — іншим кольором (жовтим чи червоним). Таким чином, користувач отримує наочні підказки в робочій таблиці щодо того, які товари варто додати в свій асортимент на Amazon, а яких краще уникнути. Маркування оновлюється автоматично при зміні вхідних даних або при періодичному повторному аналізі, що дозволяє оперативно реагувати на зміну цін чи показників постачальників.[35]

Бренд	Amazon	Ebay	Ціна ebay	Кількість	Ціна amazon	Profit	Класифікація товару	Маржа
Peerless	https://www.amaz	https://w	55.50	2	125.43	42.63	Рекомендований	70.80%
Draw-Tite	https://www.amaz	https://w	130.00	2	289.00	95.93	Рекомендований	68.01%
Simplex	https://www.amaz	https://w	181.00	4	425.00	152.12	Рекомендований	77.46%
Connecticut Electri	https://www.amaz	https://w	261.00	1	580.00	192.42	Не рекомендований	67.95%
Seco-Larm	https://www.amaz	https://w	32.72	2	74.00	25.18	Рекомендований	70.92%
Buyers Products Cc	https://www.amaz	https://w	51.03	10	115.32	39.20	Нейтральний	70.80%
Hobart	https://www.amaz	https://w	99.20	1	225.00	76.87	Рекомендований	71.42%
Perko	https://www.amaz	https://w	94.32	5	209.00	69.04	Рекомендований	67.47%
Hobart	https://www.amaz	https://w	318.80	5	719.00	243.68	Не рекомендований	70.45%
Lennox	https://www.amaz	https://w	44.64	1	100.88	34.29	Не рекомендований	70.80%
General Tools	https://www.amaz	https://w	25.00	5	65.00	26.18	Рекомендований	96.50%
Xyron	https://www.amaz	https://w	29.80	8	75.00	29.17	Нейтральний	90.21%
Sea Gull Lighting	https://www.amaz	https://w	142.88	1	322.90	109.75	Рекомендований	70.80%
Globe Electric	https://www.amaz	https://w	42.84	1	96.82	32.91	Нейтральний	70.80%
Sea Gull Lighting	https://www.amaz	https://w	120.00	1	479.00	262.58	Рекомендований	201.67%
Robinair	https://www.amaz	https://w	497.63	1	995.00	275.97	Рекомендований	51.11%
Cutler-Hammer	https://www.amaz	https://w	129.00	3	291.53	99.09	Рекомендований	70.80%
Cutler-Hammer	https://www.amaz	https://w	79.63	1	159.00	43.98	Рекомендований	50.91%
Cutler-Hammer	https://www.amaz	https://w	300.00	1	1,199.00	657.68	Рекомендований	202.05%
Cutler-Hammer	https://www.amaz	https://w	50.00	1	113.00	38.41	Нейтральний	70.80%
Cutler & Hammer	https://www.amaz	https://w	95.00	7	214.69	72.97	Рекомендований	70.80%
Peerless	https://www.amaz	https://w	64.45	1	145.65	49.51	Рекомендований	70.80%
Swingline	https://www.amaz	https://w	41.00	9	92.66	31.49	Не рекомендований	70.80%
Swingline	https://www.amaz	https://w	18.50	1	55.00	25.03	Не рекомендований	124.69%
Swingline	https://www.amaz	https://w	180.00	1	399.00	131.88	Не рекомендований	67.53%
Swingline	https://www.amaz	https://w	22.50	1	61.00	25.61	Нейтральний	104.90%

Рисунок 1.9 – Приклад маркування в таблиці

Джерело: інтерфейс Google Sheets

Зазначимо, що використання логіки на основі правил (rule-based) забезпечує прозорість і передбачуваність роботи системи. На відміну від чорних ящиків складних AI-моделей, правила можна вручну налаштовувати під бізнес-вимоги, а результати їх застосування легко інтерпретувати. Це особливо важливо на початковому етапі, коли даних для навчання моделей може бути недостатньо або коли критерії відбору повинні точно відповідати бізнес-стратегії підприємця.

Рисунок 1.10 – Логіка роботи rule-based системи підтримки дропшипінгу

Джерело: розроблено автором

Отже, на основі проведеного аналізу сформульовано проблему підтримки процесів дропшипінгу на Amazon і обґрунтовано необхідність створення спеціалізованої інтелектуальної системи для її розв’язання. Визначено мету і основні функціональні можливості майбутньої системи, а також окреслено конкретні задачі, що підлягають вирішенню в ході виконання магістерської роботи. Реалізація запропонованої системи дозволить підвищити надійність та ефективність бізнес-процесів у дропшипінгу, мінімізувати ризики, пов’язані з вибором партнерів і товарів, та забезпечити конкурентні переваги для продавця на платформі Amazon. Заданий комплекс задач у подальших розділах буде детально розглянуто та вирішено, що підтвердить практичну цінність і доцільність розробленої системи.

1.4 Обґрунтування вибору підходів і технологій для проектування та створення СШІ

Вимоги до функціоналу системи

З урахуванням специфіки дропшипінгу на платформі Amazon, система повинна забезпечувати наступний основний функціонал:

- **Автоматичний збір даних:** автоматичне отримання ключових параметрів про товари та постачальників (ціна товару, рейтинг продавця, умови та вартість доставки тощо) без ручного втручання.[17]
- **Обробка та класифікація:** аналіз зібраних даних і класифікація постачальників на категорії (наприклад, надійний або ризикований) на основі визначених критеріїв. Ця класифікація дозволить виявляти небезпечних партнерів і відбирати надійних для співпраці.[20]
- **Маркування результатів:** відображення результатів класифікації у зручному вигляді, зокрема маркування постачальників у таблиці Google Sheets. Для кожного постачальника у таблиці повинна автоматично проставлятися відповідна помітка (статус), що спрощує подальший аналіз та прийняття рішень менеджером.[8]
-

Таблиця 1.2 – Переваги Google Sheets як інтерфейсу СШІ

Критерій	Оцінка	Коментар
Зручність	Висока	Знайома більшості користувачів
Гнучкість форматування	Висока	Можна додавати фільтри, умовне форматування
Інтеграція зі скриптами	Висока	Через Apps Script або API
Придатність для аналізу	Середня	Не підходить для складної візуалізації

Масштабованість	Середня	Існують обмеження на кількість рядків/стовпців

Джерело: розроблено автором на основі попереднього аналізу

Обґрунтування вибору Random Forest

У рамках розробки інтелектуальної системи для автоматизованої підтримки процесів дропшипінгу було проаналізовано низку методів машинного навчання, що застосовуються для вирішення задач класифікації. Серед них особливу увагу привернув алгоритм Random Forest (укр. «випадковий ліс») — один із найефективніших ансамблевих методів, що базується на побудові множини дерев рішень.[12] Його переваги такі:

Таблиця 1.3 – Порівняння моделей машинного навчання для класифікації постачальників

Критерій	Random Forest	Decision Tree	Logistic Regression
Точність	Висока	Середня	Середня
Схильність до перенавчання	Низька	Висока	Середня
Інтерпретованість	Низька	Висока	Висока
Швидкість навчання	Середня	Висока	Висока
Підходить для rule-based	Ні	Так	Так

Джерело: розроблено автором на основі попереднього аналізу

Random Forest є одним з найбільш популярних алгоритмів класифікації, що демонструє відмінні результати “з коробки” для широкого кола задач. Зокрема, він добре підходить для бінарної класифікації на зразок поділу постачальників на надійних/ризикованих.

Даний алгоритм ефективно працює з табличними наборами даних, які містять числові та категоріальні ознаки. Це важливо, оскільки параметри постачальників (рейтинг, відгуки, наявність доставки тощо) представлені у вигляді чисел або категорій.

Random Forest не потребує складного попереднього опрацювання даних. Зокрема, для нього не обов'язкове масштабування чи нормалізація ознак перед навчанням, оскільки дерева рішень нечутливі до монотонних трансформацій вхідних змінних. Алгоритм стійкий до шумів і пропусків у даних, що спрощує підготовчий етап.

Модель Random Forest надає засоби для оцінки внеску кожної ознаки у процес прийняття рішення. Після навчання можна легко отримати важливість ознак – умовну значущість параметрів, розраховану на основі зменшення невизначеності при розподілах у деревах. Це дозволяє проаналізувати, які фактори найбільше впливають на класифікацію постачальників, і надалі вдосконалювати модель або бізнес-стратегію з урахуванням цих факторів.[11]

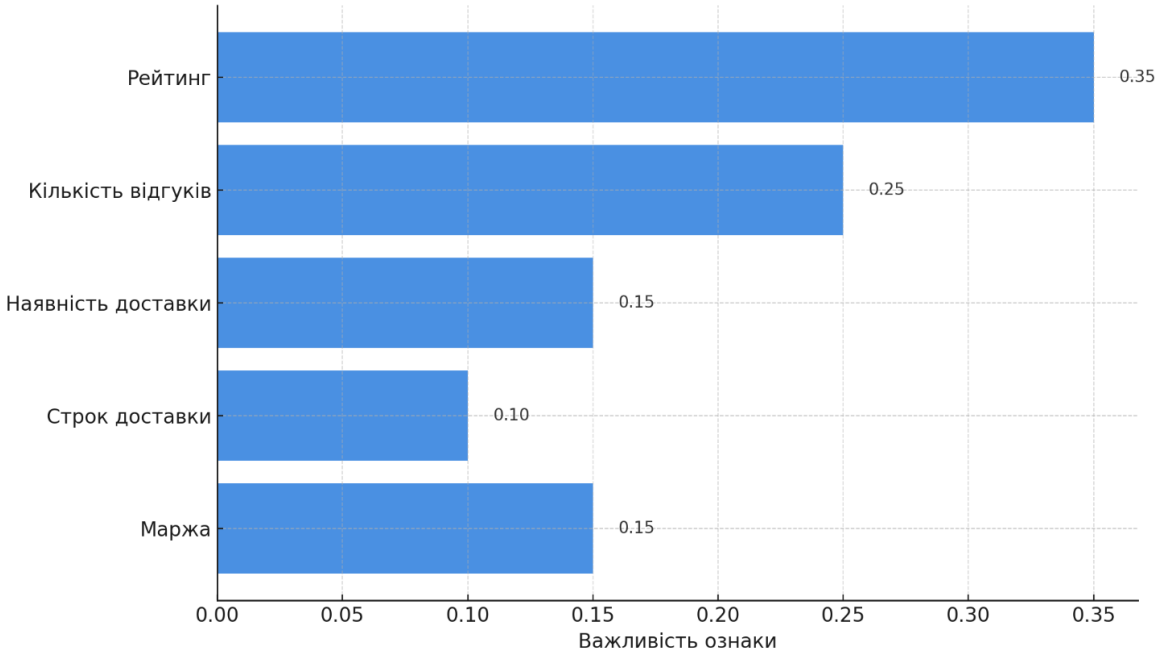


Рисунок 1.11 – Важливість ознак у моделі класифікації постачальників

Джерело: розроблено автором

Для навчання моделі та ухвалення рішення щодо надійності постачальників було визначено низку ключових ознак (факторів):

Рейтинг продавця. Середня оцінка постачальника (наприклад, за п’ятибальною шкалою). Високий рейтинг (ближче до 5) свідчить про добру репутацію продавця та якісний сервіс, тоді як низький рейтинг є ознакою можливих проблем. Згідно з практичними порадами, одними з кращих вважаються постачальники з рейтингом понад 4,5[3]

Кількість відгуків. Загальна кількість відгуків клієнтів про постачальника. Велика кількість відгуків сигналізує, що продавець має значний досвід і історію продажів. Постачальників, у яких майже немає відгуків, варто розглядати з обережністю, оскільки відсутність відгуків не дає впевненості в їх надійності.

Час доставки. Очікуваний час доставки замовлення до кінцевого клієнта (вимірюється в днях). Чим швидше постачальник може виконати доставку, тим вище його цінність, адже затримки з постачанням призводять до незадоволення клієнтів. Дана ознака дозволяє врахувати швидкість логістики при оцінці постачальника.

Рівень маржі. Рівень цінової маржі для товару, який забезпечує постачальник (різниця між ціною постачальника і ціною продажу на Amazon). Ця ознака відображає потенційну прибутковість співпраці з конкретним постачальником. Вища маржа означає більший прибуток для дропшипера, тоді як низька маржа може зробити співпрацю економічно невигідною.[10]

Частка негативних відгуків. Відсоток негативних (незадовільних) відгуків серед усіх відгуків про постачальника. Висока частка скарг або негативу є тривожним сигналом щодо якості товарів чи обслуговування. Постачальник, у якого постійно з’являються негативні відгуки, ймовірно, має проблеми з надійністю або якістю, що може створити ризики

Як працює Random Forest

Основні характеристики цього підходу такі:

Ансамбль дерев: модель формує ансамбль із багатьох дерев рішення (так званий “ліс” дерев). Кожне окреме дерево навчено на випадковій вибірці

навчальних даних (із застосуванням методики бутстреп-вибірки) і є незалежним класифікатором. Усереднення (агрегування) результатів багатьох дерев дозволяє підвищити загальну точність прогнозу та контролювати перенавчання моделі.

Випадкові ознаки при побудові: при побудові кожного дерева використовується випадковий піднабір ознак. На кожному вузлі дерева алгоритм Random Forest обирає не з усіх можливих параметрів, а лише з випадково вибраного підмножинного набору ознак. Такий підхід (відомий як метод випадкових підпросторів) забезпечує низьку кореляцію між окремими деревами і додатково підвищує стабільність ансамблю.

Голосування більшістю: для отримання кінцевого результату класифікації Random Forest поєднує виходи всіх дерев. Кожне дерево “голосує” за певний клас (надійний або ризикований постачальник), і фінальний прогноз визначається класом, який отримав більшість голосів серед дерев. Таким чином, якщо більшість дерев вказують на те, що постачальник ризикований, то ансамбль видасть рішення "ризикований", і навпаки.

Переваги над одним деревом: ансамблевий підхід Random Forest дає значно кращі результати, ніж використання одного дерева рішення. По-перше, об’єднання декількох моделей зменшує варіативність і робить прогноз більш стабільним і точним. По-друге, за рахунок усереднення некорельованих помилок різних дерев знижується ризик перенавчання моделі (overfitting) і підвищується її здатність до узагальнення. Іншими словами, Random Forest успадковує інтерпретованість дерев рішень, але водночас долає їхню вразливість до шуму та переобучення, забезпечуючи більш надійний результат класифікації.[12]

Висновки до розділу 1

У першому розділі було проведено всебічне дослідження предметної області, пов’язаної з автоматизацією дропшипінг-процесів на платформі Amazon із використанням засобів штучного інтелекту.

На етапі аналізу предметної області було окреслено ключові особливості бізнес-моделі дропшипінгу, ідентифіковано основних учасників процесу та

виявлено характерні проблеми, пов’язані з ручною обробкою даних, ненадійністю постачальників, коливанням закупівельних цін та складністю вибору товарів. Було сформовано загальне бачення цілей автоматизації в дропшипінгу та структури майбутньої інформаційної системи.

У підпункті 1.2 було здійснено порівняльний аналіз сучасних систем штучного інтелекту, які використовуються в e-commerce та дропшипінгу. Розглянуто функціональні можливості таких інструментів, як Keepa, Helium 10, Grabley, OAGenius, AutoDS і DSers. Виявлено, що більшість із них вирішують окремі завдання (моніторинг цін, пошук товарів, автоматизація фулфілменту), однак жодна з них не забезпечує комплексної оцінки надійності постачальників або гнучкої системи прийняття рішень на основі інтегрального скорингу. Це підкріплює актуальність розробки нової інтелектуальної системи з фокусом на аналітику та вибір постачальників.

У підпункті 1.3 було сформульовано проблему, яка полягає у відсутності інструментів для системної оцінки та класифікації постачальників у дропшипінгу, а також поставлено мету дослідження — створення інтелектуальної системи, яка зможе автоматично аналізувати дані про постачальників і маркувати найвигідніші позиції в таблиці Google Sheets. На основі цього було визначено ключові задачі дослідження: вивчення критеріїв оцінки, розробка структури системи, створення класифікаційної моделі та її інтеграція з робочим середовищем користувача.

У підпункті 1.4 обґрунтовано вибір підходу та технологій для реалізації системи. Визначено, що оптимальним підходом є поєднання rule-based логіки (для збору та фільтрації параметрів) з алгоритмом Random Forest — ефективною, інтерпретованою та легко реалізованою моделлю машинного навчання для класифікації постачальників.

Узагальнюючи викладене, можна стверджувати, що перший розділ сформував теоретичне та прикладне підґрунтя для реалізації магістерського дослідження. Визначено предметну область, актуальність проблеми, технічне обґрунтування підходів і сформовано логіку побудови інтелектуальної системи, яка буде розроблена у наступних розділах.

РОЗДІЛ 2 ХАРАКТЕРИСТИКА СШІ ТА ПОСТАНОВКА ЗАДАЧІ

2.1 Характеристика об’єкта дослідження

Об’єктом дослідження є інтелектуальна система підтримки процесів дропшипінгу на платформі Amazon. Інтелектуальна система в даному контексті розуміється як спеціалізоване програмне рішення, що використовує методи штучного інтелекту для автоматизації та оптимізації бізнес-процесів. Така система призначена для допомоги продавцям, які працюють за моделлю дропшипінгу, у виконанні рутинних завдань і прийнятті рішень на основі даних. Дропшипінг – це бізнес-модель електронної комерції, за якою компанії продають товари, не маючи власних запасів: після отримання замовлення від клієнта його виконання передається постачальнику, котрий доставляє товар покупцю, а продавець виступає посередником і заробляє націнку. В умовах платформи Amazon дропшипер несе відповідальність за своєчасне виконання замовлень і дотримання метрик якості сервісу, тому якість вибору постачальників є критичною. Інтелектуальна система підтримки дропшипінгу покликана мінімізувати ручні процеси та знизити ризики помилок при виконанні замовлень.

Таблиця 2.1 – Ризики ручного дропшипінгу

Проблема	Опис проблеми	Потенційні наслідки
Часті зміни цін	Ціни постачальника змінюються без попередження	Збитки, скасування замовлень
Наявність товару	Товар може зникнути після оформлення замовлення	Поганий сервіс, відміна продажу

Людський фактор	Помилки при ручному введенні, затримки	Зниження довіри, негативні відгуки
Логістичні збої	Втрата товару на етапі доставки	Витрати на повторну доставку, конфлікти
Репутаційні ризики	Невідповідність товару, інвойс з чужим брендом	Санкції Amazon, блокування акаунту

Джерело: розробка автора

Функціональні цілі системи

Основне призначення розроблюваної системи – підвищити ефективність і безпечність дропшипінг-операцій для продавця на Amazon. Виходячи з цього, можна виділити такі функціональні можливості інтелектуальної системи підтримки дропшипінгу:

Автоматизований аналіз постачальників. Система збирає та обробляє дані про потенційних постачальників товару. Зокрема, вона витягує ключові показники з онлайн-платформ (наприклад, з веб-сторінки товару на сайті постачальника): ціну товару, вартість і терміни доставки, рейтинг продавця або магазину, наявність товару на складі, відгуки покупців тощо. Ці дані формують інформаційну основу для подальшої оцінки постачальника.

Класифікація постачальників (надійний/ризикований). На основі зібраних даних система здійснює інтелектуальну класифікацію кожного постачальника за рівнем надійності. Постачальник, що відповідає вимогам (має високий рейтинг, позитивні відгуки, швидку доставку і т.д.), позначається як умовно надійний, тоді як постачальник із потенційно проблемними показниками (низький рейтинг, тривалі строки доставки, скарги клієнтів) отримає статус ризикований. Метою є автоматично попередити продавця про можливі ризики співпраці ще до оформлення замовлення. Така перевірка особливо важлива з огляду на політику Amazon: затримки або ненадійна відправка товарів здатні погіршити показники продавця (перевищення допустимого рівня запізнілих відправлень >4% може призвести до санкцій), а співпраця з неперевіреними постачальниками, які не

дотримуються вимог Amazon, ставить аккаунт під загрозу блокування. Отже, автоматична класифікація постачальників у режимі реального часу допомагає дотримуватися стандартів платформи.

Маркування та відображення результатів у таблиці. Результати аналізу та класифікації наочно відображаються для користувача. Система інтегрована з таблицею (наприклад, Google Sheets), де продавець працює зі списком товарів і постачальників. Кожен проаналізований постачальник у таблиці отримує відповідне маркування: це може бути кольорове виділення комірки, спеціальна позначка або текстовий коментар у полі, що вказує на статус «надійний» чи «ризикований». Таким чином, користувач у зручному табличному форматі одразу бачить, яких постачальників варто використовувати, а яких – уникати. Крім того, можуть виводитися й інші розраховані показники (детальніше розглянуто далі), що спрощують прийняття рішень.

Розрахунок маржі та фінансових показників. Система автоматично обчислює орієнтовну маржу прибутку для кожного товару при роботі з тим чи іншим постачальником. Враховується собівартість товару у постачальника (ціна закупки + доставка від постачальника до клієнта або на склад Amazon) та ціна продажу на Amazon, встановлена продавцем. На основі цих даних обчислюється потенційний прибуток (маржа) з одиниці товару. Якщо маржа занадто низька або від’ємна (наприклад, через високі витрати на доставку чи комісію платформи), система може сигналізувати про це користувачеві (наприклад, індикатором у таблиці). Таким чином, продавець отримує підказку щодо економічної доцільності дропшипінгу конкретного товару від конкретного постачальника. Додатково система може враховувати комісію Amazon за реалізацію товару, податки та інші витрати, щоб забезпечити більш точний розрахунок чистого прибутку.

Перелічені функції працюють у комплексі, забезпечуючи всебічну підтримку продавця. В результаті виконання аналізу й обчислень, інтелектуальна система надає рекомендації щодо вибору постачальників та товарів, що дозволяє мінімізувати ризики бізнесу та покращити ключові показники діяльності (рівень обслуговування, прибутковість, швидкість виконання замовлень).

Користувачі та взаємодія з системою

Система розрахована на використання продавцями, що працюють на платформі Amazon за моделлю дропшипінгу. Типовим користувачем є підприємець або менеджер інтернет-магазину, який хоче ефективно керувати замовленнями та постачальниками без власного складу. Взаємодія користувача з інтелектуальною системою здійснюється через зручний інтерфейс, інтегрований із таблицею (Google Sheets або аналогічний інструмент). Обраний інтерфейс продиктований потребою у простому й знайомому середовищі: продавці часто ведуть облік товарів і постачальників у вигляді таблиць, тому інтеграція саме з Google Sheets робить систему інтуїтивно зрозумілою та легкою у впровадженні.

Процес роботи користувача із системою можна охарактеризувати так. Спочатку продавець вручну вносить у таблицю базову інформацію про товар або посилання на сторінку товару у постачальника, якого він розглядає для дропшипінгу. Це може бути URL товару на сторонньому маркетплейсі (eBay, AliExpress тощо) або на сайті оптового постачальника. Як тільки користувач додає нове посилання або запис у таблицю, система автоматично розпізнає введення (через тригер або скрипт, прив’язаний до таблиці) і починає збір даних про відповідного постачальника. Після опрацювання даних та класифікації (виконаної на бекенд-сервісі системи), результати повертаються в таблицю: у рядку, що відповідає введеному товару, з’являються заповнені поля з оцінкою надійності постачальника, розрахованою маржею, рекомендованим рішенням (наприклад, «рекомендується» або «ризиковано»), а також інша аналітична інформація. Користувач може переглянути ці результати і використати їх для прийняття рішення – наприклад, якщо постачальник позначений як ризикований або маржа занизька, шукати іншого постачальника або коригувати ціну товару.

Важливо, що взаємодія з системою відбувається у напівавтоматичному режимі: користувачеві достатньо забезпечити введення початкових даних (посилань), а далі система автономно виконує аналіз і надає вихідну інформацію. За потреби користувач може вручну коригувати деякі параметри або повторно запускати аналіз (наприклад, якщо він оновив дані чи хоче перевірити

постачальника повторно). Загалом же інтерфейс спроектовано таким чином, щоб максимально спростити робочий процес продавця – звести до мінімуму необхідність самостійно перевіряти постачальників, рахувати маржу тощо, і водночас залишити користувачеві контроль над кінцевими рішеннями.

Структура та компоненти інтелектуальної системи

За своєю структурою система підтримки дропшипінгу складається з декількох модулів (компонентів), кожен з яких відповідає за певний етап обробки даних або функціональність. Умовно архітектуру системи можна розділити на такі основні частини:

Підсистема збору даних (вхідні дані). Цей модуль відповідає за отримання вхідної інформації про товар і постачальника. Вхідними даними є, як правило, посилання на товар у зовнішнього постачальника або інші ідентифікатори, які дозволяють отримати дані про товар. Підсистема виконує веб-скрапінг або запити до API, щоб зчитати з веб-сторінки постачальника необхідні параметри: ціну товару, вартість доставки і приблизні строки доставки до цільового регіону, рейтинг постачальника (якщо доступно), кількість відгуків, наявність товару, інформацію про бренд або упаковку тощо. Для Amazon-контексту особливо цікавими є дані, що впливають на виконання замовлення: чи зможе постачальник швидко відправити товар, наскільки він надійний у виконанні зобов’язань, чи відповідає товар вимогам (наприклад, не є забороненим). Зібрані дані передаються у наступний модуль для обробки.

Модуль обробки та підготовки даних. На цьому етапі відбувається фільтрація, очищення та перетворення отриманих сирих даних у формат, зручний для аналізу. Наприклад, текстові значення рейтингу можуть бути конвертовані у числову шкалу, строки доставки – у кількість днів, валюта цін – переведена в узгоджену валюту (щоб порівняти з ціною на Amazon), а відсутні значення заповнені прийнятними замінами або за замовчуванням. Також модуль може виконувати додаткові обчислення: наприклад, розрахувати маржу прибутку на основі ціни постачальника і поточної ціни продажу. На виході формується набір

ознак (факторів) для кожного потенційного постачальника товару – це вектор параметрів, який далі буде подано на вхід класифікаційному модулю.

Класифікаційний модуль (ядро штучного інтелекту). Це центральна частина системи, що реалізує алгоритми машинного навчання для визначення надійності постачальника. У нашому випадку ядром слугує модель типу Random Forest. Random Forest – це ансамблевий метод класифікації, який будує велику кількість дерев рішень і визначає результат за більшістю голосів цих дерев. Такий підхід підвищує точність і стійкість моделі, знижуючи ризик перенавчання у порівнянні з використанням одного дерева рішень. Набір ознак, підготовлений попереднім модулем, подається на вхід моделі Random Forest, яка видає прогноз – до якого класу віднести постачальника (надійний чи ризикований). Крім самого класу, модель може генерувати ймовірність належності до кожного класу, що дозволяє ранжувати постачальників за ступенем ризику. Варто зазначити, що алгоритм Random Forest добре підходить для подібного завдання: він здатен врахувати відразу багато різнорідних факторів (таких як час доставки, рейтинг, ціна) і на основі комплексного аналізу оцінити загальну надійність постачальника. Згідно з літературними даними, використання Random Forest для оцінки постачальників дозволяє проаналізувати показники продуктивності постачальника (швидкість доставки, якість обслуговування, цінова ефективність тощо) та виявити найбільш надійних і вигідних партнерів. Це робить прийняття рішень більш обґрунтованим, оскільки система фактично навчена на історичних даних і надає експертну оцінку замість суб’єктивної інтуїції продавця.

Модуль результатів та інтерфейс. Після того, як класифікаційний модуль видав результат, інформація повертається у зрозумілому вигляді до користувача. Цей компонент відповідає за інтеграцію з зовнішнім інтерфейсом (у нашому випадку – таблицею Google Sheets). За допомогою скриптів або API Google Sheets, система автоматично заповнює відповідні поля в таблиці: виставляє мітку класу для постачальника, додає колірне маркування рядка (наприклад, зелений для надійного, червоний для ризикованого), вписує розраховану маржу в окрему колонку, а також може додатково навести пояснювальну інформацію. Наприклад,

якщо постачальник віднесений до «ризикованих», система може додатково зазначити, який саме фактор вплинув найбільше (низький рейтинг, тривалий час доставки чи інше) – це стане можливим завдяки аналізу важливості ознак у моделі Random Forest. Таким чином, інтерфейсний модуль реалізує зворотній зв’язок з користувачем, надаючи результати роботи AI-моделі у наочному і зручному форматі. Користувач, отримавши ці результати, може приймати остаточні рішення щодо подальших дій (працювати з постачальником чи ні, коригувати ціну, шукати альтернативи тощо).

Модуль навчання та оновлення моделі. Окремо варто виділити компонент, що відповідає за постійне навчання системи на нових даних. Інтелектуальна система передбачає, що модель Random Forest буде періодично перевнавчатися (або донавчатися) у міру накопичення даних про роботу з різними постачальниками. Кожен виконаний випадок дропшипінгу може надавати зворотний зв’язок для системи: чи було замовлення виконано успішно, чи не виникло проблем (затримки, скасування, негативний відгук від клієнта). Таку інформацію можна використовувати як нові тренувальні приклади для моделі: наприклад, якщо модель позначила постачальника як «надійного», а на практиці співпраця з ним призвела до затримок або скарг, то цей кейс додається в датасет як негативний приклад, що допоможе моделі скоригувати свої рішення надалі. Модуль навчання може працювати у фоні, періодично оновлюючи модель (наприклад, раз на тиждень чи місяць, або при надходженні суттєвої кількості нових даних). Таким чином, інтелектуальна система постійно вдосконалюється, адаптуючись до змін у середовищі дропшипінгу (нові правила Amazon, зміна ринку постачальників, сезонні коливання тощо). Ця властивість самонавчання дуже важлива, оскільки забезпечує актуальність і довговічність системи: вона не просто працює за наперед запрограмованими правилами, а еволюціонує на основі накопиченого досвіду.

2.2 Структура і характеристика СШІ та її компонентів.

Інтелектуальна система для підтримки процесів дропшипінгу на Amazon розроблена з модульною архітектурою, що об’єднує кілька взаємопов’язаних компонентів. Такий підхід забезпечує гнучкість і масштабованість рішення: кожен модуль виконує окрему функцію і може вдосконалюватися незалежно від інших. Система призначена для автоматизованого аналізу постачальників у моделі дропшипінгу, оскільки успіх бізнесу значною мірою залежить від надійності постачальників. Зокрема, ненадійний постачальник (низька якість, часті затримки чи відсутність товару) може призвести до незадоволеності клієнтів і погіршення показників продавця на Amazon. Тому система покликана спростити процес оцінювання постачальників та швидко визначати, наскільки надійним чи ризиковим є кожен із них.

Загальна архітектура системи. СШІ являє собою цілісне рішення, що інтегрується з хмарними сервісами та локальними ресурсами для збору, обробки і аналізу даних про постачальників. В основі – електронна таблиця Google Sheets (Google Таблиці), яка слугує простим інтерфейсом введення та виведення даних для користувача. Ця таблиця водночас виконує роль своєрідної бази даних для системи, зберігаючи введену інформацію та результати класифікації. За допомогою Google Sheets API програма може зчитувати дані з таблиці та записувати результати автоматично. Логіка системи реалізована в декількох модулях, які працюють послідовно: модуль введення приймає початкові дані, модуль збору параметрів автоматично отримує додаткову інформацію про постачальника, модуль обробки очищує й готує ці дані, класифікаційний модуль присвоює клас надійності, а модуль відображення повертає інформацію користувачу. Для зберігання історичних даних та моделі використовується локальна база даних.

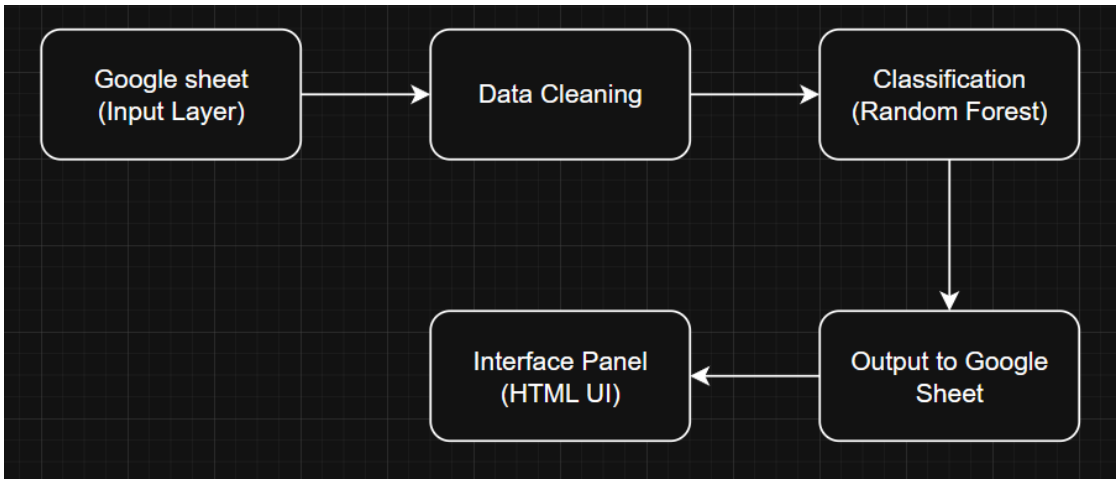


Рисунок 2.1 – Загальна архітектура СШІ для дропшипінгу

Джерело: розробка автора

Модуль введення даних

Модуль введення забезпечує отримання початкової інформації про потенційного постачальника, яку користувач вводить вручну в інтегровану таблицю Google Sheets. Відомості включають: посилання на товар або профіль постачальника (URL), початкову ціну товару у цього постачальника, а також доступну кількість одиниць (або мінімальне замовлення). Використання Google Sheets як інтерфейсу дозволяє легко вводити дані та одночасно слугує сховищем – електронна таблиця стає простим і знайомим середовищем для користувача. Дані, внесені в таблицю, автоматично зчитуються системою через скрипт-інтеграцію. Google Apps Script (вбудований скрипт для Google Таблиць) реагує на появу нових або змінених записів і ініціює подальші кроки обробки. Такий підхід усуває потребу у спеціальному додатку для введення – достатньо стандартного інструменту Google, доступного з будь-якого пристрою.

Модуль автоматичного збору параметрів

Після введення базових даних система переходить до автоматизованого збору додаткових параметрів про постачальника з веб-середовища. Модуль автоматичного збору здійснює парсинг веб-сторінок (веб-скрапінг) за наданим посиланням, витягуючи релевантну інформацію з платформи Amazon без потреби вручну відкривати сторінку. Зокрема, автоматично отримуються такі показники:

- Рейтинг продавця. Загальна оцінка постачальника на платформі (наприклад, відсоток позитивних відгуків або зірковий рейтинг).
- Кількість відгуків від покупців про цього продавця (показує масштаб і досвід постачальника).
- Інформація про доставку. Наявність та вартість доставки, чи підтримує постачальник швидку доставку (наприклад, Amazon Prime), географічні обмеження доставки.
- Строки виконання замовлення. Орієнтовний час відправлення і прибуття товару до клієнта, наявність товару на складі (щоб оцінити ризик затримок).

Модуль звертається до HTML-коду сторінки постачальника або товару і вилучає потрібні елементи (наприклад, блок, де зазначено рейтинг продавця та кількість відгуків). Цей процес відповідає сутності веб-скрейпінгу – автоматичного перетворення інформації з веб-сторінок, призначених для людського читання, у структуровані дані.

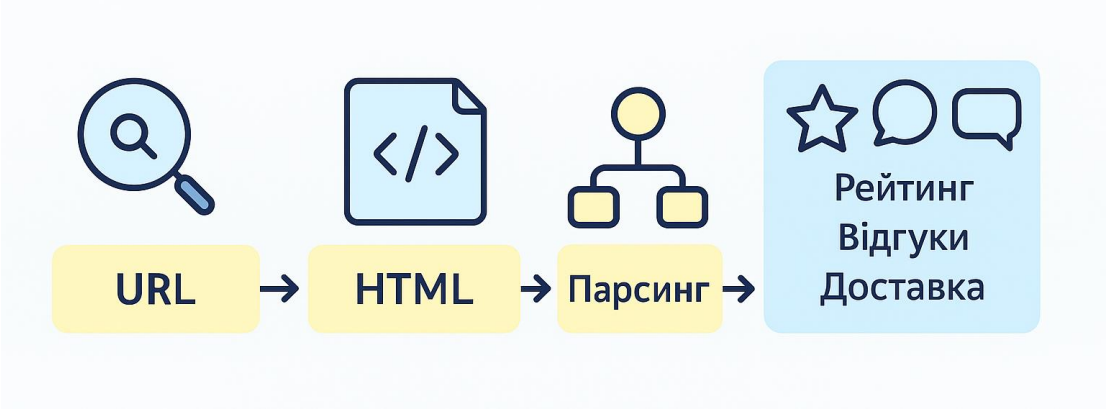


Рисунок 2.2 – Алгоритм роботи модуля автоматичного збору параметрів

Джерело: розробка автора

Отримані сирі дані потім структуруються у внутрішньому форматі (наприклад, числове значення рейтингу, кількість відгуків як ціле число, строки доставки у днях тощо). Автоматизований збір даних позбавляє користувача рутинної перевірки постачальників вручну і мінімізує людський фактор. Важливо,

що при веб-скрапінгу невпорядковані текстові дані (HTML-код) конвертуються у впорядковану форму, придатну для аналізу, яку можна зберігати в локальній базі або таблиці. Такий підхід значно підвищує швидкість оцінки постачальників, дозволяючи оперативно опрацювати велику кількість потенційних джерел товару.

Модуль обробки і нормалізації даних

Інформація, отримана з різних джерел (від користувача та з веб-скрапінгу), може бути неоднорідною, містити пропуски або шум. Модуль обробки даних виконує очищення, фільтрацію та нормалізацію зібраних параметрів перед подальшим аналізом. На цьому етапі система усуває дублікатні або нерелевантні записи, перевіряє дані на повноту та узгодженість формату. Наприклад, якщо рейтингове значення відсутнє або представлено нечисловим текстом, воно позначається як пропущене і обробляється відповідно (наприклад, заміною на спеціальний маркер чи середнє значення). Цілісність і якість даних критично важливі, адже «нечисті» дані можуть спотворити роботу алгоритму класифікації. Як зазначається в літературі, очищення даних є необхідним кроком у ML-проєктах: воно забезпечує точність, узгодженість та відсутність помилок у даних, що безпосередньо впливає на надійність моделі.

Нормалізація даних включає приведення показників до єдиного формату та шкали. Наприклад, текстові поля можуть бути переведені до нижнього регістру чи уніфікованого словника категорій, числові значення – до потрібного діапазону. Можливо, різні постачальники зазначають ціну в різних валютах – система конвертує їх у базову валюту для коректного порівняння. Так само строки доставки можуть бути перетворені в числове значення (кількість днів). Структуровані й нормалізовані таким чином дані придатні для машинного аналізу: вони подаються на вхід класифікаційної моделі в єдиному форматі та одиницях виміру. Окрім того, модуль обробки може виконувати виділення ознак – розрахунок додаткових індикаторів на основі наявних даних (наприклад, прапорець «Prime» як булева ознака швидкої доставки, або категоризація рейтингу на високий/низький). На виході цього модуля формується очищений набір ознак постачальника, який

відображає його надійність за різними критеріями (репутація, масштаб діяльності, швидкість доставки тощо).

Класифікаційний модуль (Random Forest)

Оброблені та підготовлені дані надходять до класифікаційного модуля, який здійснює автоматичну оцінку надійності постачальника за допомогою моделі машинного навчання. У ролі ядра інтелектуальної системи виступає алгоритм Random Forest (випадковий ліс) – ансамблевий метод ML для розв'язання задач класифікації. Він будує під час навчання велику кількість дерев рішень і на основі їхнього голосування визначає підсумковий клас. У нашому випадку модель Random Forest класифікує постачальників на два класи: «надійний» (умовно позначає постачальника з прийнятним рівнем ризику) або «ризикований» (ненадійний партнер, з яким пов'язана висока імовірність проблем).

Вибір Random Forest зумовлений його високою точністю та стійкістю до шумових даних: цей алгоритм поєднує виходи багатьох дерев, зменшуючи варіативність окремих прогнозів і ризик перенавчання. Модель здатна врахувати різнорідні характеристики (числові, категорійні ознаки) постачальника і виявити нелінійні залежності між ними та підсумковою надійністю. Перед тренуванням моделі необхідно підготувати навчальний набір даних: історичні випадки постачальників, для яких відомо, наскільки добре вони себе зарекомендували. На основі цього набору Random Forest будує ансамбль дерев, де кожне дерево навчається на випадковій підвибірці ознак і прикладів, що підвищує узагальнюючу здатність алгоритму.

Самонавчання моделі. Система передбачає можливість поступового поліпшення класифікатора в процесі експлуатації. Нові дані про постачальників (наприклад, результати співпраці з ними: виконані замовлення, отримані оцінки) накопичуються у локальній базі. Ці дані можуть періодично використовуватися для перенавчання моделі Random Forest, аби вона адаптувалася до актуальної інформації та потенційних змін ринку. Таким чином, класифікаційний модуль є самонавчальним у сенсі поступового вдосконалення: зі збільшенням обсягу даних точність розпізнавання надійних/ненадійних постачальників зростає. Модель та

необхідні для її роботи дані зберігаються локально (на сервері або комп'ютері розробника) – це забезпечує конфіденційність комерційної інформації та можливість роботи офлайн.

Модуль відображення результатів

Після визначення класу постачальника результати автоматично повертаються до користувача через Google Sheets. Модуль відображення бере прогноз, сформований класифікатором, та подає його у зручній формі поруч із початковими даними. У таблиці з'являється відповідна позначка надійності постачальника – наприклад, текстовий тег «Надійний» або «Ризикований», або умовний індикатор. Одночасно може застосовуватися колірне виділення: рядок або клітинка з результатом автоматично забарвлюється (наприклад, зелений фон для надійних, червоний для ризикових). Google Sheets дозволяють налаштовувати умовне форматування, за яким колір фону або тексту комірки змінюється при виконанні заданої умови. Це можна реалізувати або засобами самої таблиці (за допомогою вбудованих правил форматування), або через Apps Script, який після отримання результату змінює формат комірки програмно.

Відображення результатів безпосередньо в Google Таблиці робить систему максимально прозорою для користувача – оцінка постачальника з'являється поряд з інформацією про нього. За потреби, можна виводити також деталі: які саме показники вплинули на рішення (наприклад, «низький рейтинг: 80%» або «мало відгуків: 5»), щоб користувач розумів причини класифікації. Результати також дублюються у локальній базі даних або окремому аркуші Google Sheets для формування історії (логів) аналізу. Це дозволяє зберігати статистику спрацювань системи і згодом використовувати ці дані для донавчання моделі чи додаткової аналітики.

Інтерфейс користувача

Хоча основним інтерфейсом системи є Google Sheets, додатково передбачено окремий модуль інтерфейсу користувача у вигляді панелі керування. Цей компонент покращує зручність роботи з системою та надає розширені можливості взаємодії. Інтерфейс реалізовано як окреме вікно або вкладка (наприклад, HTML-

сторінка, що запускається локально, або бічна панель у Google Sheets через HTML Service Apps Script). У панелі відображається історія аналізу постачальників – список раніше оброблених записів із їхніми результатами (датою перевірки, отриманим класом, ключовими показниками). Це дає змогу відстежувати, яких постачальників вже аналізовано і які рішення були прийняті.

Через інтерфейс користувач також може керувати роботою системи – наприклад, вмикати або вимикати автоматичний режим збору даних. Кнопка “Старт/Стоп” дозволяє призупинити сканування постачальників (якщо, скажімо, потрібно тимчасово відключити автоматичне парсинг через зміну умов або щоб не перевищити ліміти запитів). Також користувач може ініціювати примусовий аналіз – наприклад, повторно запустити перевірку для певного рядка чи оновити вже наявну інформацію.

Функція дублювання результатів передбачає можливість експорту або резервного копіювання висновків системи. Через інтерфейс можна, наприклад, зберегти звіт про усіх постачальників у форматі CSV або PDF, чи імпортувати результати до іншої системи. Таким чином, UI-модуль виконує роль інформаційної панелі та консолі керування, що підвищує прозорість і контроль над роботою СШІ. Користувач отримує не лише статичний висновок у таблиці, але й інструмент для моніторингу та налаштування процесу аналізу постачальників.

Взаємодія компонентів системи

Всі описані компоненти інтегровані у єдиний робочий процес, де вихід одних модулів стає входом для інших. Послідовність роботи СШІ можна описати такими кроками:

1. Ініціація аналізу. Користувач додає в Google Таблицю новий запис про постачальника (або кілька записів). Спрацьовує тригер (наприклад, onEdit у Apps Script) або користувач натискає кнопку “Аналізувати” у панелі керування. Система зчитує введені дані (URL, ціна, кількість тощо) і створює завдання на перевірку постачальника.
2. Збір даних. Модуль автоматичного збору отримує завдання та надсилає HTTP-запити до відповідних веб-сторінок Amazon. Наприклад, за URL

товару витягується блок з інформацією про продавця цього товару (назва продавця, рейтинг, відгуки). Якщо URL веде на профіль постачальника, парситься сторінка профілю. Дані, отримані у вигляді HTML, розбираються парсером: виділяються цільові показники (значення рейтингу, кількість відгуків, текст про доставку тощо). Модуль формує проміжну структуру даних (наприклад, у форматі JSON або як запис у базі) з усіма знайденими атрибутами постачальника.

3. Обробка і підготовка. Структура з сирими параметрами передається до модуля обробки. Тут виконується очистка: відкидаються зайві символи (наприклад, «%» в кінці числового рейтингу), текстові поля обрізаються до релевантних частин (наприклад, з тексту "Ships within 2-3 days" вилучається число 2-3 як діапазон днів). Заповнюються відсутні значення (якщо, скажімо, не знайдено якихось даних – можуть бути задані дефолтні значення або позначки). Далі дані нормалізуються: числа приводяться до потрібного формату (типу Integer чи Float), строки переводяться мовою моделі (наприклад, "Yes/No" -> 1/0). Якщо потрібно, модуль додає обчислені ознаки (флажки, категорії). Після цього підготовлені ознаки постачальника подаються на вхід класифікаційного модуля.
4. Класифікація. Класифікаційний модуль завантажує актуальну модель Random Forest (збережену локально) та виконує прогноз для отриманих ознак. Модель опрацьовує значення (рейтинг, відгуки, швидкість доставки тощо) через ансамбль дерев рішень і видає прогнозований клас – «надійний» або «ризикований». Окрім класу, модель може генерувати ймовірність належності до кожного класу (наприклад, 90% надійний, 10% ризикований), що в подальшому може бути використано для тоншого налаштування (наприклад, позначати «умовно надійний», якщо впевненість моделі невисока). Результат прогнозу разом із часовою міткою та, за потреби, проміжними даними, зберігається у

локальній базі даних (для історії) та передається до модуля відображення.

5. Вивід результатів. Модуль відображення оновлює Google Таблицю: у відповідному рядку, поруч з введеними користувачем даними, з'являється результуюча позначка. Це може бути здійснено через Apps Script, який має доступ до потрібного діапазону комірок і записує туди значення класифікації. Одночасно застосовується колірне форматування до цього рядка/комірки для візуального акцентування результату (зеленим – якщо постачальник надійний, червоним – якщо ризикований). Користувач, переглянувши таблицю, одразу бачить висновок системи. Також система оновлює внутрішній лог: заносить запис про виконаний аналіз до історії (в базі або окремому листі). Якщо панель керування відкрита, вона може динамічно відобразити новий запис в історії аналізу.
6. Зворотній зв'язок та уточнення. Користувач може на основі результату прийняти рішення щодо співпраці з постачальником (наприклад, додати його до списку схвалених чи відхилених). За необхідності користувач може внести правки – наприклад, якщо відомо, що модель помилилась, можна вручну позначити постачальника як надійного/ненадійного в окремому полі. Ця інформація згодом буде врахована при перенавчанні моделі (тобто здійснюється наставницький зворотній зв'язок, коли користувач підтверджує або спростовує вердикт системи). Через певний інтервал (наприклад, раз на місяць) модуль навчання моделі може запустити процес оновлення Random Forest, використовуючи накопичені в базі нові дані та коригування від користувача, аби підвищити майбутню точність.

Узагальнена технічна реалізація

Для реалізації описаної системи використано поєднання кількох технологій та мов програмування, оптимально підібраних під задачі кожного модуля:

Google Sheets та Apps Script. Google Таблиці слугують не лише інтерфейсом, але й хранилищем даних. За допомогою Google Apps Script (JavaScript-подібна мова сценаріїв для сервісів Google) реалізовано автоматизацію взаємодії з таблицею: скрипти відслідковують додавання/зміну даних, виконують запити до зовнішніх сервісів і оновлюють значення комірок. Apps Script також використано для створення користувацького меню та діалогових вікон у Google Sheets, що дозволяє інтегрувати панель керування прямо в інтерфейс таблиці. Ця хмарна платформа зручна тим, що не потребує встановлення і може виконуватися на стороні сервера Google, реагуючи на події у електронній таблиці.

Мова Python та бібліотеки. Основна логіка збору даних і машинного навчання реалізована на Python. Python обрано завдяки його багатій екосистемі для веб-скрейпінгу та ML. Зокрема, для парсингу веб-сторінок використовуються бібліотеки на кшталт BeautifulSoup або Scrapy (для розбору HTML і вибірки потрібних елементів), а для виконання HTTP-запитів – requests. Модель Random Forest навчено і застосовано через бібліотеку scikit-learn (або альтернативні, наприклад, XGBoost для ансамблевих методів). Python-скрипт може працювати локально або на сервері, періодично звертаючись до API Google Sheets для читання нових записів та запису результатів. Така архітектура відповідає принципу розподілу навантаження: Google Sheets забезпечує введення/виведення, а Python виконує інтенсивні обчислення і мережеві запити.

Локальна база даних. Для зберігання історичних даних та моделі використовується легка реляційна база (наприклад, SQLite) або файли у форматі CSV/JSON. База містить таблицю постачальників з усіма зібраними характеристиками та присвоєними класами, а також таблицю з журналом аналізів (час запуску, які постачальники перевірені, результати). Локальне зберігання гарантує, що навіть при відсутності інтернет-з’єднання або збоях сервісів Google основні дані не будуть втрачені. Крім того, це вирішує питання конфіденційності – комерційно чутлива інформація (показники надійності постачальників, власні нотатки користувача) лишається на стороні користувача і не передається стороннім сервісам.

Інтеграційні механізми. Зв’язок між Google Sheets і локальною Python-програмою здійснюється через Google API. Скрипт на Python використовує офіційний Google Sheets API для запиту даних з таблиці та запису результатів назад. Це потребує налаштування облікового запису та OAuth2 авторизації, після чого Python-скрипт діє від імені користувача таблиці. Альтернативно, можна використати веб-хуки або хмарні функції: Apps Script може робити HTTP-виклики (UrlFetch) до локального веб-сервісу, запущеного на Python, передаючи йому дані для аналізу, і отримувати від нього відповідь з результатом. Обраний підхід – пряме читання/запис через Sheets API – є більш простим і надійним, оскільки мінімізує кількість точок відмови.

Платформи розгортання. Система може бути запущена в різних середовищах: Google Apps Script виконується у хмарі Google, тоді як Python-скрипт з ML може працювати на локальному ПК, на сервері чи в хмарному середовищі (наприклад, Google Cloud Functions або AWS Lambda, за умови, що він має доступ до Sheets і необхідні бібліотеки). У процесі розробки і тестування скрипт запускався локально, а для постійної роботи може бути налаштований як сервіс, що працює у фоновому режимі. Важливо забезпечити синхронізацію: або скрипт перевіряє таблицю на наявність нових записів через інтервали часу, або ініціюється через виклики Apps Script. Обидва варіанти можливі – для спрощення у прототипі використовувався періодичний опитувальний механізм (polling) з боку Python.

Загалом обране технічне рішення демонструє поєднання переваг хмарних сервісів (доступність, простота інтерфейсу) та потужності локальних обчислень (гнучкість Python для ML). Розділення на компоненти за технологічною ознакою дозволило використати найефективніші інструменти для кожного завдання, що підвищує надійність і продуктивність системи.

2.3 Методи, моделі і моделювання процесів і елементів складних систем

2.3.1 Методи дослідження й синтезу компонент систем штучного інтелекту

При розробці інтелектуальної системи необхідно поєднувати методи дослідження (аналізу) та синтезу (побудови рішення). Системний аналіз дає можливість формалізовано описати складну задачу дропшипінгу, виділити її компоненти та критерії оцінки. Згідно з класичним визначенням, системний аналіз – це «сукупність методів, що ґрунтуються на використанні ЕОМ і орієнтованих на дослідження складних систем». Такий підхід поєднує як формальні (математичні) моделі, так і евристичні процедури. У контексті СШІ це означає, що на етапі дослідження визначаються вимоги до системи, збираються вихідні дані й оцінюються головні фактори впливу. Етап синтезу полягає у проектуванні компонент системи на основі отриманих результатів – тобто виборі ознак, моделей та архітектури, що найкраще вирішують поставлену задачу.

Методи дослідження, використані в роботі

Формування ознак на основі практичного досвіду автора: на основі практичної діяльності та знань з дропшипінгу визначено початковий набір характеристик постачальників (наприклад, рейтинг, кількість відгуків, строки та вартість доставки тощо). Використання експертних знань дозволяє швидше виділити релевантні ознаки та зменшити розмірність даних на етапі підготовки.

Спостереження та аналіз поведінки постачальників: у процесі роботи здійснювалося спостереження за тим, як постачальники виконують замовлення (час відвантаження, кількість повернень, якість упаковки тощо). Аналіз цих даних у поєднанні з бізнес-метриками дає уявлення про їхню стабільність і надійність.

Таке спостереження допомагає виявити закономірності, які не завжди формалізовані в наявних даних.

Підготовка навчального набору даних вручну (експертна розмітка): для навчання моделі побудовано якісний тренувальний датасет, де кожному запису постачальника вручну присвоєно етикетки відповідно до його показників.

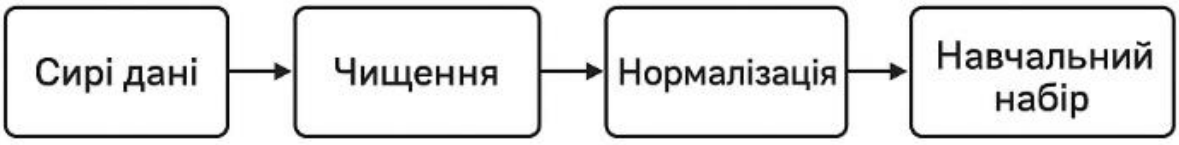


Рисунок 2.3 – Формування навчального набору для СШІ

Джерело: розробка автора

Відомо, що в машинному навчанні якісний набір розмічених даних є критичним для побудови ефективної моделі, а ручна розмітка забезпечує гнучкість і точність обробки складних випадків. Такий підхід дозволяє врахувати нюанси даних, які автоматизовані методи можуть пропустити, і підвищити точність подальшої класифікації.

Методи синтезу компонентів СШІ

Формування ознак (рейтинг, відгуки, доставка, строк): на етапі синтезу обрано набір ключових ознак для моделі. До них відносяться, зокрема, час доставки товарів, середній рейтинг якості продукції та інші релевантні показники постачальника. Ці метрики широко визнані критично важливими для оцінки постачальників: наприклад, «строки доставки, рейтинги якості тощо» називають ключовими даними для класифікації постачальників. Також до ознак можуть входити додаткові критерії (ціна, відсоток виконаних замовлень тощо), зібрані за допомогою аналізу поведінки.

Вибір моделі Random Forest (порівняння з Decision Tree, Logistic Regression): серед алгоритмів вибрано випадковий ліс (Random Forest) як основний метод

класифікації. Random Forest є ансамблевим методом – він «об'єднує результати декількох дерев рішень для забезпечення надійних прогнозів». Ансамблевий підхід здатний вирішувати недоліки окремого дерева (перенавчання, нестійкість до змін даних).

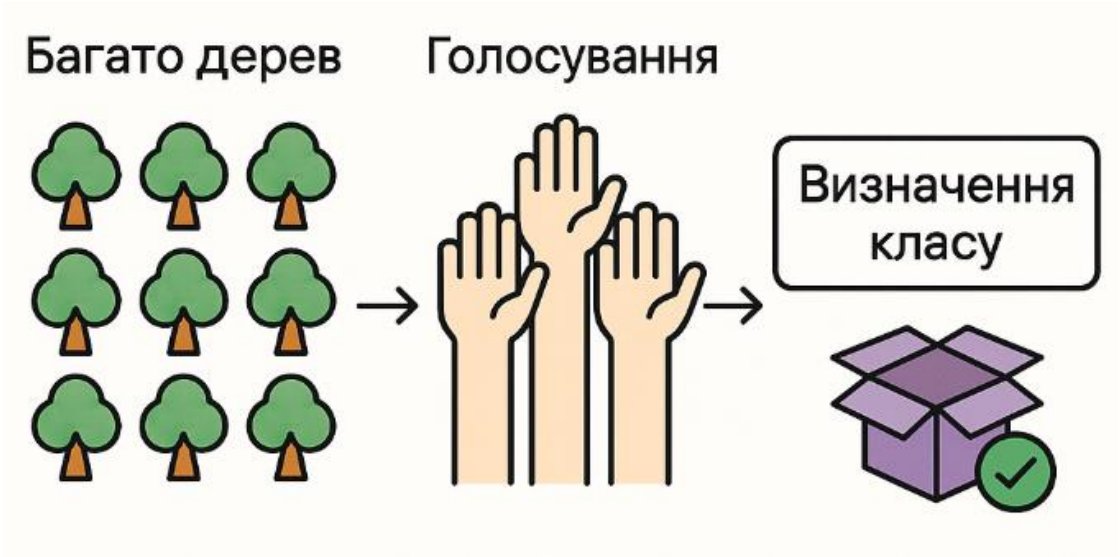


Рисунок 2.4 – Робота алгоритму Random Forest у класифікації постачальників

Джерело: розробка автора

Крім того, великі емпіричні дослідження показують перевагу RF над простими моделями: у бенчмарковому тестуванні Random Forest перевершив логістичну регресію за точністю приблизно в 69% випадків. Інші роботи також підтверджують, що Random Forest демонструє високу ефективність при класифікації постачальників, демонструючи «надійність та ефективність» у таких прикладних задачах. Порівняння з односпіними моделями (Decision Tree) та лінійною регресією показало, що RF забезпечує кращу загальну точність класифікації.

Архітектурна побудова логіки (збір даних → обробка → класифікація → відображення): запропоновано конвеєрний підхід до побудови системи. Спочатку реалізовано модуль збору даних із різних джерел (бази постачальників, API Amazon, логи угод), який очищує та нормалізує їх. Далі дані потрапляють до модуля класифікації (Random Forest) і, нарешті, результати (клас постачальника,

оцінки ризику) відображаються користувачу чи передаються в BI-систему. Такий етапний конвеєр відповідає загальним рекомендаціям з підготовки даних та побудови моделей у машинному навчанні: наприклад, «правильна підготовка даних критично важлива для точності класифікації», що включає чітке визначення цілей збору даних та їх очищення

Рисунок 2.5 – Синтез і взаємодія компонентів СШІ

Джерело: розробка автора

2.3.2 Моделі та методи оптимізації в системах штучного інтелекту

Оптимізація в контексті штучного інтелекту — це систематичне налаштування компонентів моделі для підвищення її продуктивності. Основною метою є покращення якості класифікації, збільшення швидкості обробки та забезпечення зручності взаємодії з користувачем. Зокрема, оптимізація може включати підвищення точності передбачень, зниження обчислювальних витрат і врахування відгуків користувачів для підвищення їх задоволеності та довіри до системи. Оптимізація відбувається на різних етапах розробки та експлуатації моделі (навчання, тестування та робота в реальному часі) шляхом корегування параметрів моделі і архітектури системи.

Основними цілями оптимізації в рамках розроблюваної системи дропшипінгу є:

- підвищення точності класифікації постачальників, щоб модель рідше помилялася з вибором надійних продавців;
- зменшення частки хибних рішень (помилкових спрацьовувань), що підвищує якість рекомендацій і довіру системи;
- прискорення реакції системи на запити користувача, забезпечуючи швидкий аналіз даних і видачу результатів.

Налаштування моделі Random Forest є важливою складовою оптимізації. Основні гіперпараметри, які впливають на продуктивність моделі, включають:

Таблиця 2.2 – Основні гіперпараметри моделі Random Forest

Гіперпараметр	Опис	Вплив на модель
n_estimators	Кількість дерев у лісі	Більше дерев — вища точність, але повільніше
max_depth	Максимальна глибина дерев	Глибші дерева — вища складність, ризик перенавчання
max_features	Кількість ознак для кожного розбиття	Менше ознак — менше перенавчання, вища стабільність
criterion	Функція розбиття (Gini, Entropy)	Впливає на точність розгалужень

Джерело: розробка автора

n_estimators (кількість дерев) – збільшення числа дерев зазвичай підвищує якість передбачень моделі (точність), але одночасно зростає обчислювальне навантаження

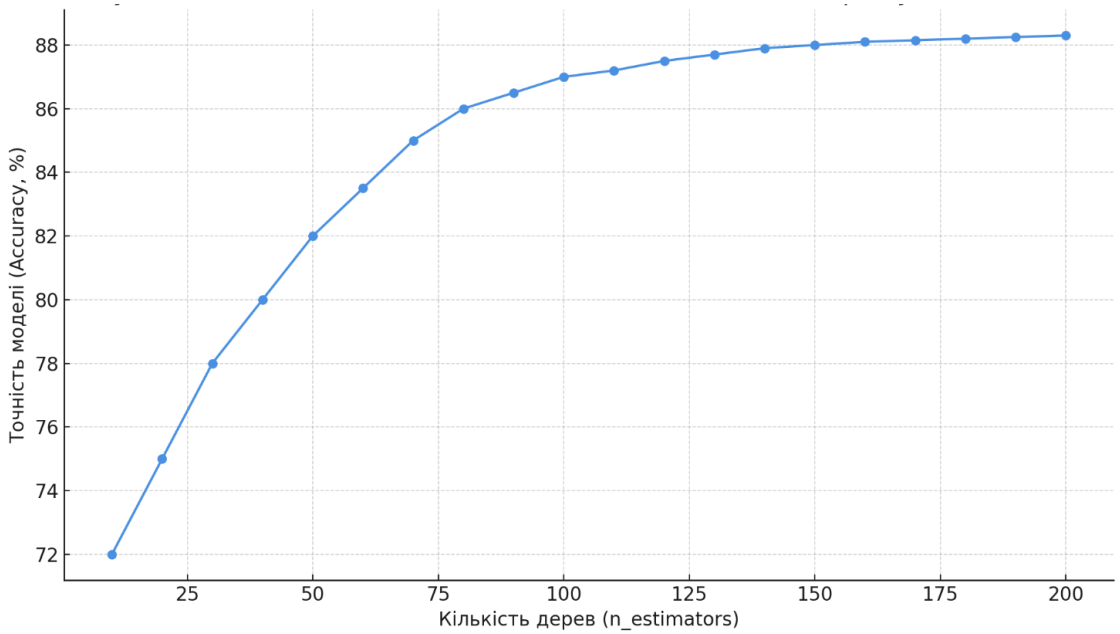


Рисунок 2.6 – Залежність точності моделі від кількості дерев у Random Forest

Джерело: розробка автора

max_depth (максимальна глибина дерев) – визначає глибину кожного дерева. Якщо вона занадто мала, модель може недонавчатися (недоцільна складність), якщо дуже велика – виникає ризик перенавчання (overfitting). Підбір адекватної глибини дозволяє збалансувати ці ефекти.

max_features (кількість ознак на розгалуження) – обмежує число ознак, які розглядаються при кожному розбитті вузла. Це допомагає контролювати переобучення і може підвищити стабільність моделі. Наприклад, варіанти «sqrt» або «log2» зменшують кількість випадковості при побудові дерев.

Пошук оптимальних гіперпараметрів (Grid Search, Randomized Search) – автоматизація підбору параметрів через повний або випадковий перебір конфігурацій. Інструменти типу GridSearchCV та RandomizedSearchCV з бібліотеки scikit-learn дозволяють ефективно знаходити найкращу комбінацію гіперпараметрів для підвищення точності моделі. Ці методи є перспективними для подальшої дооптимізації системи, особливо при роботі з великими наборами параметрів.

Оптимізація набору ознак (features) також суттєво впливає на результати. Random Forest забезпечує властивість feature_importances_, що оцінює важливість

кожної ознаки за зменшенням ентропії (Gini impurity) при розбитті даних. За допомогою цієї метрики можна виявити найінформативніші ознаки. Відсікаючи менш значущі ознаки, ми зменшуємо розмірність задачі та час навчання моделі без суттєвої втрати точності. Так, вибір лише найважливіших атрибутів дозволяє побудувати більш ефективні, інтерпретовані та продуктивні моделі. Наприклад, приклади показують, що після відбору ознак точність моделі може навіть зрости завдяки зменшенню надмірності та переобучення (ілюстрація підвищення точності з 89% до 94%)

На рівні архітектури системи оптимізація полягає у підвищенні пропускної здатності та масштабованості. У поточній реалізації обробка даних відбувається послідовно (синхронно), що може призводити до неефективного використання ресурсів і збільшення часу відповіді при зростанні навантаження. Для зниження затримок на рівні інфраструктури може бути реалізоване кешування результатів: повторні запити з однаковими параметрами можуть швидко повернути дані з кешу, що суттєво зменшує час відповіді системи. Крім того, перехід на асинхронну архітектуру дозволить обробляти кілька запитів паралельно. Асимптоматичне програмування забезпечує кращу масштабованість та вищу пропускну здатність: система зможе обслуговувати більше користувачів з меншими затримками завдяки неблокуючому виконанню операцій. У сукупності ці підходи (черговість обробки, кешування, асинхронність) сприятимуть збалансуванню навантаження та оперативності без значного ускладнення коду.

2.3.3 Методи та моделі управління в системах штучного інтелекту

У загальному розумінні, управління в системах штучного інтелекту означає набір механізмів, що забезпечують ефективний контроль функціонування системи, її адаптацію до змін навколишнього середовища та реакцію на непередбачувані

події. Метою такого управління є підтримка стабільності роботи алгоритмів, своєчасне оновлення моделей при надходженні нових даних та здатність швидко реагувати на відхилення від очікуваної поведінки. Іншими словами, система повинна вчасно відслідковувати зміни (наприклад, нові відгуки чи зміни ринкових умов) і коригувати власні параметри чи стратегії аналізу. У сучасних ШІ-системах широко застосовуються механізми зворотного зв'язку: наприклад, процес «людина в контурі» реалізує безперервний цикл оцінювання та уточнення результатів, що з часом робить алгоритм більш ефективним і точним.

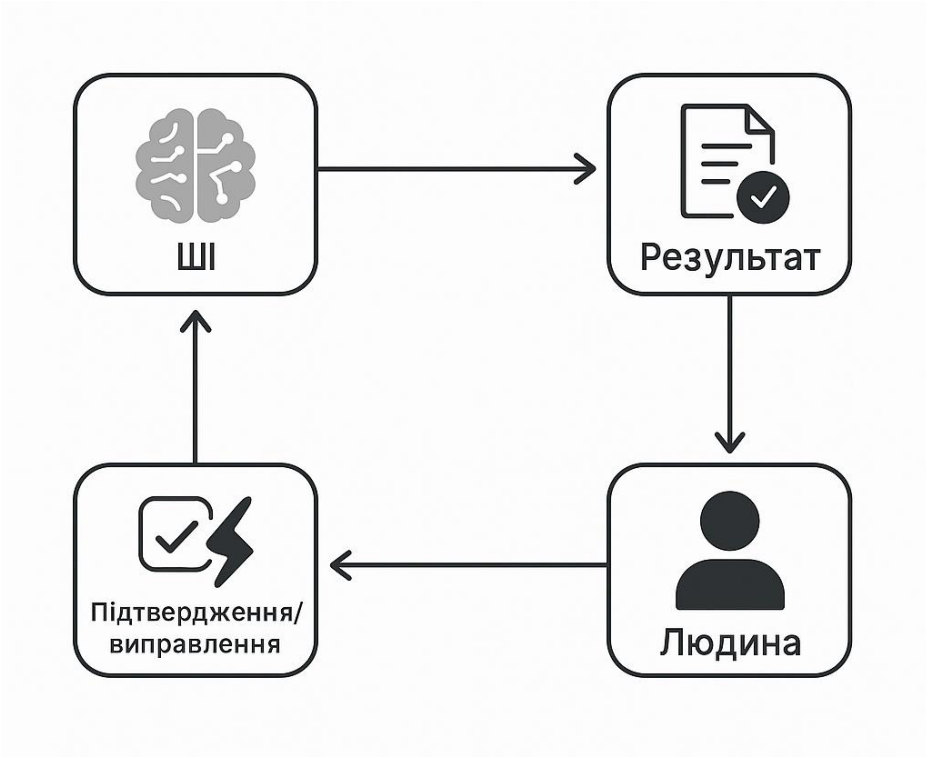


Рисунок 2.7 – Принцип роботи моделі людина в контурі у СШІ для дропшипінгу

Джерело: розробка автора

У запропонованій системі підтримки дропшипінгу управління реалізовано через координацію трьох основних компонентів: контроль процесу аналізу даних, регулярне оновлення моделей і взаємодію з користувачем. Зокрема, система може планово збирати дані про товари на Amazon (наприклад, характеристики, рейтинги та відгуки) і, накопичивши достатньо інформації, автоматично перебудовувати класифікатор на основі Random Forest чи інших алгоритмів. Уодночас за реалізацію

принципу людини в контурі (HITL) відповідає користувач: він може запускати аналіз товарів вручну, налаштовувати параметри перевірок або затверджувати результати класифікації. Такий інтегрований підхід, коли автоматична обробка доповнюється перевіркою людини, дозволяє досягти вищої точності рішень і підтримувати людські стандарти якості роботи системи.

Правило-базоване управління: логічні правила забезпечують попередній скринінг даних та ухвалення простих рішень. Наприклад, система налаштована не запускати детальний аналіз товару, якщо у нього менше ніж 10 відгуків, а товари з рейтингом нижче 85% позначати як високоризикові. Такий підхід гарантує прозорість і передбачуваність роботи – правила є явними й легко оновлюються (за потреби їх можна змінити або доповнити). Це забезпечує консистентність рішень і швидке налаштування системи під нові правила чи порогові значення

Тригери запуску: автоматичні сценарії або події можуть ініціювати процес аналізу. Наприклад, при надходженні нових даних (оновлення таблиці товарів чи появі нових відгуків) система сама запускає обробку, або ж аналіз запускається за розкладом. Одночасно передбачено ручне керування через інтерфейс: користувач може в будь-який момент ініціювати перевірку чи зупинити її.

Користувацький інтерфейс (UI): окремий модуль контролю дозволяє вмикати/вимикати автоматичний аналіз, переглядати поточні результати і налаштовувати процес. Такий інтерфейс забезпечує безпечне втручання людини: аналітик може призупинити аналіз або змінити параметри правив у реальному часі.

Ручне коригування результатів: користувачі мають можливість вручну редагувати автоматичний висновок системи (наприклад, змінювати категорію товару або позначати аномальні дані). Це додає ще один рівень контролю та дозволяє враховувати контекстні знання, яких може бракувати алгоритму.

Модель «Людина в контурі» (Human-in-the-loop): у цій моделі людина бере участь у процесі прийняття рішень ШІ. Наприклад, після первинної класифікації товарів користувач підтверджує або коригує категорію кожного випадку. Такий підхід суттєво підвищує точність і надійність системи – дослідження показують,

що системи з контролем людини часто працюють ефективніше, ніж чисто автоматичні.

Адаптивне управління: система реалізує механізм зворотного зв’язку для самонавчання. Вихідні дані (рейтинги, відгуки) і рішення користувача використовуються для донавчання моделі. Зокрема, алгоритм Random Forest як ансамбль дерев будує прогноз на основі голосування більшості. Зібрані нові мітки чи виправлення з боку користувача подаються назад у систему, яка періодично перебудовує класифікатор. Це дозволяє алгоритму адаптуватися до змін у даних та підвищувати якість прогнозів з часом.

Взаємодія модулів: інтерфейс керує викликом обчислювальних модулів (аналіз даних, класифікація) та фіксує результати у системі. Наприклад, після завершення аналізу модуль управління записує висновки в базу даних і формує повідомлення для користувача. Чіткий розподіл функцій між підсистемами забезпечує узгодженість дій і контрольований потік даних.

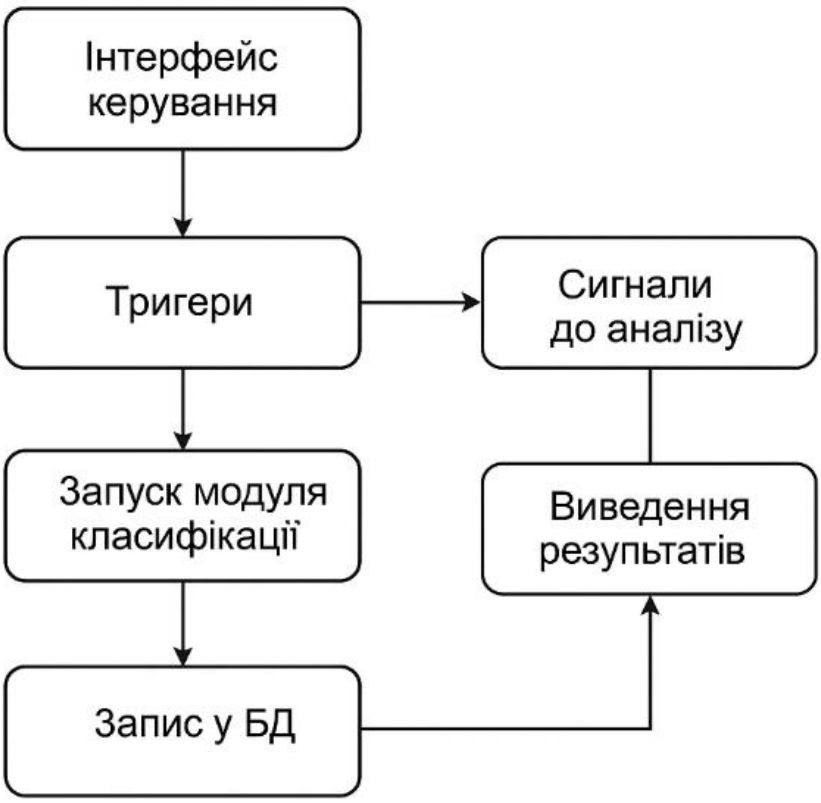


Рисунок 2.8 – Взаємодія компонентів управління в СШІ

Джерело: розробка автора

Висновки до розділу 2

Проектована система має модульну архітектуру, що складається з взаємопов’язаних блоків: збирання та обробки даних, аналітичного модуля, блоку прийняття рішень та інтерфейсу користувача. Кожен блок працює з визначеним набором ознак і формує дані у табличному форматі, що гарантує узгодженість всього рішення. Наприклад, інформація про товари та показники постачальників структурована у єдиній базі даних, що полегшує її аналіз. Такий підхід забезпечує цілісність системи та спрощує її масштабування

Для класифікації постачальників обрано модель Random Forest, оскільки цей ансамблевий алгоритм відзначається гнучкістю і простотою використання та дає хороші результати навіть без ретельного налаштування гіперпараметрів. Random Forest ефективно оперує з великою кількістю характеристик і знижує ризик перенавчання завдяки усередненню прогнозів із різних дерев. Вибір цього методу забезпечує надійність та стабільність результатів класифікації постачальників на основі наявних даних.

Логіка на основі правил введена як додатковий рівень управління. Rule-based компоненти забезпечують явність і прозорість процесу ухвалення рішень, адже правила визначені явно та зрозумілі. Це дозволяє користувачу швидко відстежувати та змінювати бізнес-правила в міру необхідності. Крім того, такий шар забезпечує послідовність і повторюваність ухвалених рішень (наприклад, валідації даних) незалежно від статистичних моделей.

Взаємодія користувача із системою реалізована через Google Sheets, оскільки ця платформа забезпечує інтуїтивно зрозумілий інтерфейс для роботи з даними. У ній дані відображаються у найпростіших табличних структурах — знайомому користувачу форматі таблиць, що спрощує їхню візуалізацію та редагування. Такий підхід дозволяє користувачу без спеціальних технічних знань оперативно вводити та коригувати інформацію (наприклад, параметри товарів чи профілі постачальників) і спільно працювати над даними.

Отже, запропоноване рішення є цілісним та водночас модульним. Кожен компонент реалізовано як незалежний сервіс із чітко визначеною функціональністю, що забезпечує узгоджену роботу всіх частин та спрощує інтеграцію і масштабування. Завдяки цьому нові можливості можна додавати без перебудови всієї системи, а збільшення навантаження приводить лише до розширення окремих модулів. В цілому, цілісність і модульність архітектури гарантують надійність, гнучкість і високу масштабованість запропонованого рішення.

РОЗДІЛ 3 РОЗРОБЛЕННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Моделювання та проектування бази знань прийняття інтелектуальних рішень і управління

База знань інтелектуальної системи – це централізований репозиторій структурованої інформації, який інтегрує дані з різних джерел та забезпечує швидкий доступ до релевантних даних, значною мірою підвищуючи якість прийнятих рішень. У контексті системи підтримки дропшипінгу база знань містить зібрані дані про потенційних постачальників з платформи eBay, що слугують навчальним набором для моделі класифікації. Вона включає ключові ознаки продавців (рейтинг, відгуки, умови доставки та ін.), які характеризують їх надійність. Кожен запис бази знань відповідає певному товарному лоту та містить інформацію про продавця, з якою проводиться надалі машинний аналіз. Джерелом вихідної інформації є відкриті дані з веб-сайтів eBay та API eBay Developers Program, що дозволяють отримати офіційні показники продавців (наприклад, детальні рейтинги та метрики).

Вхідна структура бази знань представлена CSV-файлом (див. рис. 3.1), що складається з кількох стовпців. Типовими полями є link (URL сторінки товару у продавця), Ціна eBay (рядкове подання ціни у локальній валюті), Кількість (кількість одиниць товару у замовленні). Наприклад, один рядок CSV може мати такий вигляд: "https://www.ebay.com/itm/145149796899?,"18,70",1. Ці поля заповнюються на початковому етапі збору даних через парсинг веб-сторінок. На рисунку 3.1 наведено приклад структури вхідного файлу з даними про постачальників.

```
link,Р¦С–РЅР° ebay,РљС–Р»СЊРєС–СЃС‚СЊ,Р РµР№С‚РёРЅРі,Р’С–РґРіСѓРєРё,Р‘РµР·РєРѕС€
https://www.ebay.com/itm/145149796899?,"18,70",1
https://www.ebay.com/itm/226587099075?,"17,30",1
https://www.ebay.com/itm/356426202461?,"11,60",3
https://www.ebay.com/itm/146395879495?,"20,00",1
https://www.ebay.com/itm/387667078484?,"48,71",1
https://www.ebay.com/itm/405452305787?,"21,30",1
https://www.ebay.com/itm/315499781166?,"25,00",1
https://www.ebay.com/itm/275031913916?,"20,00",2
https://www.ebay.com/itm/186758129539?,"12,70",1
https://www.ebay.com/itm/226556424674?,"51,90",9
https://www.ebay.com/itm/294394178858?,"19,70",1
https://www.ebay.com/itm/273007950164?,"44,00",10
https://www.ebay.com/itm/174183456027?,"193,00",3
https://www.ebay.com/itm/156355886289?,"169,00",1
https://www.ebay.com/itm/326011785042?,"175,00",1
https://www.ebay.com/itm/402749753525?,"70,00",1
https://www.ebay.com/itm/256824141387?,"149,10",1
https://www.ebay.com/itm/325750625155?,"46,65",1
https://www.ebay.com/itm/267045690857?,"13,00",1
https://www.ebay.com/itm/126876419658?,"19,50",1
https://www.ebay.com/itm/176729201395?,"12,00",1
https://www.ebay.com/itm/235948456461?,"15,00",10
https://www.ebay.com/itm/126938208597?,"29,20",1
https://www.ebay.com/itm/296921963969?,"11,00",1
https://www.ebay.com/itm/226590324789?,"32,00",10
https://www.ebay.com/itm/395359032004?,"113,00",1
https://www.ebay.com/itm/387349068758?,"180,00",1
https://www.ebay.com/itm/114666934390?,"94,24",1
https://www.ebay.com/itm/234527519367?,"46,43",1
```

Рисунок 3.1 – Приклад структури вхідного CSV-файлу з інформацією про постачальників

Джерело: розробка автора

Система автоматично витягує з веб-сторінок eBay наступні ключові параметри для кожного продавця:

- Рейтинг продавця – загальна оцінка надійності (наприклад, відсоток позитивних відгуків або середньозважена оцінка зі шкалою 1–5 зірок). Детальні рейтинги в eBay встановлюються за шкалою від 1 до 5, де 5 зірок означає найвищу оцінку
- Кількість відгуків – число покупців, що залишили відгуки про цього продавця (показує масштаби та репутацію).
- Дані про доставку – інформація про вартість та наявність доставки, а також орієнтовні строки доставки товару до цільового регіону.

Зазвичай з HTML-сторінки витягуються рядкові дані про ціну доставки або метки «безкоштовна доставка».

Операція збору цих даних здійснюється шляхом веб-скрапінгу: парсер вибирає з HTML-коду сторінки потрібні елементи (наприклад, блок із рейтингом продавця чи полі з інформацією про доставку) і формує відповідні рядки CSV. Фрагмент коду такого парсингу наведено на рисунку 3.2.

Рисунок 3.2 – Фрагмент коду парсингу даних зі сторінок eBay (Python)

Джерело: розробка автора

Отримані сирі дані підлягають ретельному очищенню та нормалізації перед побудовою моделі. На етапі попередньої обробки виконуються такі кроки:

Перевірка на пропущені та хибні значення. Виявляються та обробляються пусті поля чи аномальні значення (наприклад, якщо рейтинг невідомий або відгуки не відображені). Записи з браком критичної інформації можуть бути виключені або доповнені за правилами обробки.

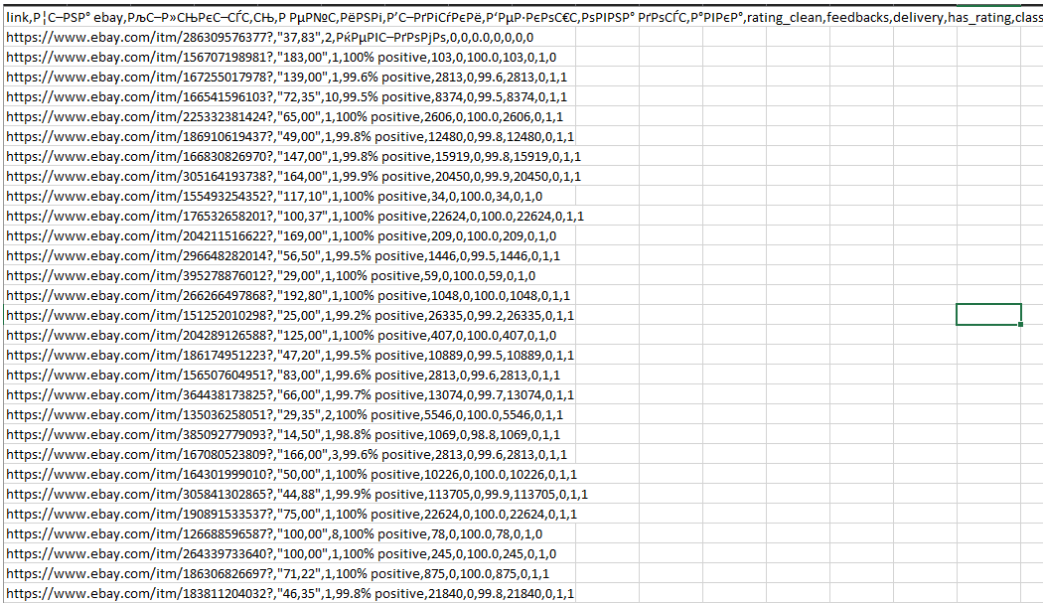
Типове перетворення форматів. Числові значення, що подані у текстовому форматі, конвертуються у числові змінні. Наприклад, рядок "18,70" у полі ціни перетворюється в тип float 18.70, а відсоткові рейтинги на кшталт "100% positive" – у число 100.0. Булеві поля («так/ні») замінюються на 1/0.

Нормалізація ознак. Для уніфікації одиниць вимірювання всі числові параметри переводяться у єдину шкалу. Наприклад, рейтинг продавця масштабується до відрізку [0, 1], вартість доставки може бути стандартізована (z-score) або приведена до відсотків від максимальної величини. Такі операції забезпечують правильну роботу алгоритмів машинного навчання. Чисті, якісні дані є основою для стабільної роботи моделі.

Після трансформації формується остаточний набір ознак для тренування моделі. Для кожного постачальника до набору даних входять відфільтровані числові атрибути (раніше згадані рейтинг, число відгуків, параметри доставки тощо), які складають вектор ознак X. Відповідні мітки класів Y («надійний»/«ризикований») задаються виходячи зі встановлених правил чи історичної інформації про постачальників (наприклад, високий рейтинг + багато позитивних відгуків означає «надійний»). Підготовлені дані поділяються на тренувальний і тестовий набори. Як алгоритм класифікації використовується Random Forest – ансамбль із багатьох дерев рішень. Цей метод широко застосовується у задачах класифікації завдяки високій точності і стійкості до «шумових» даних: він комбінує прогнози численних дерев і мінімізує ризик перенавчання. Random Forest дозволяє врахувати різнорідні характеристики постачальника і виявити складні залежності між ними та загальною надійністю. Перед застосуванням моделі даних передбачено випадкове перемішування та (за необхідності) крос-валідацію для оцінки узагальнюючої здатності.

Результатом роботи модуля класифікації є маркування кожного запису вхідних даних відповідним класом. Файл classified_output.csv містить початкові поля разом із підготовленими числовими ознаками (rating_clean, feedbacks, delivery) та присвоєним класом class (наприклад, 1 – «надійний», 0 – «ризикований»). Наприклад, структура рядка у фінальному файлі може бути такою:

https://www.ebay.com/itm/167255017978?,"139,00",1,99.6% positive,2813,0,99.6,2813,0,1,1 – де останнє число 1 означає «надійний». Збереження здійснюється у форматі CSV з латинськими заголовками колонок (як показано вище). На рисунку 3.3 наведено фрагмент результуючого файлу після класифікації, де за кожним постачальником закріплено отриманий клас.



link,P¦C–PSP° ebay,PљC–P»CЊPєC–CЃC,CЊ,P PµPNєC,PёPSPi,P'C–PrPiCґPєPё,P'PµP·PєPsC€C,PsPIPSP° PrPsCЃC,P°PIPєP°,rating_clean,feedbacks,delivery,has_rating,class
https://www.ebay.com/itm/286309576377?,"37,83",2,PќPµPIC–PrPsPjPs,0,0,0.0,0,0,0,0
https://www.ebay.com/itm/156707198981?,"183,00",1,100% positive,103,0,100.0,103,0,1,0
https://www.ebay.com/itm/167255017978?,"139,00",1,99.6% positive,2813,0,99.6,2813,0,1,1
https://www.ebay.com/itm/166541596103?,"72,35",10,99.5% positive,8374,0,99.5,8374,0,1,1
https://www.ebay.com/itm/225323381424?,"65,00",1,100% positive,2606,0,100.0,2606,0,1,1
https://www.ebay.com/itm/186910619437?,"49,00",1,99.8% positive,12480,0,99.8,12480,0,1,1
https://www.ebay.com/itm/166830826970?,"147,00",1,99.8% positive,15919,0,99.8,15919,0,1,1
https://www.ebay.com/itm/305164193738?,"164,00",1,99.9% positive,20450,0,99.9,20450,0,1,1
https://www.ebay.com/itm/155493254352?,"117,10",1,100% positive,34,0,100.0,34,0,1,0
https://www.ebay.com/itm/176532658201?,"100,37",1,100% positive,22624,0,100.0,22624,0,1,1
https://www.ebay.com/itm/204211516622?,"169,00",1,100% positive,209,0,100.0,209,0,1,0
https://www.ebay.com/itm/296648282014?,"56,50",1,99.5% positive,1446,0,99.5,1446,0,1,1
https://www.ebay.com/itm/395278876012?,"29,00",1,100% positive,59,0,100.0,59,0,1,0
https://www.ebay.com/itm/266266497868?,"192,80",1,100% positive,1048,0,100.0,1048,0,1,1
https://www.ebay.com/itm/151252010298?,"25,00",1,99.2% positive,26335,0,99.2,26335,0,1,1
https://www.ebay.com/itm/204289126588?,"125,00",1,100% positive,407,0,100.0,407,0,1,0
https://www.ebay.com/itm/186174951223?,"47,20",1,99.5% positive,10889,0,99.5,10889,0,1,1
https://www.ebay.com/itm/156507604951?,"83,00",1,99.6% positive,2813,0,99.6,2813,0,1,1
https://www.ebay.com/itm/364438173825?,"66,00",1,99.7% positive,13074,0,99.7,13074,0,1,1
https://www.ebay.com/itm/135036258051?,"29,35",2,100% positive,5546,0,100.0,5546,0,1,1
https://www.ebay.com/itm/385092779093?,"14,50",1,98.8% positive,1069,0,98.8,1069,0,1,1
https://www.ebay.com/itm/167080523809?,"166,00",3,99.6% positive,2813,0,99.6,2813,0,1,1
https://www.ebay.com/itm/164301999010?,"50,00",1,100% positive,10226,0,100.0,10226,0,1,1
https://www.ebay.com/itm/305841302865?,"44,88",1,99.9% positive,113705,0,99.9,113705,0,1,1
https://www.ebay.com/itm/190891533537?,"75,00",1,100% positive,22624,0,100.0,22624,0,1,1
https://www.ebay.com/itm/126688596587?,"100,00",8,100% positive,78,0,100.0,78,0,1,0
https://www.ebay.com/itm/264339733640?,"100,00",1,100% positive,245,0,100.0,245,0,1,0
https://www.ebay.com/itm/186306826697?,"71,22",1,100% positive,875,0,100.0,875,0,1,1
https://www.ebay.com/itm/183811204032?,"46,35",1,99.8% positive,21840,0,99.8,21840,0,1,1

Рисунок 3.3 – Фрагмент файлу classified_output.csv з результатами класифікації постачальників

Джерело: розробка автора

3.2 Розроблення користувацького інтерфейсу. Елементи та структура.

У рамках реалізації програмного засобу для збору, обробки та класифікації даних про продавців на платформі eBay було використано середовище Anaconda Navigator, яке забезпечило зручну організацію робочого простору, а також інтерактивний інструмент Jupyter Notebook, що виконує роль інтерфейсу користувача та середовища взаємодії з побудованою системою.

Anaconda Navigator є графічною оболонкою для управління інструментами наукових обчислень, зокрема засобами машинного навчання, і дозволяє створювати ізольовані віртуальні середовища, встановлювати необхідні бібліотеки та запускати зовнішні інструменти без використання командного рядка. Його застосування забезпечило швидке розгортання середовища розробки, уникнення конфліктів між бібліотеками та стабільну роботу над проєктом у межах окремого середовища Python. Усі ключові компоненти — від бібліотек для збору та обробки даних (Selenium, pandas) до моделей машинного навчання (scikit-learn) — були встановлені та керовані централізовано через інтерфейс Navigator, що значно спростило процес налаштування.

Із цього середовища було запущено Jupyter Notebook — основний інструмент для реалізації логіки системи, її тестування та демонстрації результатів класифікації продавців.

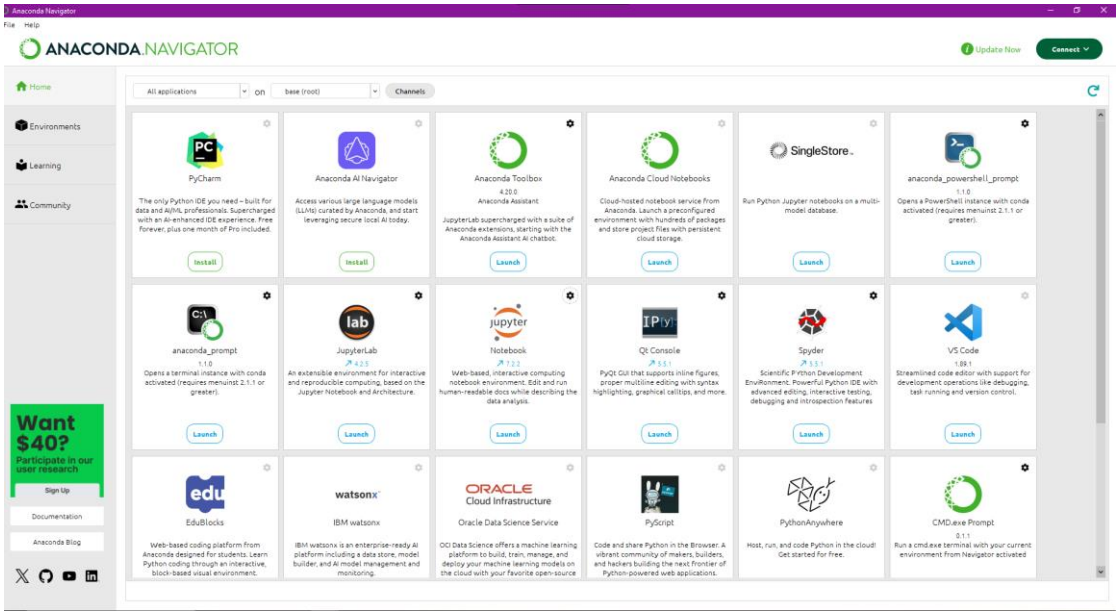


Рисунок 3.4 – Інтерфейс Navigator

Джерело: інтерфейс Navigator

Після запуску та ініціалізації ноутбука наступним кроком є завантаження початкового набору даних – CSV-файлу зі списком посилань на eBay-продавців. У

спеціальному блоці коду виконується команда pd.read_csv, яка завантажує файл і відображає перші рядки у вигляді таблиці

```
import pandas as pd

# Шлях до CSV
df = pd.read_csv("D:/Amazon drop diplom - Лист1.csv")

# Перевіримо перші рядки
df.head()
```

	link	Ціна ebay	Кількість
0	https://www.ebay.com/itm/335633117175	107,04	10
1	https://www.ebay.com/itm/335713174597	178,10	3
2	https://www.ebay.com/itm/296754140904?	18,55	10
3	https://www.ebay.com/itm/335785968631	150,00	1
4	https://www.ebay.com/itm/393666137340	75,96	1

Рисунок 3.5 – блок завантаження CSV у Jupyter

Джерело: розробка автора

Користувачу достатньо вказати шлях до файлу – і система одразу демонструє зміст таблиці, що дозволяє перевірити коректність даних перед подальшою обробкою. Цей блок формує DataFrame з початковим набором даних та відображає його як вихід, забезпечуючи зручність попереднього перегляду вхідних даних.

Наступний блок відповідає за парсинг інформації за посиланнями. У ньому використовується запит до eBay та бібліотеки веб-скрапінгу, щоб отримати деталі про продавців (рейтинг, кількість відгуків, умови доставки тощо)

```
        # ⭐ Рейтинг
        try:
            rating_element = driver.find_element(By.CSS_SELECTOR, "span.ux-textspans.ux-textspans--PSEUDOLINK")
            rating_text = rating_element.text.strip()
        except:
            rating_text = "Невідомо"

        # 🚚 Доставка
        try:
            shipping_element = driver.find_element(By.XPATH, "//div[@data-testid='ux-labels-values__values-content']//span[@class='ux-textspans ux-textsp
            shipping_text = shipping_element.text.strip().lower()
            delivery = 1 if "free" in shipping_text else 0
        except Exception as e:
            print(f"❌ Помилка при визначенні доставки: {e}")
            delivery = 0

        print(f"⭐ Рейтинг продавця: {rating_text}")
        print(f"💬 Кількість відгуків: {feedbacks}")
        print(f"🚚 Безкоштовна доставка: {'Так' if delivery else 'Ні'}")

        return rating_text, feedbacks, delivery

    except Exception as e:
        print(f"🔴 Загальна помилка для {url}: {e}")
        return None, 0, 0

    finally:
        driver.quit()
```

```
]: url = "https://www.ebay.com/itm/335633117175"
rating, feedbacks, delivery = parse_with_selenium(url)

print("⭐ Рейтинг продавця:", rating)
print("💬 Кількість відгуків:", feedbacks)
print("🚚 Безкоштовна доставка:", "Так" if delivery else "Ні")
```

```
⭐ Рейтинг продавця: 99.4% positive
💬 Кількість відгуків: 1311
🚚 Безкоштовна доставка: Так
```

Рисунок 3.6 – Блок парсингу посилань та виведення сирих даних

Джерело: розробка автора

Приклад результату цього етапу: у консолі відображено рядки з інформацією по кожному продавцю – наприклад, «Рейтинг: 99,4% positive, Відгуки: 1311, Доставка: Так» (де іконками позначено категорії даних). Кожен рядок відповідає окремому товару з eBay і містить сирі значення атрибутів. Метою цього блоку є автоматичне збирання “сирих” даних з інтернету для кожного посилання, щоб підготувати дані для подальшої обробки. Користувач у цьому етапі фактично бачить, які дані система отримала з сайту eBay і може переконатися в тому, що скрапінг відбувається правильно.

Основний функціонал системи реалізовано в блоці класифікації продавців. Тут виконується вже машинне навчання: передбачення належності продавця до певного класу на основі отриманих ознак. Наприклад, може використовуватись навчена модель з бібліотеки scikit-learn або інший алгоритм класифікації.

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Вхідні змінні та мітки
X = df_final[['rating_clean', 'feedbacks', 'delivery']]
y = df_final['label']

# Розділення на train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Навчання RandomForest
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Перевірка точності
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"🎯 Точність моделі: {accuracy * 100:.2f}%")
```

```
🎯 Точність моделі: 68.75%
```

Рисунок 3.7 – Навчання моделі

Джерело: розробка автора

На завершальному етапі результати вивантажуються у зовнішній файл для зручної роботи користувача. Зокрема, створюється файл classified_output.csv, в якому містяться всі ключові колонки: посилання на продавця, початкові атрибути, очищені значення і прогнозований клас.

A	B	C	D	E	F	G	H	I	J	K
link	Ціна ebay	Кількість	Рейтинг	Відгуки	Безкоштовна дс	rating_clean	feedbacks	delivery	has_rating	class
https://www.ebay	37,83	2	Невідомо	0	0	0.0	0	0	0	0
https://www.ebay	183	1	100% positive	103	0	100.0	103	0	1	0
https://www.ebay	139	1	99.6% positive	2813	0	99.6	2813	0	1	1
https://www.ebay	72,35	10	99.5% positive	8374	0	99.5	8374	0	1	1
https://www.ebay	65	1	100% positive	2606	0	100.0	2606	0	1	1
https://www.ebay	49	1	99.8% positive	12480	0	99.8	12480	0	1	1
https://www.ebay	147	1	99.8% positive	15919	0	99.8	15919	0	1	1
https://www.ebay	164	1	99.9% positive	20450	0	99.9	20450	0	1	1
https://www.ebay	117,1	1	100% positive	34	0	100.0	34	0	1	0
https://www.ebay	100,37	1	100% positive	22624	0	100.0	22624	0	1	1
https://www.ebay	169	1	100% positive	209	0	100.0	209	0	1	0
https://www.ebay	56,5	1	99.5% positive	1446	0	99.5	1446	0	1	1
https://www.ebay	29	1	100% positive	59	0	100.0	59	0	1	0
https://www.ebay	192,8	1	100% positive	1048	0	100.0	1048	0	1	1
https://www.ebay	25	1	99.2% positive	26335	0	99.2	26335	0	1	1
https://www.ebay	125	1	100% positive	407	0	100.0	407	0	1	0
https://www.ebay	47,2	1	99.5% positive	10889	0	99.5	10889	0	1	1
https://www.ebay	83	1	99.6% positive	2813	0	99.6	2813	0	1	1
https://www.ebay	66	1	99.7% positive	13074	0	99.7	13074	0	1	1
https://www.ebay	29,35	2	100% positive	5546	0	100.0	5546	0	1	1
https://www.ebay	14,5	1	98.8% positive	1069	0	98.8	1069	0	1	1
https://www.ebay	166	3	99.6% positive	2813	0	99.6	2813	0	1	1
https://www.ebay	50	1	100% positive	10226	0	100.0	10226	0	1	1
https://www.ebay	44,88	1	99.9% positive	113705	0	99.9	113705	0	1	1
https://www.ebay	75	1	100% positive	22624	0	100.0	22624	0	1	1
https://www.ebay	100	8	100% positive	78	0	100.0	78	0	1	0
https://www.ebay	100	1	100% positive	245	0	100.0	245	0	1	0
https://www.ebay	71,22	1	100% positive	875	0	100.0	875	0	1	1
https://www.ebay	46,35	1	99.8% positive	21840	0	99.8	21840	0	1	1
https://www.ebay	26,35	1	100% positive	226	0	100.0	226	0	1	0
https://www.ebay	15	2	99.3% positive	43660	0	99.3	43660	0	1	1
https://www.ebay	66	1	98.7% positive	767	0	98.7	767	0	1	1
https://www.ebay	52,16	4	100% positive	1256	0	100.0	1256	0	1	1
https://www.ebay	95,37	2	100% positive	22624	0	100.0	22624	0	1	1
https://www.ebay	100	8	100% positive	78	0	100.0	78	0	1	0
https://www.ebay	34,05	1	99.8% positive	21840	0	99.8	21840	0	1	1
https://www.ebay	108,06	10	98.4% positive	27956	0	98.4	27956	0	1	1

Рисунок 3.8 – Відкритий CSV у Google Таблицях

Джерело: розробка автора

Користувач бачить знайомий інтерфейс таблиці з усіма даними у зручному для аналізу вигляді. Такий фінальний крок дає змогу експортувати результати, поділитися ними або провести подальший аналіз поза середовищем Jupyter.

Таким чином, інтерфейс системи побудовано на принципі розбиття процесу на логічні блоки, що відповідають окремим стадіям обробки даних (завантаження, парсинг, обробка, класифікація, вивантаження). Кожен блок надає чіткий візуальний зворотний зв’язок: користувач одразу бачить результати виконання коду (таблиці або текстові повідомлення). Така організація інтерфейсу робить його надзвичайно зручним і інтуїтивним – як і очікується від графічного інтерфейсу з відомих GUI-принципів. Використання знайомих елементів (клітинок коду і результатів, кнопок виконання тощо) дозволяє виконувати складний аналітичний процес з мінімальними знаннями про внутрішню реалізацію. Розбиття на обчислювальні блоки допомагає уникнути помилок і зменшує потребу у послідовному запуску всього сценарію: користувач може протестувати і перезапустити окремі частини без зупинки всього алгоритму. Як наслідок, такий організований та інтуїтивно зрозумілий інтерфейс значно підвищує зручність роботи з системою і сприяє швидкому отриманню результатів.

3.3 Проектування забезпечувальних підсистем СШІ. Реалізація системи.

3.3.1 Інформаційне забезпечення.

Інформаційне забезпечення розробленої інтелектуальної системи — це сукупність даних та механізмів їхнього збору, необхідних для формування навчальної вибірки та роботи моделі. Ефективність будь-якої моделі машинного навчання, включаючи систему класифікації продавців eBay, безпосередньо залежить від якості, повноти та структури використовуваної бази знань. У рамках

цієї системи базою знань слугує зібрана з офіційної веб-платформи eBay інформація про продавців та їх товари.

Для збору вказаної інформації джерелом даних обрано офіційну платформу eBay. Конкретно, як вихідні дані використовується перелік URL-адрес товарних лотів або профілів продавців на сайті eBay. Для автоматизації процесу збору даних застосовано метод веб-скрапінгу із використанням бібліотеки Selenium WebDriver. Це дозволяє керувати браузером програмно: переходити за заданими посиланнями, виконувати дії на сторінці (наприклад, прокручування) та витягувати необхідні елементи HTML-коду. Завдяки такому підходу система може надійно отримувати динамічно завантажені дані про рейтинги, відгуки та умови доставки.

Отримані з платформи eBay дані зберігаються у вигляді таблиць у форматі CSV. Така таблиця організована за класичною табличною структурою: стовпці відповідають ознакам продавця або лоту, а рядки — окремим записам (посиланням на лоти чи продавцям)

Окрім сирих полів, у цій таблиці можуть бути присутніми й попередньо оброблені ознаки, наприклад числова версія рейтингу (стовпець rating_clean), кількість відгуків (feedbacks), булевий показник наявності безкоштовної доставки (delivery), а також цільова змінна класу (class), яка використовується при навчанні моделі.

Дані проходять шлях від початкових посилань до фінальних ознак так: спочатку формується перелік URL-адрес товарів або профілів продавців на eBay. За кожним таким посиланням Selenium автоматизовано переходить на відповідну сторінку і витягує потрібні атрибути (наприклад, текстове значення рейтингу, числове значення ціни, статус доставки тощо). Зібрані сирі дані потім очищуються та перетворюються у зручний для моделі формат: наприклад, відсоток позитивних відгуків перетворюється у десяткове число, а інформація про безкоштовну доставку кодується як булева ознака. Таким чином формується вектор ознак для кожного запису. Ці вектори ознак разом з цільовою змінною (міткою класу продавця) використовуються для навчання та роботи класифікаційного алгоритму.

Інформаційне забезпечення системи виконує ключову роль у її функціонуванні. Саме завдяки зібраним даним система може навчитися відрізняти різні класи продавців і робити точну класифікацію. Якість, повнота та своєчасне оновлення інформації безпосередньо впливають на ефективність роботи моделі. Отримана і структурована інформація забезпечує основу для аналізу та прийняття рішень: без належного інформаційного забезпечення система не зможе адекватно оцінити контекст продавця чи його лотів і, як наслідок, видавати коректні класифікаційні рішення.

3.3.2 Програмне забезпечення.

У процесі розробки інтелектуальної системи збору та класифікації даних про продавців eBay на основі методів машинного навчання було обґрунтовано обрано мову програмування Python (версії 3.10) як основну платформу реалізації програмного забезпечення. Python — це високорівнева, інтерпретована, мультипарадигмальна мова програмування з простим та читабельним синтаксисом, що поєднує потужний набір можливостей і широку підтримку наукової спільноти. Використання Python надає доступ до численних готових бібліотек, що охоплюють увесь цикл роботи системи: від збору даних до побудови і навчання моделей машинного навчання. У цьому проєкті Python дозволив реалізувати повний цикл: автоматизований збір даних про продавців з веб-сторінок eBay, їхню попередню обробку, побудову й навчання моделі класифікації, а також збереження отриманих результатів. Такий підхід уникає необхідності застосування різних мов програмування або окремих інструментів для кожного етапу, що значно спрощує підтримку й масштабування системи.

Розробка проєкту виконувалась під операційною системою Windows 10, яка забезпечує стабільну роботу необхідних програмних компонентів та драйверів. В якості оболонки середовища розробки було використано Anaconda Navigator —

інтегрований інструмент для управління віртуальними середовищами та бібліотеками Python. Основним середовищем написання й виконання коду обрано Jupyter Notebook, який дає змогу розбивати програму на логічно відокремлені клітинки із послідовним виконанням. Такий підхід забезпечує інтерактивне документування процесу: спочатку виконується блок коду для збору даних, потім — блок для їхньої обробки, далі — моделювання і, нарешті, збереження результатів. Для автоматизованого збору даних застосовується браузер Google Chrome у поєднанні з бібліотекою Selenium WebDriver, що забезпечує програмне керування браузером на платформі Windows 10.

У розробці системи були задіяні такі основні бібліотеки:

- pandas – для завантаження, обробки та аналізу табличних даних про продавців.
- selenium – для автоматизованого збору даних з веб-сторінок eBay.
- scikit-learn – для побудови, налаштування та навчання моделей класифікації.
- joblib – для серіалізації та збереження навчених моделей і оброблених даних.

Бібліотека pandas використовується для ефективної роботи з табличними даними у форматі DataFrame. У проєкті вона відповідає за зчитування зібраних даних (наприклад, CSV-файлів зі збереженою інформацією про продавців eBay) та їх попередню обробку. За допомогою pandas проводиться очищення таблиць, вибір необхідних стовпців та перетворення форматів даних, що дозволяє підготувати зручний для аналізу вхідний набір даних. Також бібліотека забезпечує експорт оброблених результатів (наприклад, попередньо згенерованих датафреймів) у файли формату CSV для подальшого аналізу або візуалізації.

Бібліотека selenium застосовується для автоматизованого збору даних з веб-сторінок eBay. Вона забезпечує управління браузером Google Chrome (за допомогою ChromeDriver) у середовищі Windows 10 та дозволяє керувати веб-інтерфейсом програмно. Скрипти на основі Selenium послідовно відкривають

сторінки продавців на eBay, витягають необхідну інформацію (таку як рейтинг продавця, кількість продажів, відгуки тощо) та зберігають її у табличному форматі. Таким чином бібліотека виконує функцію інструменту для збору первинних даних без необхідності ручного втручання.

Для етапу моделювання використовується бібліотека scikit-learn, що надає широкий набір алгоритмів машинного навчання для класифікації. Scikit-learn забезпечує функціонал для підготовки вибірок, розділення даних на тренувальну та тестову множини, налаштування та навчання обраної моделі класифікації. У цьому проєкті за допомогою scikit-learn реалізується процес навчання моделі, що класифікує продавців eBay за вибраними ознаками (наприклад, за кількістю позитивних та негативних відгуків, рівнем активності тощо). Бібліотека також дозволяє оцінювати якість навченої моделі за допомогою різних метрик (точності, повноти, F1 тощо) та виконувати прогнозування на нових даних.

Бібліотека joblib використовується для серіалізації та збереження результатів роботи моделі. Після навчання класифікаційної моделі отриманий об’єкт зберігається у файл за допомогою функції joblib.dump, що забезпечує швидке завантаження моделі в майбутньому без повторного навчання. Таким чином joblib дозволяє зберігати не тільки навчену модель, але й інші об’єкти проєкту (наприклад, попередньо оброблені дані чи метадані), забезпечуючи відновлення стану системи.

Поєднання мови Python 3.10, інтерактивного середовища Jupyter Notebook, оболонки Anaconda Navigator та перелічених бібліотек (pandas, selenium, scikit-learn, joblib) забезпечує комплексну реалізацію всіх етапів проєкту — від збору й обробки даних до побудови, навчання та збереження моделі класифікації без потреби в зовнішніх інструментах.

3.3.3 Технічне забезпечення.

Розробка, навчання та тестування інтелектуальної системи збору та класифікації даних про продавців eBay виконувалась на персональному комп’ютері, який має достатню обчислювальну потужність для обробки масивів текстових та числових даних, а також реалізації моделей машинного навчання середнього рівня складності. Обрана конфігурація системи забезпечила стабільну роботу при розробці, тестуванні та запуску основних компонентів програмного комплексу, включно з парсингом даних через Selenium, обробкою таблиць, навчанням моделі класифікації та збереженням результатів у зовнішні файли.

Таблиця 3.2 – Апаратне забезпечення робочого середовища

Компонент	Характеристика
Процесор (CPU)	AMD Ryzen 5 3600, 6 ядер, 12 потоків, 3.59 GHz
Відеокарта (GPU)	NVIDIA GeForce GTX 1660 SUPER, 6 ГБ GDDR6
Оперативна пам’ять (RAM)	16 ГБ DDR4
Накопичувачі	SSD Kingston SA2000M8250G + HDD WDC WD10EZEX
Операційна система	Windows 10 Корпоративна, 64-біт, версія 22H2
Середовище виконання	Anaconda (Python 3.10), Jupyter Notebook

Джерело: розробка автора

Уся робота над проєктом здійснювалась у локальному середовищі розробки на базі Anaconda Navigator, з запуском коду у Jupyter Notebook. Така конфігурація забезпечила зручне виконання коду в клітинках, а також можливість

інтерактивного тестування окремих фрагментів, документування етапів проєкту, побудови графіків та формування вихідних таблиць.

Для автоматизованого збору даних із платформи eBay активно використовувалась бібліотека Selenium у поєднанні з веббраузером Google Chrome, який взаємодіяв із системою через ChromeDriver. Парсинг даних здійснювався в безголовому режимі (headless), що дозволяло зменшити навантаження на систему та уникнути зайвих візуальних ресурсів при зборі великої кількості посилань.

Навчання моделі класифікації здійснювалось на центральному процесорі (CPU), хоча наявна дискретна відеокарта NVIDIA GeForce GTX 1660 SUPER потенційно дозволяє здійснювати обчислення з використанням GPU (наприклад, при перенесенні моделі до PyTorch чи TensorFlow). У межах даного проєкту графічний процесор не задіювався, оскільки модель мала невелику кількість параметрів, а обсяг вхідних даних був обмежений (до кількох сотень прикладів на запуск).

Повний цикл обробки даних — від парсингу до збереження класифікованого файлу у форматі CSV — зазвичай займав не більше 2–3 хвилин при обробці десятків посилань. Обчислювальних потужностей процесора Ryzen 5 3600 та 16 ГБ оперативної пам’яті було достатньо для роботи в середовищі Jupyter з одночасним відкриттям браузера та обробкою результатів у Google Таблицях або Excel.

3.3.4 Організаційно- економічне забезпечення

Дипломний проєкт є навчальним за своєю природою, тому його реалізація не передбачала залучення зовнішнього фінансування. Усі необхідні для роботи ресурси належать автору роботи. Проєкт реалізувався автором самостійно, без залучення інших виконавців чи установ.

Для розробки системи використовувалися власні технічні та програмні ресурси автора: персональний комп’ютер, встановлена операційна система та необхідне програмне забезпечення. Зокрема, для обробки даних та розробки алгоритмів машинного навчання використовувалися мова програмування Python та відповідні бібліотеки (наприклад, Pandas, NumPy, Scikit-learn). Також застосовувалися середовища Anaconda та Jupyter Notebook для організації робочого процесу, а для доступу до необхідних онлайн-даних — веб-браузер Google Chrome. Всі зазначені програмні засоби є безкоштовними для використання.

Тривалість розробки проєкту склала приблизно два місяці. У цей період автор виконував весь комплекс робіт: дослідження теоретичних основ, збір та попередню обробку даних, розробку та тренування моделей машинного навчання, а також тестування системи й оформлення необхідної документації.

Організаційна підтримка проєкту полягала в консультаціях з науковим керівником, який надавав методичні рекомендації та наглядав за етапами роботи. Крім того, автор мав доступ до навчальних матеріалів університету та онлайн-ресурсів, що сприяло поглибленню знань з теми і успішному виконанню дипломного проєкту.

Висновки до розділу 3

У цьому розділі було здійснено повну реалізацію інтелектуальної системи збору та класифікації даних про продавців платформи eBay. Система розроблена як інтерактивний програмний засіб на базі Python із використанням середовища Jupyter Notebook, що дозволило реалізувати гнучкий та зручний для користувача інтерфейс.

Було описано архітектуру системи, яка включає модуль автоматизованого збору даних із сайту eBay за допомогою Selenium, блок попередньої обробки даних та класифікаційний модуль, що базується на моделі машинного навчання. У якості моделі було обрано алгоритм Random Forest, який продемонстрував стабільну роботу з невеликим обсягом даних і забезпечив достатню точність при класифікації продавців на надійних та потенційно ризикованих.

Інтерфейс реалізовано у вигляді послідовних клітинок у Jupyter Notebook, що дозволяє виконувати всі етапи обробки та класифікації покроково — від завантаження списку посилань до вивантаження результатів у форматі CSV. Це робить систему зручною в експлуатації як для дослідника, так і для користувача без глибоких технічних знань.

Інформаційне забезпечення системи базується виключно на даних, зібраних із відкритих джерел — сторінок товарів на платформі eBay. Основними ознаками, що використовуються для класифікації, є рейтинг продавця, кількість відгуків, наявність безкоштовної доставки та наявність самого рейтингу.

Усі етапи проєкту були реалізовані з використанням вільного програмного забезпечення: мови програмування Python, бібліотек Pandas, Selenium, Scikit-learn, а також оболонки Anaconda та середовища Jupyter Notebook. Це дозволило уникнути додаткових фінансових витрат на ліцензовані продукти.

Розробка системи проводилась на власному обладнанні — настільному комп’ютері з процесором AMD Ryzen 5 3600, 16 ГБ оперативної пам’яті та відеокартою NVIDIA GeForce GTX 1660 SUPER. Такої конфігурації було достатньо для повного циклу обробки даних і навчання моделі.

Проєкт реалізовано без залучення зовнішнього фінансування. Усі роботи — від дослідження й програмування до тестування та документування — були виконані автором самостійно протягом двох місяців. Організаційна підтримка полягала в методичних консультаціях наукового керівника та використанні відкритих навчальних ресурсів.

ВИСНОВКИ

У межах виконання дипломної роботи було реалізовано повнофункціональний програмний засіб, призначений для збору, обробки та класифікації інформації про продавців на платформі eBay із використанням методів машинного навчання. Система розроблена з метою автоматизації процесу оцінки надійності продавців на основі відкритих даних, що дозволяє спростити процес прийняття рішень у сфері електронної комерції, зокрема дропшипінгу.

У процесі роботи було виконано такі основні завдання:

- Проведено огляд актуальних підходів до збору інформації з відкритих джерел та класифікації об'єктів за допомогою алгоритмів машинного навчання
- Запроєктовано архітектуру інтелектуальної системи, що складається з модулів збору даних, попередньої обробки, побудови ознак і класифікації
- Розроблено парсер на базі бібліотеки Selenium, який автоматично витягує ключові атрибути продавців із сайту eBay (рейтинг, кількість відгуків, інформацію про доставку)
- Створено систему підготовки даних для навчання моделі, включаючи етапи очищення, нормалізації та перетворення у формат, придатний для моделювання
- Проведено навчання та тестування моделі класифікації (Random Forest), яка здатна диференціювати надійних і потенційно ризикованих продавців на основі заданих ознак
- Реалізовано інтерфейс у вигляді Jupyter Notebook, який дозволяє послідовно запускати усі етапи обробки, отримувати проміжні результати та зберігати фінальні дані у зручному форматі CSV.

- Проведено експериментальну перевірку працездатності системи на реальних даних eBay, що підтвердило коректність її роботи та потенційну практичну цінність

Розробка була виконана із використанням відкритого програмного забезпечення (Python, pandas, scikit-learn, selenium, joblib, Jupyter Notebook), що дозволяє розповсюджувати систему без додаткових витрат. Технічні ресурси, задіяні у проєкті, забезпечили достатню продуктивність для реалізації повного циклу обробки та навчання.

Таким чином, поставлені цілі дипломної роботи досягнуті. Система, створена в ході дослідження, може бути використана як основа для подальшого розвитку інструментів аналізу продавців на маркетплейсах, а також інтегрована у процеси дропшипінгу та закупівель для автоматичної фільтрації ненадійних постачальників.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дропшипінг: Що це та як працює (Повний гайд). – Режим доступу: horoshop.ua/blog/what-is-dropshipping-and-how-does-it-work (дата звернення: 11.05.2025).

2. Market Data Forecast. Dropshipping Market – Global Industry Analysis (2024–2029) // Викладено в огляді G2 Research: Why Dropshipping Holds a Promising Future..., 2024. – Режим доступу: research.g2.com/insights/drop-shipping-for-e-commerce-entrepreneurs (дата звернення: 11.05.2025).

3. Amazon обмежує Dropshipping? Як дропшиппери можуть знайти вихід? – CJ Dropshipping Blog, 16 березня 2021. – Режим доступу: cjdropship.com/uk/amazon-обмежує-продавців-від-дропшиппінгу (дата звернення: 11.05.2025).

4. SellerAssistant App. How to Use an Amazon Prep Center to Enhance Your Business (Blog). – Режим доступу: sellerassistant.app/blog/amazon-fba-prep-center (дата звернення: 11.05.2025).

5. Dropship.io. Dropshipping Statistics & Trends To Know In 2024 (Rebecca G, 18.02.2024). – Режим доступу: dropship.io/blog/dropshipping-statistics (дата звернення: 11.05.2025).

6. VidaXL DropshippingXL. Короткий посібник із дропшипінгу на Amazon (укр. версія). – Режим доступу: dropshippingxl.com/uk/blog-a-brief-guide-to-amazon-dropshipping (дата звернення: 11.05.2025).

7. Shopify. AI Tools for Dropshipping (by Shopify Staff, updated Apr 29, 2025). – Режим доступу: shopify.com/blog/ai-dropshipping (дата звернення: 11.05.2025).

8. AppScenic. How AI is Transforming the Dropshipping Industry (12 Aug 2024). – Режим доступу: appscenic.com/blog/how-ai-is-transforming-the-dropshipping-industry (дата звернення: 11.05.2025).

9. Shopify. AI Tools for Dropshipping (by Shopify Staff, updated Apr 29, 2025). – Режим доступу: https://www.shopify.com/blog/ai-dropshipping (дата звернення: 11.05.2025).

10. Shopify. What Is AI Demand Forecasting? Use Cases For Ecommerce. – Режим доступу: https://www.shopify.com/blog/ai-demand-forecasting (дата звернення: 11.05.2025).

11. Jacques van der Wilt. AI-Powered Dynamic Pricing Strategies for eCommerce. – DataFeedWatch Blog. – Режим доступу: https://www.datafeedwatch.com/blog/ai-dynamic-pricing (дата звернення: 11.05.2025).

12. ResearchGate. Demand Forecasting in E-Commerce Using AI & ML. – Режим доступу: https://www.researchgate.net/publication/388779767_Demand_Forecasting_in_E-Commerce_Using_AI_ML (дата звернення: 11.05.2025).

13. ChromeStats. GRABLEY – Product Search Tools (Amazon, eBay, others...). – Режим доступу: https://chrome-stats.com/d/hppdgjpcbnbfapnailmeiibngpolplao (дата звернення: 11.05.2025).

14. David Cullum. What is Helium 10, and why does every Amazon FBA seller use it? – Global E-commerce Experts. – Режим доступу: https://globale-commerceexperts.com/what-is-helium-10-and-why-does-every-amazon-fba-seller-use-it (дата звернення: 11.05.2025).

15. Google Play. Keepa – Amazon Price Tracker (app description). – Режим доступу: https://play.google.com/store/apps/details?id=com.keepa.mobile (дата звернення: 11.05.2025).

16. RevenueGeeks. Helium 10 vs Keepa: 2025 Guide for Amazon Sellers. – Режим доступу: https://revenuegeeks.com/helium10-vs-keepa (дата звернення: 11.05.2025).

17. GroupBuyExpert. OAGenius: Best Online Arbitrage And Dropshipping Software. – Режим доступу: https://groupbuyexpert.com/oagenius-best-online-arbitrage-and-dropshipping-software (дата звернення: 11.05.2025).

18. Similarweb. autods.com (site info snippet). – Режим доступу: https://www.similarweb.com/website/autods.com (дата звернення: 11.05.2025).

19. AliDrop Blog. DSers AliExpress Dropshipping: Features, Benefits, and How to Get Started. – Режим доступу: https://www.alidrop.co/blogs/dsers-aliexpress-dropshipping-features-benefits-and-how-to-get-started (дата звернення: 11.05.2025).

20. InventorySource Blog. Creating a Supplier Scorecard for Your Dropshipping Partners, Dec 26, 2023. – Режим доступу: https://www.inventorysource.com/dropshipping-supplier-scorecard (дата звернення: 11.05.2025).

21. AutoDS Blog. What Is Dropshipping? How It Works, Pros & Cons, Oct 5, 2023. – Режим доступу: https://www.autods.com/blog/dropshipping-guides/what-is-dropshipping (дата звернення: 11.05.2025).

22. Amazon Seller Central. Dropshipping Policy – Режим доступу: https://sellercentral.amazon.com/gp/help/external/G6DQ4P3WUNP2F3J3 (дата звернення: 11.05.2025).

23. Grabley Product Sourcing Extension. Features Overview. – Режим доступу: https://chrome.google.com/webstore/detail/grabley-product-sourcing-ext/ (дата звернення: 11.05.2025).

24. Keepa – Amazon Price Tracker. – Режим доступу: https://keepa.com/ (дата звернення: 11.05.2025).

25. Helium 10 – Amazon Seller Software. – Режим доступу: https://www.helium10.com/tools (дата звернення: 11.05.2025).

26. OAGenius Platform Overview. – Режим доступу: https://www.oagenius.com/ (дата звернення: 11.05.2025).

27. Shopify Help Center. How to Start a Dropshipping Business – Режим доступу: https://www.shopify.com/blog/dropshipping (дата звернення: 11.05.2025).

28. DSers – Best AliExpress Dropshipping Tool – Режим доступу: https://www.dsers.com/ (дата звернення: 11.05.2025).

29. Coresight Research. The State of Dropshipping 2024 – Режим доступу: https://coresight.com/research/dropshipping-2024 (дата звернення: 11.05.2025).

30. DSM Tool – All-in-One Dropshipping Software. – Режим доступу: https://www.dsmtool.com/ (дата звернення: 11.05.2025).

31. ZIK Analytics. eBay Product Research Tool for Dropshipping. – Режим доступу: https://www.zikanalytics.com/ (дата звернення: 11.05.2025).

32. AutoDS – All-in-One Dropshipping Platform. – Режим доступу: https://www.autods.com/ (дата звернення: 11.05.2025).

33. SkuGrid – Automation for eBay & Amazon Sellers. – Режим доступу: https://www.skugrid.com/ (дата звернення: 11.05.2025).

34. PriceYak – Scalable Automation for Amazon & eBay. – Режим доступу: https://www.priceyak.com/ (дата звернення: 11.05.2025).

35. Easync – Dropshipping Automation Tool. – Режим доступу: https://www.easync.io/ (дата звернення: 11.05.2025).

36. RepricerExpress – Amazon & eBay Repricing Tool. – Режим доступу: https://www.repricerexpress.com/ (дата звернення: 11.05.2025).

37. InkFrog – eBay Listing Tool. – Режим доступу: https://www.inkfrog.com/ (дата звернення: 11.05.2025).

38. Hustle Got Real – Dropshipping Tool for eBay, Amazon. – Режим доступу: https://hustlegotreal.com/ (дата звернення: 11.05.2025).

39. ShopMaster – All-in-One Dropshipping Platform (архівована версія). – Режим доступу: https://web.archive.org/web/20210224090542 / (дата звернення: 11.05.2025).

ДОДАТКИ

Додаток А

```
import os
import pandas as pd
import time
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from sklearn.ensemble import RandomForestClassifier
import joblib

# 1. Ініціалізація Selenium
options = Options()
options.add_argument('--headless')
options.add_argument('--disable-gpu')
driver = webdriver.Chrome(options=options)

# 2. Завантаження CSV
df = pd.read_csv("new_perevirka.csv")
df['Рейтинг'] = None
df['Відгуки'] = None
df['Безкоштовна доставка'] = None

# 3. Парсер
def parse_ebay_data(url):
    try:
        if not url.strip().endswith("?"):
            url += "?"

        driver.get(url)
```

```
            time.sleep(2)

            rating_text = "Невідомо"
            try:
                elems = driver.find_elements(By.XPATH, "//span[contains(@class, 'ux-textspans--PSEUDOLINK')]")
                for el in elems:
                    txt = el.text.lower()
                    if "%" in txt and "positive" in txt and "product" not in txt:
                        rating_text = el.text.strip()
                        break
            except:
                pass

            feedback_count = 0
            try:
                elems = driver.find_elements(By.XPATH, "//span[contains(text(), '(')]")
                for el in elems:
                    digits = ''.join(filter(str.isdigit, el.text))
                    if digits:
                        feedback_count = int(digits)
                        break
            except:
                pass

            delivery = 0
            try:
                shipping_price_element = driver.find_element(By.XPATH, "//div[contains(@data-testid, 'ux-labels-values__values-content')]//span[contains(@class, 'ux-textspans--BOLD')]")
```

```
            if "free" in shipping_price_element.text.strip().lower():
                delivery = 1
        except:
            pass

        print(f"☆ Рейтинг: {rating_text}, 💬 Відгуки: {feedback_count}, 🚚 Доставка: {'Так' if delivery else 'Ні'}")
        return rating_text, feedback_count, delivery

    except Exception as e:
        print(f"✖ Помилка: {e}")
        return "Невідомо", 0, 0

# 4. Парсинг
for i in range(len(df)):
    url = df.loc[i, 'link']
    rating, feedbacks, delivery = parse_ebay_data(url)
    df.loc[i, 'Рейтинг'] = rating
    df.loc[i, 'Відгуки'] = feedbacks
    df.loc[i, 'Безкоштовна доставка'] = delivery

driver.quit()

# 5. Обробка даних
df['rating_clean'] = pd.to_numeric(df['Рейтинг'].str.extract(r'([\d\.]+)')[0], errors='coerce').fillna(0)
df['feedbacks'] = pd.to_numeric(df['Відгуки'], errors='coerce').fillna(0).astype(int)
df['delivery'] = pd.to_numeric(df['Безкоштовна доставка'], errors='coerce').fillna(0).astype(int)
```

```
df['has_rating'] = df['rating_clean'].apply(lambda x: 0 if x == 0 else 1)

# 6. Завантаження моделі
model = joblib.load("rf_model.joblib")

# 7. Формуємо X — ТІЛЬКИ з ознак, які знає модель
expected_features = list(model.feature_names_in_)
X = df[expected_features]

# 8. Прогноз
df['class'] = model.predict(X)

# 9. Збереження
df.to_csv("classified_output2.csv", index=False)
print("✅ Успішно збережено як classified_output.csv")
```

Додаток Б – Копія тез

Павленко А.В., студент
Київський національний економічний університет імені Вадима Гетьмана
pavlenko.artem@kneu.ua

ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ДРОПШИПІНГУ НА ПЛАТФОРМІ AMAZON

Розвиток електронної комерції зумовив стрімке зростання популярності дропшипінгу як бізнес-моделі, що дозволяє здійснювати торгівлю без необхідності зберігання товару або управління логістикою. Серед найбільш динамічних платформ у цій сфері — Amazon, яка забезпечує глобальний доступ до ринку, проте одночасно накладає жорсткі вимоги до ефективності, точності й швидкості бізнес-процесів. У таких умовах ключовим фактором успішної діяльності стає автоматизація основних операцій із використанням інтелектуальних систем, здатних адаптуватися до ринкових умов у реальному часі.

Проєктування інтелектуальної системи для автоматизації процесів дропшипінгу передбачає створення комплексного програмного рішення, яке забезпечує повний цикл роботи — від збору даних про постачальників, моніторингу залишків і цін, до автоматичного розміщення замовлень, обробки запитів клієнтів і прогнозування попиту. Застосування методів машинного навчання, нейронних мереж і алгоритмів обробки природної мови (NLP) забезпечує гнучкість, масштабованість та адаптивність системи. Така система здатна виявляти приховані закономірності в поведінці користувачів, оцінювати ризики, а також автоматично приймати рішення на основі великого обсягу вхідних даних.

Автоматизовані модулі для аналізу ринку й оптимізації цінової політики можуть бути реалізовані за допомогою методів глибинного навчання, що дозволяє враховувати не лише історичні дані, а й поточні тренди, сезонність, поведінкові патерни споживачів. Такі підходи значно перевершують традиційні інструменти управління ціноутворенням, які не мають гнучкості та не пристосовані до високої мінливості умов торгівлі на Amazon.

Система також повинна враховувати специфіку міжнародної логістики, зокрема — варіативність строків доставки, обмеження для імпорту в певні країни, правила повернень, що встановлені платформою. Завдяки автоматизованому підбору постачальників та маршрутів доставки, можна зменшити кількість відмов, затримок або скарг з боку клієнтів, що позитивно впливає на рейтинг продавця. Окрему увагу необхідно приділити модулю виявлення фрод-активності та системам безпеки, які аналізують транзакції, підозрілі замовлення, поведінку користувачів та попереджають можливі шахрайські дії.

Крім технічного аспекту, важливим є й дотримання вимог політик Amazon, які постійно оновлюються. Автоматизована система повинна регулярно перевіряти актуальність даних щодо правил платформи та, за потреби, змінювати сценарії взаємодії. Це мінімізує ризик блокування акаунту, видалення товарів або втрати позицій у пошуковій видачі.

Розробка системи має бути орієнтована на користувача і передбачати зручний інтерфейс для адміністрування, а також функціонал для аналізу показників ефективності — прибутковості, часу обробки замовлень, частоти відгуків тощо. У перспективі така система може бути масштабована на інші платформи електронної комерції, що відкриває нові можливості для бізнесу.

Отже, інтелектуалізація процесів дропшипінгу на платформі Amazon сприяє не лише підвищенню ефективності операцій, а й створенню більш стабільної, гнучкої та конкурентоспроможної бізнес-моделі. Комплексне проєктування системи з використанням сучасних технологій штучного інтелекту забезпечує автоматизовану взаємодію з усіма елементами торгового ланцюга та відповідає актуальним вимогам електронної торгівлі.

Впровадження таких рішень є важливим кроком до створення інноваційних систем електронного бізнесу нового покоління.

Список використаних джерел

1. Moshkovich, H., & LaViola, J. Amazon Marketplace Automation: Integrating API-Based Solutions for Inventory and Order Management. Journal of E-Commerce Research, 2021, 22(3), 45–58
2. Gupta, S., & Srivastava, S. Intelligent Dropshipping Systems: A Review of AI-Powered Automation Tools. International Journal of Retail & Distribution Management, 2020, 48(9), 987–1002.
3. Chen, J., & Hu, Y. Intelligent Decision Support for E-Commerce Using Machine Learning: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, 2022, 52(1), 42–55.

Науковий керівник: Галузинський Г.П., кандидат технічних наук