

КРОХМАЛЮК ВІТАЛІЙ ВІТАЛІЙОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій
к.т.н., доцент
_____ О. В. Зелінська
« ____ » _____ 2024 р.

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ДАНИХ ОЦІНКИ
СПОЖИВЧОЇ ЦІННОСТІ ТОВАРУ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (магістерська) робота

Науковий керівник:
Н. А. Потапова, доцент кафедри
інформаційних технологій,
к.е.н., доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

АНОТАЦІЯ

Крохмалюк В. В. Інформаційна технологія аналізу даних оцінки споживчої цінності товару. Спеціальність 122 «Комп'ютерні науки», Освітня програма «Комп'ютерні технології обробки даних (Data Science)». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (магістерській) роботі досліджено поняття інформаційної технології для аналізу даних оцінки споживчої цінності товару. Інформаційна технологія збирає актуальні дані про товар, за допомогою парсера з використанням мови програмування Python та додаткових бібліотек, таких як requests, BeautifulSoup, time та json. Після чого відбираються основні характеристики для їх нормалізації та вираховування вагових коефіцієнтів. На основі цих даних відбувається оцінка споживчої цінності товарів.

Магістерська робота складається з вступу, трьох розділів, висновків. У вступі обґрунтовується актуальність теми, описується мета роботи та постановка завдань дослідження. У першому розділі розглянуто теоретичні основи технологій аналізу даних при оцінці споживчої цінності товару. У другому розділі висвітлено ключові аспекти розробки програмного забезпечення з урахуванням важливості вибору середовища, мови програмування та супутніх інструментів. Розглянута модель та принцип роботи парсера. У третьому розділі описаний процес розробки та практичний результат роботи інформаційної технології для оцінки споживчої вартості товарів. Виявлені ключові характеристики що впливають на споживчу вартість та розглянуті перспективи розвитку та вдосконалення технології.

72 с., 20 рис., 50 джерел.

Ключові слова: інформаційна технологія, парсинг, обробка даних, аналіз даних, споживча цінність товарів, JSON, Python.

ABSTRACT

Krokhmalyuk V. V. Information technology of data analysis of product consumer value estimation. Specialty 122 "Computer science", Educational program "Data Science". Vasyl' Stus Donetsk National University, Vinnytsia, 2024.

In the qualification (master's) work, the concept of information technology for the analysis of the data of the evaluation of the consumer value of the product was investigated. Information technology collects relevant product data using a parser using the Python programming language and additional libraries such as requests, BeautifulSoup, time and json. After that, the main characteristics are selected for their normalization and calculation of weighting factors. Based on these data, the consumer value of goods is estimated.

The master's thesis consists of an introduction, three sections, and conclusions. The introduction substantiates the relevance of the topic, describes the purpose of the work and setting the research objectives. The first chapter deals with the theoretical foundations of data analysis technologies for evaluating the consumer value of goods. The second chapter covers at the key aspects of software development, taking into account the importance of choosing an environment, programming language, and related tools. The considered model and working principle of the parser. The third chapter describes the process of development and the practical result of information technology for assessing the consumer value of goods. The key characteristics affecting the consumer price are identified and the prospects for the development and improvement of the technology are considered.

72 pages, 20 pictures, 50 sources.

Keywords: information technology, parsing, data processing, data analysis, consumer value of goods, JSON, Python.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	5
ВСТУП.....	6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТЕХНОЛОГІЙ АНАЛІЗУ ДАНИХ СПОЖИВЧОЇ ЦІННОСТІ ТОВАРУ	9
1.1 Сутність процесу збору даних при оцінці споживчої цінності товару	9
1.2 Порівняльний аналіз аналогів технологій аналізу даних споживчої цінності товару.....	12
1.3 Основні задачі аналізу даних споживчої цінності товару.....	16
Висновки до 1 розділу.....	23
РОЗДІЛ 2. ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ДОДАТКУ АНАЛІЗУ ДАНИХ СПОЖИВЧОЇ ЦІННОСТІ ТОВАРУ	24
2.1 Аналіз інструментів та методів для створення інформаційної технології	24
2.2 Огляд мов програмування та бібліотек	29
2.3. Створення моделі парсеру для збору даних характеристик товару	44
Висновки до 2 розділу.....	46
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ДОДАТКУ ДЛЯ ОЦІНКИ СПОЖИВЧОЇ ЦІННОСТІ ТОВАРУ	47
3.1 Процес збору даних для оцінки споживчої цінності товару	47
3.2 Аналіз даних споживчої цінності товару	51
3.3 Перспективи вдосконалення технології аналізу даних споживчої цінності товару	61
Висновки до 3 розділу.....	64
ВИСНОВКИ	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ	67
ДОДАТКИ	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – (application programming interface) програмний інтерфейс.

Web scrapping – (веб-скрапінг) технологія отримання веб-даних шляхом вилучення їх зі сторінок веб-ресурсів.

HTML – (HyperText Markup Language) мова розмітки документів для перегляду вебсторінок у браузері.

CSS – (Cascading Style Sheets) каскадні таблиці стилів.

XML – (EXtensible Markup Language) розширювана мова розмітки.

JSON – (JavaScript Object Notation) текстовий формат обміну даними між комп'ютерами.

NLP – (Natural Language Processing) обробка природної мови.

IDE – (integrated development environment) інтегрована середовище розробки.

ВСТУП

Актуальність теми. Зміни в умовах ринку та постійне зростання конкуренції вимагають від підприємств постійного оновлення своїх стратегій продажу та маркетингу. Інформаційна технологія аналізу даних дозволяє підприємствам ефективно збирати, обробляти та аналізувати величезні обсяги даних про споживчу поведінку. Це допомагає визначати стратегії ціноутворення, прогнозувати попит та адаптувати асортимент товарів, щоб максимізувати задоволення потреб споживачів, а також виявляти залежності, тенденції та патерни, що стає важливим для прийняття обґрунтованих рішень. Аналіз даних дозволяє точно визначити цільові аудиторії, розробляти персоналізовані кампанії та вдосконалювати комунікаційні стратегії. За допомогою аналізу даних можна розробляти персоналізовані стратегії маркетингу, спрямовані на індивідуальні потреби споживачів. Це не лише підвищує ефективність рекламних кампаній, але й сприяє збереженню та залученню нових клієнтів.

Інформаційна технологія дозволяє в режимі реального часу відслідковувати та адаптуватися до змін в уподобаннях та потребах споживачів. Це дозволяє швидко реагувати на зміни в ринкових умовах і вносити корективи в стратегії підприємства. Усі ці фактори свідчать про важливість та актуальність інформаційної технології аналізу даних оцінки споживчої цінності товару для підприємств, які прагнуть ефективно конкурувати та відповідати на зміни у споживчому попиті.

Мета магістерської роботи полягає у розробці та реалізації інформаційної технології для аналізу даних оцінки споживчої цінності товару.

Завдання магістерської роботи. Для досягнення мети було поставлено ряд завдань:

1. Дослідити теоретичні основи технологій аналізу даних споживчої цінності товару з метою систематизації та узагальнення ключових понять та характеристик даних процесів.

2. Дослідити методи для створення інформаційної технології, що будується на процесах збору та аналізу даних, сутність яких відтворює особливості категорій оцінки споживчої цінності товарів (на прикладі ноутбуків), включаючи технічні характеристики, відгуки користувачів, цінову інформацію та інші важливі параметри.

3. Проаналізувати інструменти для реалізації інформаційної технології, на підставі чого обґрунтувати можливість використання парсингу для збору даних та їх подальшої обробки в системі оцінювання споживчої цінності товарів.

4. Реалізувати інформаційну технологію, яка включає в себе методи парсингу даних, обробки і аналізу, для отримання повної та об'єктивної оцінки споживчої цінності ноутбуків

5. Провести аналіз отриманих даних для визначення ключових факторів, які впливають на споживчу цінність товару та характеризують вплив технічних характеристик, бренду, регіональних відмінностей тощо.

6. Зробити ключові висновки на основі проведеного аналізу та розробити рекомендації для вирішення питань застосування розробленої інформаційної технології.

Об'єктом дослідження є інформаційна технологія аналізу даних оцінки споживчої цінності товару.

Предметом дослідження є методи, інструменти та програмні засоби, що дозволяють сформулювати технологію обробки та аналізу даних з акцентом на ефективні характеристики споживчої цінності товарів.

Наукова новизна дослідження полягає в удосконаленні підходів щодо аналізу даних оцінки споживчої цінності товару (зокрема, ноутбуків), яка поєднує елементи системи парсингу та критеріальної рейтингової оцінки. Спрямовання на конкретну категорію товарів (ноутбуки) дозволяє розглядати специфічні аспекти та особливості споживчої цінності цієї техніки.

Структура роботи. Магістерська робота складається із вступу, трьох розділів, висновків та списку використаних джерел. В першому розділі досліджуються теоретичні основи технологій аналізу даних споживчої цінності

товару. Другий розділ присвячений питанням аналізу інструментів та методів для створення технології. В третьому розділі надані результати розробки та практичної реалізації технології аналізу даних оцінки споживчої цінності товару.

Практичне значення дослідження полягає у можливості використання наданих конкретних інструментів та методів для ефективного аналізу даних оцінки споживчої цінності товару, зосереджуючись на ноутбуках.

Результати дослідження можуть служити підставою для вирішення питань щодо покращення технічних характеристик товарів, їхньої цінової політики та інших параметрів, що впливають на споживчу цінність ноутбуків. Це дозволяє компаніям удосконалювати свій продуктовий асортимент відповідно до потреб споживачів.

Аналіз даних дозволяє передбачати та уникати можливих ризиків, пов'язаних із змінами в ринкових умовах та споживчому попиті. Це допомагає підприємствам бути готовими до змін і адаптуватися до них.

Апробація результатів дослідження:

- Потапова Н.А., Денисюк В.О., Крохмалюк В.В. Парсинг як метод обробки даних при оцінці споживчої цінності товарів. Наука і техніка сьогодні. 2023. №13(27). С.819-827. (Фахова, категорія Б)
- Крохмалюк В.В., Нескородєва Т.В. Взаємозв'язок задач оптимізації та задач рекомендаційних системах. Матеріали IV всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених. Вінниця, 2023. С. 265 – 266.
- Крохмалюк В.В., Потапова Н.А. Використання інструментів для збору даних в дослідницькій діяльності як ефективний метод отримання інформації. *Комп'ютерні технології обробки даних: матеріали IV Всеукраїнської науково-практичної конференції* (м. Вінниця, 8 грудня 2023 р.). Вінниця: ДонНУ імені Василя Стуса, 2022. С. 31 - 33.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ТЕХНОЛОГІЙ АНАЛІЗУ ДАНИХ

СПОЖИВЧОЇ ЦІННОСТІ ТОВАРУ

1.1 Сутність процесу збору даних при оцінці споживчої цінності товару

Інформаційна технологія – це широкий термін, який використовується для опису використання комп'ютерних систем, мереж, програмного забезпечення та інших технічних засобів для збирання, зберігання, обробки, передачі та отримання інформації [40]. ІТ включає в себе різноманітні технології, які дозволяють працювати з інформацією в різних формах: текстовій, звуковій, графічній, відео та інші. Основні складові інформаційної технології включають:

1. Комп'ютерні системи. Включають в себе апаратне забезпечення (комп'ютери, сервери, сенсори тощо) та програмне забезпечення (операційні системи, додатки, програми).
2. Мережі. Забезпечують зв'язок між комп'ютерами та іншими пристроями, щоб дозволити обмін інформацією.
3. Програмне забезпечення. Включає в себе програми та додатки, які використовуються для виконання різних завдань, обробки даних та автоматизації різноманітних процесів.
4. Інтернет та веб-технології. Забезпечують доступ до світової мережі та можливість спілкування, обміну даними та використання веб-ресурсів.
5. Бази даних. Використовуються для зберігання та управління великими обсягами даних.
6. Інформаційні системи. Забезпечують обробку, аналіз та представлення інформації для підтримки прийняття рішень [6].

Інформаційна технологія є невіддільною частиною сучасного суспільства та бізнесу, допомагає покращувати ефективність, зручність та доступність обробки інформації в різних галузях життя.

Збір та аналіз даних є важливою складовою в умовах сучасного швидкого технологічного розвитку та динамічних змін у споживчих уподобаннях.

Мета збору даних про характеристики ноутбуків полягає у створенні цілісної бази даних, яка містить інформацію про різні моделі девайсів і їх характеристики. Ці дані можуть бути корисні для різних зацікавлених сторін, включаючи споживачів, IT-компанії, ремонтні сервіси та навіть роздрібних магазинів. Ось кілька причин, чому збір таких даних може бути корисним:

1. Збір і аналіз характеристик ноутбуків може допомогти вам зрозуміти тенденції на ринку, такі як попит на певні технічні характеристики (наприклад, процесори, обсяги пам'яті, роздільна здатність екрану) і як ці характеристики змінюються з часом.

2. Ви можете створити інструмент, який дозволяє користувачам порівнювати різні моделі ноутбуків за їхніми характеристиками. Це може бути корисно для покупців, які шукають ноутбук, що відповідає їхнім потребам.

3. Збір даних про характеристики ноутбуків і їхні ціни може допомогти вам розробити модель для прогнозу цін на ринку. Це може бути корисно як для покупців, так і для продавців.

4. Дані про характеристики ноутбуків можуть бути використані для аналізу популярності певних брендів, моделей або функціональних можливостей серед споживачів.

Збір даних дозволяє мати доступ до широкого спектру інформації про ринок ноутбуків, що може бути важливим для прийняття управлінських рішень або для інформування споживачів. Ця інформація допомагає слідкувати за конкурентами, прогнозувати попит, а якщо також наявна інформація про вимоги та уподобання користувачів, можна надавати персоналізовані рекомендації для покупців та використовувати їх для орієнтування рекламних кампаній на споживачів, які шукають конкретні характеристики ноутбуків [39]. Нижче більш детально наведені приклади використання цих даних в різних сферах.

Виробники ноутбуків можуть використовувати дані для оцінки популярності своїх моделей та визначення тих характеристик, які найбільше

цікавлять споживачів. Це допомагає виробникам розробляти більш ефективні рекламні кампанії та маркетингові стратегії.

Онлайн-роздрібні магазини можуть використовувати зібрані дані для надання персоналізованих рекомендацій користувачам. Наприклад, якщо користувач шукає ноутбук для ігор, система може рекомендувати моделі з потужними графічними картами. Сервіси порівняння цін можуть використовувати дані для порівняння цін на різні моделі ноутбуків та показати користувачам найкращі пропозиції [40].

Індивідуальні покупці можуть використовувати дані для порівняння різних ноутбуків і прийняття кращих рішень при покупці.

Аналітики ринку можуть використовувати дані для відстеження тенденцій та змін на ринку ноутбуків. Наприклад, вони можуть виявити, які компанії випускають найбільше ноутбуків з певними характеристиками.

Компанії можуть збирати дані про характеристики ноутбуків, які вони продали, та використовувати ці дані для визначення рівня задоволеності своїх клієнтів. Це може бути корисним для поліпшення якості продуктів і обслуговування [29].

Академічні та дослідницькі організації можуть використовувати дані для проведення досліджень, таких як аналіз впливу технологій на продуктивність користувачів або оцінка ефективності різних функцій ноутбуків.

Виробники та сервісні центри можуть використовувати дані для вирішення проблем із ноутбуками та надання більш якісної технічної підтримки.

Ці приклади показують, що дані про характеристики ноутбуків можуть бути використані для різних цілей, від бізнес-аналітики до поліпшення досвіду покупців. Важливо зберігати та обробляти ці дані з уважністю до конфіденційності та законності збору.

1.2 Порівняльний аналіз аналогів технологій аналізу даних споживчої цінності товару

В розділі виконується аналіз та порівняння існуючих аналогічних технологій або рішень, що вже існують на ринку. Перед впровадженням розробленої інформаційної технології важливо визначити, чим саме вона вирізняється від інших схожих продуктів, а також які переваги та недоліки можуть бути виявлені під час порівняння.

Пошук аналогів дозволяє визначити ринкову конкурентоспроможність та унікальність розробленої технології. Під час аналізу аналогів слід звертати увагу на їхні можливості, функціональність, ефективність, а також на відгуки користувачів і відомості про їхнє використання в реальних умовах.

Деякі ключові аспекти, що можуть бути враховані під час порівняння, включають точність аналізу, швидкість обробки даних, зручність інтерфейсу, можливість масштабування та адаптивність до змінних умов. Пошук аналогів стає стратегічним етапом, допомагаючи виявити конкурентні переваги та можливості для вдосконалення розробленої технології перед введенням її на ринок. Було знайдено та розглянуто такі аналоги для парсингу даних:

1. Octoparse представляє собою потужний інструмент для веб-скрапінгу, що відзначається наявністю графічного інтерфейсу для створення правил збору інформації (рис. 1.1). Цей інструмент надає можливість вилучати дані з різноманітних джерел, таких як веб-сторінки, PDF-файли, бази даних та інші ресурси. У своєму функціоналі Octoparse має вбудовані інструменти для обробки даних та здатність експортувати зібрану інформацію у різні формати. Octoparse вирізняється не лише своєю потужністю, але й зручним графічним інтерфейсом, що значно спрощує процес створення правил для збору даних. Це робить роботу з інструментом більш інтуїтивно зрозумілою та приємною для користувача [18].

Octoparse наділено вбудованими інструментами для ефективної обробки даних та пропонує різні варіанти експорту. Це дозволяє користувачам гнучко працювати з інформацією, роблячи Octoparse універсальним інструментом для аналізу та використання даних.

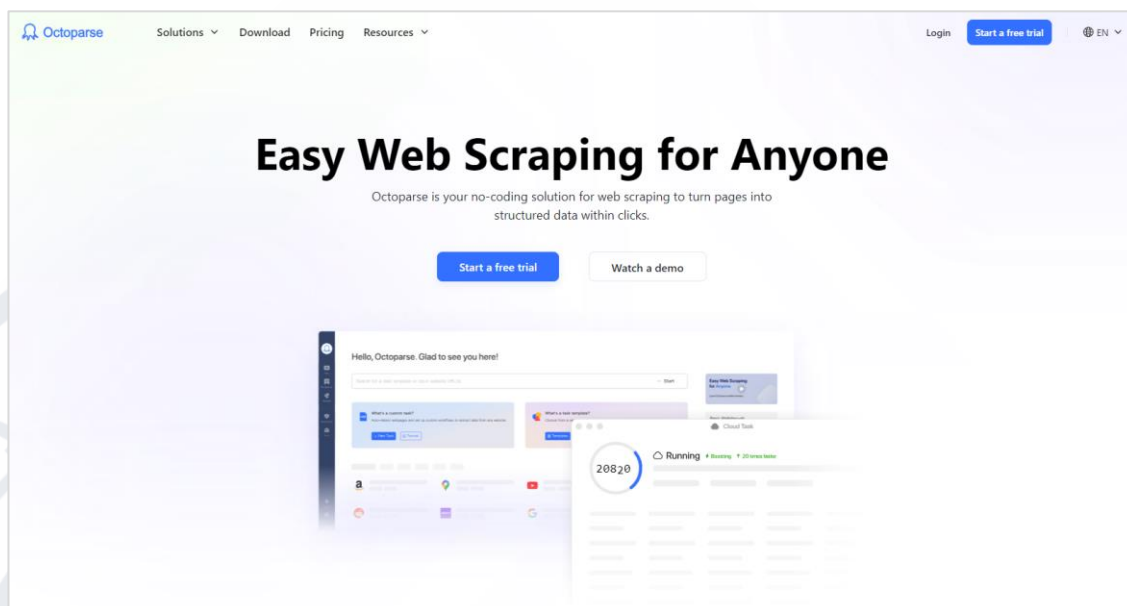


Рисунок 1.1 – Головна сторінка Octoparse

Проте він має свої недоліки. Освоєння функціоналу Octoparse може зайняти певний час через необхідність ознайомлення з інтерфейсом інструменту. Це може вимагати деякого терпіння та зусиль від користувача, перш ніж він зможе повністю використовувати всі можливості інструменту.

2. ParseHub також є високоефективним інструментом для веб-скрапінгу, який вражає своїм графічним інтерфейсом (рис. 1.2). Цей інструмент дозволяє створювати правила парсингу, обираючи необхідні елементи на веб-сторінці та вилучаючи важливі дані. ParseHub володіє автоматичним розпізнаванням структури веб-сторінок і вбудованими функціями для ефективної обробки та очищення зібраних даних [28].

ParseHub видається важливим інструментом завдяки своєму інтуїтивно зрозумілому графічному інтерфейсу та автоматичному розпізнаванню структури веб-сторінок. Це значно полегшує процес парсингу, роблячи його доступнішим для користувачів з різним рівнем досвіду.

В порівнянні з іншими інструментами, ParseHub може виявитися менш ефективним у сфері обробки даних. Особливо при порівнянні з конкурентами, інструмент може мати певні обмеження в області обробки та аналізу інформації.

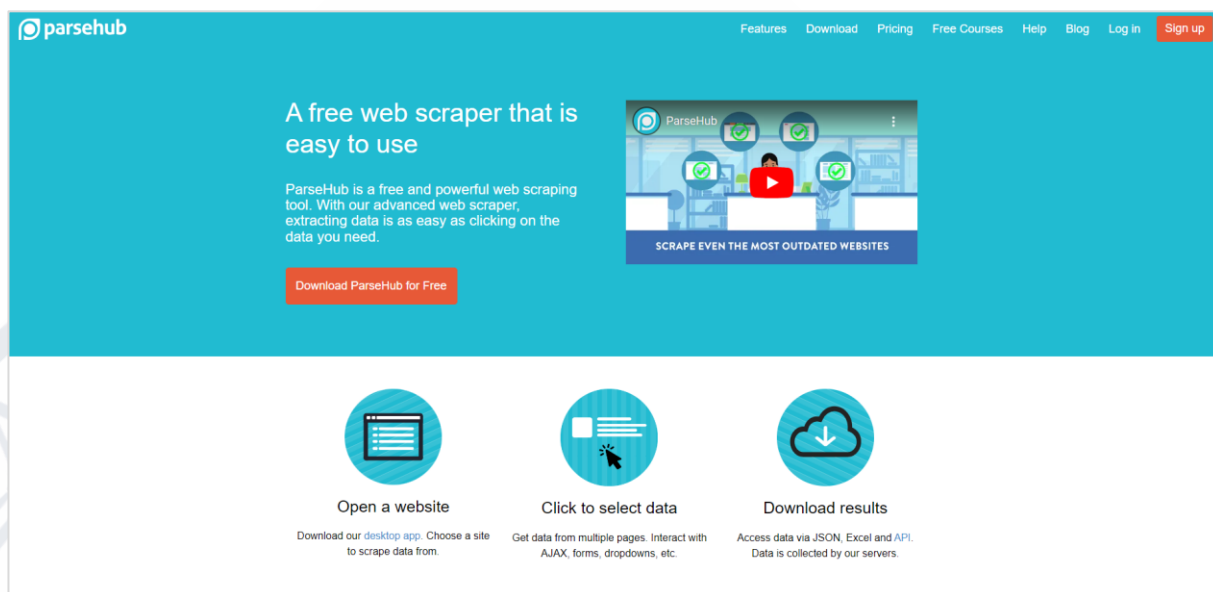


Рисунок 1.2 – Головна сторінка ParseHub

3. Import.io представляє собою хмарний сервіс, спрямований на парсинг та збір інформації з веб-сторінок (рис. 1.3). Цей сервіс надає інструменти для створення правил парсингу та видобування необхідних даних. Крім того, Import.io володіє можливостями автоматизованого оновлення інформації та інтеграції з іншими системами, роблячи його комплексним інструментом для здійснення ефективного аналізу даних.

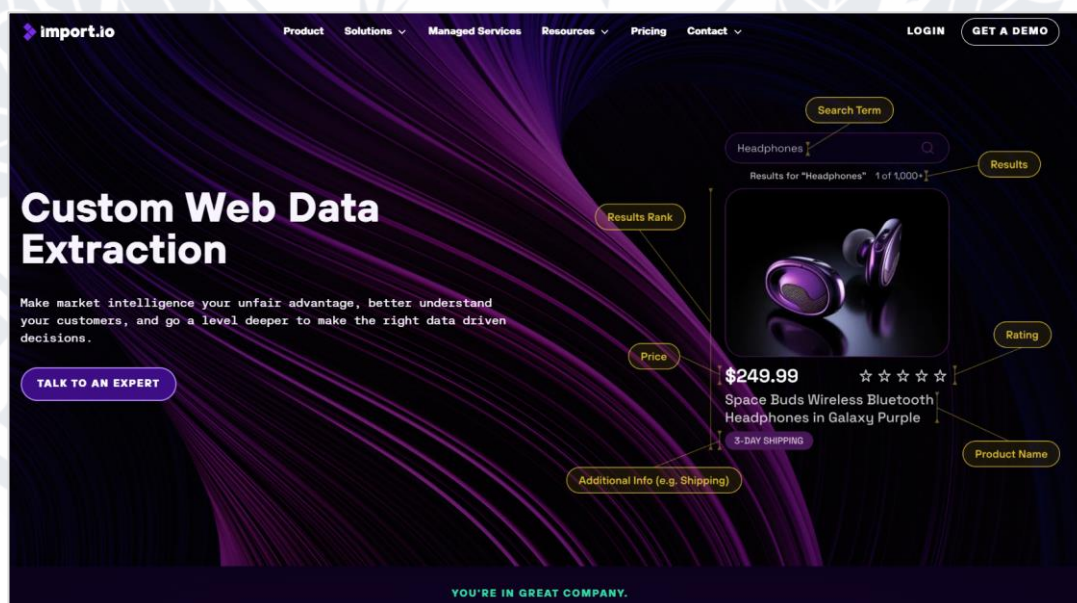


Рисунок 1.3 – Головна сторінка Import.io

Import.io виступає як ефективний хмарний сервіс, забезпечуючи доступ до даних з будь-якого місця і автоматизовані функції оновлення інформації. Це робить процес отримання актуальних даних більш зручним та ефективним.

Користування Import.io може вимагати оплати за певний обсяг даних, що може впливати на вартість користування. Платіжні обмеження можуть становити виклик для користувачів, особливо в залежності від їхніх потреб та фінансових можливостей.

4. Webhose.io є API-платформою, яка дозволяє отримати доступ до широкого спектру структурованих даних з Інтернету. Ця платформа пропонує різноманітні категорії інформації, включаючи новини, соціальні медіа, відгуки, блоги та інше. Webhose.io забезпечує зручні інструменти для налаштування запитів, щоб користувач міг легко отримати необхідні дані з визначених джерел в Інтернеті.



Рисунок 1.4 – Головна сторінка Webhose.io

Webhose.io є відмінним інструментом завдяки своєму широкому спектру категорій даних. Це надає користувачам можливість отримувати різноманітну та вичерпну інформацію з різних джерел та галузей.

У порівнянні з іншими аналогами, Webhose.io може виявитися менш потужним щодо глибокого аналізу даних. Його інструменти для детального

розгляду інформації можуть бути обмеженими, що варто врахувати при виборі для конкретних завдань.

Ці готові додатки для парсингу даних мають свої унікальні функції і можливості. Вибір конкретного додатку залежить від ваших потреб, технічних вимог і бюджету проекту.

1.3 Основні задачі аналізу даних споживчої цінності товару

Постановка задачі полягає в зборі та структуризації даних, які були зібрані за допомогою парсера з інтернет магазину, з метою їх подальшої оцінки споживчої вартості товару. Головною метою цього процесу є подальший аналіз і оцінка споживчої вартості товарів. Визначення споживчої вартості є ключовим етапом для розкриття значущості продуктів для кінцевих споживачів і вироблення стратегій у сферах маркетингу та продажу. Цей аналіз сприятиме не лише підвищенню розуміння цінності товарів для споживачів, але й надаватиме підстави для розробки ефективних стратегій, спрямованих на задоволення їхніх потреб і підвищення їхнього інтересу до продукції.

Вирішення завдання синтаксичного аналізу вмісту веб-сторінки та подальшої структуризації вимагає використання належно підібраних інструментів. Саме для цих потреб ідеальним інструментом є парсинг даних, який забезпечує ефективний збір значущих обсягів інформації, охоплюючи відгуки, оцінки та коментарі споживачів про товар або важливі дані про сам продукт [45].

Парсинг даних стає ключовим інструментом для виконання цієї задачі, відкриваючи можливість збирати і аналізувати та порівнювати дані з різних джерел, виявляти ключові фактори, що впливають на споживчу вартість, створювати рейтинги та прогнозувати тенденції споживання.

Парсинг даних є цінним інструментом для оцінки та аналізу споживчої вартості товару, допомагаючи компаніям покращувати свої продукти та задовольняти потреби споживачів.

Основні переваги парсингу для оцінки та аналізу споживчої вартості полягають у тому, що:

1. Парсинг даних дозволяє автоматично збирати великі обсяги даних з різних джерел, таких як веб-сторінки, соціальні мережі, форуми та інші джерела. Це дозволяє отримати доступ до великої кількості відгуків, оцінок, коментарів та інших даних, пов'язаних зі споживчою вартістю товару [35].

2. Парсинг даних дозволяє проводити аналіз та порівняння даних з різних джерел. Наприклад, можна зібрати відгуки про товар з різних веб-сайтів та соціальних мереж, а потім аналізувати їх для виявлення загальних тенденцій, популярних функцій або недоліків, а також порівнювати товари між собою за різними характеристиками.

3. Парсинг даних дозволяє виявити ключові фактори, що впливають на споживчу вартість товару. Наприклад, шляхом аналізу відгуків та коментарів споживачів можна виявити, які конкретні аспекти товару є найбільш значущими для споживачів, чи це якість, функціональність, дизайн, ціна або інші фактори.

4. Зібрані дані можуть бути використані для створення рейтингів, оцінок та репутаційних показників для товарів. На основі аналізу даних можна визначити, які товари мають високу споживчу вартість та позитивну репутацію, а які можуть мати проблеми або низьку оцінку від споживачів.

5. Парсинг даних дозволяє виявляти та прогнозувати тенденції споживання, популярність та споживчу вартість товарів у часі. Наприклад, на основі аналізу відгуків та оцінок можна визначити зміни в сприйнятті товару споживачами та передбачити, які характеристики можуть бути популярними в майбутньому.

Загалом, парсинг даних допомагає зрозуміти та оцінити споживчу вартість товару шляхом збору, аналізу та порівняння великих обсягів даних. Це дозволяє компаніям отримувати цінні інсайди, поліпшувати продукти та стратегії маркетингу, а також задовольняти потреби та вимоги споживачів ефективніше [24].

Постановка задачі для створення інформаційної технології аналізу даних оцінки споживчої вартості ноутбуків передбачає визначення ключових кроків та завдань, які необхідно виконати для розробки ефективного та функціонального інструменту. Отже розглянемо основні етапи задачі створення інформаційної технології аналізу даних оцінки споживчої вартості ноутбуків:

1. Здійснити пошук та ретельний аналіз існуючих аналогів інформаційних технологій, спрямованих на ефективний аналіз даних. Мета цієї задачі полягає в тому, щоб ідентифікувати та проаналізувати існуючі ринкові рішення, спрямовані на аналіз даних, зокрема ті, які використовуються для оцінки споживчої вартості товарів. Під час виконання цієї задачі слід звертати увагу на функціональність, ефективність, можливості масштабування та інші ключові характеристики кожного аналогу. Результати аналізу будуть використані для визначення переваг і недоліків існуючих рішень і напрямків для подальшого вдосконалення розробленої Інформаційної технології.

2. Проаналізувати та обрати інструменти для реалізації інформаційної технології, висвітлення основних аспектів парсингу даних. Ця задача передбачає детальний аналіз інструментів, які можна використовувати для реалізації інформаційної технології, зокрема тих, які відіграють ключову роль у парсингу даних. Важливо врахувати основні аспекти парсингу, такі як ефективність, точність, можливості роботи зі структурованими та неструктурованими даними. Результати аналізу будуть використані для обрання оптимальних інструментів для подальшої реалізації технології.

3. Зібрати та підготувати дані. Передбачається систематичне та організоване здобування інформації з різних джерел, щоб забезпечити повноту та репрезентативність даних. Одним із ключових завдань є налагодження ефективного механізму збору інформації, включаючи інтернет-магазини, форуми та соціальні мережі. Забезпечення цього процесу дозволяє отримувати різноманітні перспективи і взяття даних із різних джерел, що є ключовим для об'єктивного аналізу.

Далі, важливо використовувати технологію парсингу для отримання та структуризації інформації про технічні характеристики ноутбуків, їх вартість, відгуки користувачів та інші параметри. В процесі парсингу важливо враховувати різноманіття форматів та джерел інформації, щоб забезпечити їхню коректну та уніфіковану обробку [30]. Цей етап є основоположним для підготовки даних до подальшого аналізу та додаткової обробки.

Остаточо, після отримання даних, необхідно провести їхню обробку та очищення, щоб усунути потенційні помилки, дублікати або інші непридатні для аналізу артефакти. Цей етап має на меті забезпечити якість та достовірність інформації, яка буде використана в подальших етапах розробки технології аналізу даних.

4. Розробити аналітичний інструмент. Передбачається створення комплексу інструментів, спрямованих на глибокий аналіз та порівняння споживчої вартості ноутбуків на основі зібраних раніше даних. Однією з основних завдань є розробка алгоритмів, які дозволять системі визначити та врахувати різноманітні аспекти споживчої цінності, такі як технічні характеристики, вартість та відгуки користувачів. Спеціалізовані аналітичні алгоритми визначатимуть вагомість кожного фактора в контексті визначення споживчої вартості товарів.

Додатково, буде проведена розробка модулів, які визначатимуть та виділятимуть ключові фактори, що мають значущий вплив на споживчу цінність ноутбуків. Це дозволить системі враховувати індивідуальні вподобання споживачів та розуміти, які аспекти товарів мають найбільший вплив на їхню загальну оцінку.

5. Визначити параметри оцінки. Одним із ключових етапів у процесі розробки інформаційної технології аналізу даних щодо споживчої вартості ноутбуків є визначення критеріїв та параметрів, які будуть використовуватися для об'єктивної оцінки їхньої споживчої цінності. Це включає в себе розробку чітких та адаптивних критеріїв, які охоплюють різноманітні аспекти, такі як технічні характеристики, якість конструкції, функціональні можливості та

задоволення користувачів.

6. Моделювання та прогнозування. На цьому етапі буде проведена розробка алгоритмів, спрямованих на виявлення та прогнозування тенденцій у споживчому ринку ноутбуків.

7. Реалізувати інформаційну технологію. Ця задача передбачає створення програмного продукту, який здатний автоматизовано збирати дані з різноманітних джерел, обробляти отриману інформацію та проводити аналіз для отримання повного та об'єктивного уявлення про споживчу вартість ноутбуків. Реалізація цієї технології повинна враховувати потреби дослідження та враховувати ключові параметри, такі як технічні характеристики, відгуки користувачів, цінова інформація та інші важливі аспекти, що впливають на споживчу вартість.

8. Тестування та оптимізація. Останнім етапом реалізації інформаційної технології аналізу даних оцінки споживчої вартості ноутбуків є проведення тестування на реальних даних з метою оцінки ефективності та точності системи. Це важливий крок, оскільки він дозволяє перевірити, наскільки добре розроблена технологія виконує свої завдання в реальних умовах та з реальними даними. Тестування на реальних даних допомагає виявити можливі недоліки, оптимізувати алгоритми та вдосконалити функціональність системи [32].

Додатково, проводитиметься оптимізація алгоритмів та процесів з метою забезпечення швидкості та стабільності роботи системи. Цей етап включатиме в себе аналіз та вдосконалення коду, використання ефективних алгоритмів обробки даних, а також вдосконалення роботи системи з точки зору реакції на зміни у вхідних даних. Оптимізація алгоритмів та процесів є ключовою для забезпечення продуктивності та надійності системи в умовах реальної експлуатації.

Завершальні етапи тестування та оптимізації визначають успішність розробленої технології та готовність її впровадження для вирішення завдань аналізу та оцінки споживчої вартості ноутбуків.

9. Провести аналіз отриманих даних для визначення ключових факторів, які впливають на споживчу вартість товару. Ця задача визначає необхідність систематичного аналізу зібраних даних з метою визначення та оцінки важливих факторів, які впливають на споживчу вартість ноутбуків. Аналіз повинен охоплювати різні аспекти, такі як технічні характеристики, бренд, регіональні відмінності та інші параметри, які можуть впливати на споживчу цінність товару. Результати цього аналізу будуть використані для формулювання висновків та розробки рекомендацій для оптимізації товарної стратегії та маркетингу.

10. Зробити ключові висновки на основі проведеного аналізу та розробити рекомендації для вирішення питань оптимізації товарної стратегії та маркетингу.

Ця задача ставить перед собою завдання формулювання ключових висновків на основі аналізу отриманих даних та розробки конкретних рекомендацій щодо оптимізації товарної стратегії та маркетингових підходів. Результати цієї задачі будуть важливим етапом для прийняття обґрунтованих рішень щодо покращення позиціонування товару на ринку та задоволення потреб споживачів.

Для досягнення мети, має бути написаний парсер. Цей парсер автоматично здійснюватиме запити до інтернет магазину, отримує HTML-сторінки з даними про ноутбуки та екстрагує необхідну інформацію, таку як назви ноутбуків, їх ціни, модельний ряд, операційну систему, діагональ дисплея, тип матриці, розширення, частота оновлення дисплея, серію та модель процесора, базову частоту, кількість ядер, об'єм оперативної пам'яті, тип оперативної пам'яті, частоту оперативної пам'яті, тип накопичувача, об'єм накопичувача, тип виробника, модель та об'єм відеокарти [19].

Після збору даних, основна задача полягає в структуризації цих даних для подальшої оцінки їх споживчої вартості. Це буде включати створення бази даних, де кожен ноутбук буде мати свій запис зі зібраними атрибутами. Це дозволить легко організувати та аналізувати дані для оцінки їх значення для споживачів.

Подальша оцінка споживчої вартості товару може включати різні аспекти, такі як порівняння цін на різні девайси, аналіз впливу певних критеріїв на ціну, врахування популярності та рейтингу ноутбуків серед споживачів тощо. Це допоможе зрозуміти, як ці дані впливають на споживчу вартість товару та як їх можна використати для стратегічного маркетингу та збільшення продажів [4].

Збір інформації проводиться з сайту інтернет магазину Фокстрот за посиланням <https://www.foxtrot.com.ua>. "Фокстрот" - це один з найбільших роздрібних мереж електроніки та побутової техніки в Україні. Основана компанія "Фокстрот" у 1994 році, і з тих пір вона стала одним із лідерів на ринку електроніки та техніки в Україні. Однією з основних особливостей мережі магазинів "Фокстрот" є її широкий асортимент товарів та велика кількість фізичних магазинів у різних регіонах України.

Отже, постановка задачі включає збір та структурування даних з веб-сайту аптеки, а також подальшу оцінку споживчої вартості цих даних для лікарських засобів. Цей процес допоможе отримати цінні інсайти та підвищити ефективність маркетингових стратегій, спрямованих на задоволення потреб споживачів.

ВИСНОВКИ ДО 1 РОЗДІЛУ

У розділі ми розглянули мету збору даних про ноутбуки, а також використання цих даних. Мета збору даних полягає в отриманні інформації про ноутбуки, їх детальні характеристики та ціни, щоб зробити аналіз споживчої вартості цих товарів.

Задача збору та структуризації даних з інтернет магазину включає в себе використання технологій, таких як веб-скрапінг, парсинг даних та обробка інформації. Це дозволяє отримати дані про ноутбуки з веб-сторінок інтернет-магазину і структурувати їх в зручний формат для подальшого аналізу та використання. Також ми провели пошук аналогів інформаційних технологій для парсингу даних. Знайдені інструменти, такі як Octoparse, ParseHub, Import.io та Webhose.io, надають широкий спектр можливостей для збору, структуризації та аналізу даних з різних джерел.

Розглядаючи інструменти для парсингу даних, ми визначили, що наявні на ринку рішення дозволяють автоматизувати та полегшити процес збору інформації з веб-сайтів, забезпечуючи високу точність та швидкість обробки. Важливим етапом були збір та підготовка даних, що вимагало вдосконаленого механізму збору інформації з різних джерел. Аналіз та оцінка даних дозволили визначити ключові фактори впливу на споживчу вартість ноутбуків.

Далі, розробивши алгоритми, ми впровадили та протестували їх, переконавшись у їхній ефективності та точності. Останнім етапом було впровадження розробленої технології в робочий процес, а також моніторинг її функціонування для забезпечення найвищої ефективності.

У підсумку, ми визначили, що збір та аналіз даних про ноутбуки є ключовим етапом для отримання цінної інформації щодо їх споживчої вартості. Застосування різноманітних інструментів, таких як веб-скрапінг, API, інструменти для видобування даних, обробка природної мови та алгоритми машинного навчання, допомагає забезпечити ефективний збір, обробку та аналіз цих даних.

РОЗДІЛ 2

ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ДЛЯ СТВОРЕННЯ ДОДАТКУ АНАЛІЗУ ДАНИХ СПОЖИВЧОЇ ЦІННОСТІ ТОВАРУ

2.1 Аналіз інструментів та методів для створення інформаційної технології

Оскільки парсинг даних - це процес автоматичного видобування та обробки інформації з веб-сторінок або інших джерел даних. У контексті ІТ та веб-розвитку ця техніка використовується для аналізу та отримання конкретних даних з різних ресурсів в Інтернеті. Під час парсингу програми, які називаються парсерами, автоматично сканують вміст веб-сторінок і екстрагують необхідні дані за заданими правилами [27].

Основні етапи парсингу включають в себе:

1. Збір інформації. Парсер отримує доступ до вмісту веб-сторінки або іншого джерела та збирає потрібні дані.
2. Видобування даних. Програма виділяє конкретні елементи або текст з отриманого вмісту, використовуючи задані правила або шаблони.
3. Обробка і структуризація. Отримані дані обробляються та структуризуються для подальшого використання. Цей етап може включати в себе очищення, фільтрацію та форматування інформації.
4. Збереження або виведення. Після обробки дані можуть бути збережені в базі даних, файлі або виведені у вигляді звіту чи іншого формату [16].

Парсинг використовується в різних областях, таких як веб-розвиток, аналітика даних, маркетинг, економіка та багато інших, де потрібно автоматизовано отримувати, аналізувати та використовувати дані з Інтернету або інших джерел.

Давайте коротко ознайомимся з інструментами для парсингу:

1. Web scraping (веб-скрапінг). Ця технологія забезпечує можливість збирати дані з веб-сторінок шляхом аналізу HTML-коду. Вона дозволяє

отримати доступ до різних видів даних на веб-сторінках, таких як текст, зображення, таблиці тощо. Веб-скрапінг може бути виконаний за допомогою спеціальних бібліотек та інструментів, таких як BeautifulSoup і Scrapy. Веб-скрапінг є потужним інструментом для збору структурованої інформації з веб-сторінок, але його ефективність може залежати від складності розмітки та доступності даних на сторінці.

Парсер даних вирізняється своєю здатністю отримувати інформацію безпосередньо з HTML-коду різних веб-сторінок. Це забезпечує велику гнучкість та універсальність у процесі збору даних, дозволяючи парсеру ефективно взаємодіяти з різноманітними веб-ресурсами [31].

Гнучкість та можливість налаштування парсера для збору різних типів даних (текст, зображення, таблиці тощо) визначають його високий рівень адаптабельності. Це розширює функціональні можливості парсера та дозволяє отримувати різноманітні дані з веб-сайтів.

Ще однією значущою перевагою є широкий охоплюючий спектр джерел даних, які можуть бути використані парсером. Це відкриває безліч можливостей для збору інформації з різних джерел в Інтернеті, надаючи користувачеві широкий вибір джерел для аналізу.

Однак використання парсера даних пов'язане з певними обмеженнями. Зокрема, залежність від структури веб-сторінок може призводити до нестабільності процесу парсингу. Зміни у веб-дизайні джерела інформації можуть впливати на ефективність збору даних та вимагати адаптації парсера.

Додатково, використання парсера вимагає наявності технічних навичок та розуміння HTML/CSS для ефективного використання та налаштування. Отже, користувачам слід мати відповідні технічні знання для успішного впровадження та оптимізації парсера на різних веб-ресурсах [43].

2. API (Application Programming Interface). API надає структурований спосіб взаємодії з певними джерелами даних. Використовуючи API, можна отримувати доступ до специфічних даних без необхідності парсити веб-сторінки. Це дозволяє отримувати структуровані та оновлені дані з веб-серверів. API

дозволяють отримувати доступ до певних наборів даних, але вони можуть бути обмежені функціональністю, доступністю або вартістю.

API відрізняється структурованим підходом, що дозволяє здійснювати отримання даних безпосередньо від джерела, уникаючи необхідності розбору HTML-коду. Цей структурований інтерфейс спрощує процес взаємодії з веб-ресурсами та полегшує збір інформації [41].

API також забезпечує доступ до конкретних наборів даних, які розробники систематично викладають. Це дозволяє користувачам отримувати доступ до точно визначених та структурованих даних, що сприяє більш ефективному використанню інформації.

В порівнянні з веб-скрапінгом, використання API визначається швидкістю та вищою надійністю в отриманні даних. Цей метод дозволяє ефективно взаємодіяти із веб-ресурсами та отримувати інформацію з меншими труднощами.

Однак, існує обмеженість доступу до даних, оскільки не всі веб-ресурси надають API для зовнішнього використання. Це може обмежити можливості користувача з отримання інформації з конкретних джерел.

Також, функціональність та доступність можливостей API можуть бути обмеженими залежно від конкретної реалізації інтерфейсу. Користувачі можуть стикатися з обмеженнями у функціоналі чи можливостях, які надає конкретний API, що може впливати на розширені можливості використання отриманих даних.

3. Data Extraction Tools (інструменти для видобування даних). Ці інструменти надають можливість витягувати дані з різних джерел шляхом налаштування правил та шаблонів. Вони забезпечують автоматизований підхід до парсингу, що полегшує видобування та обробку даних. Такі інструменти, як Octoparse, Import.io та ParseHub, дозволяють використовувати графічний інтерфейс для створення правил і налаштування процесу видобування даних.

Платформи для автоматизованого видобування даних надають готові рішення, що сприяють ефективному та оперативному збору інформації з

різноманітних джерел. Ці інтегровані рішення розроблені для забезпечення швидкості та легкості в процесі видобування даних, вивільняючи користувача від необхідності створення власних алгоритмів парсингу.

Крім того, користувачі мають можливість конфігурувати правила та шаблони для збору та обробки даних, що надає гнучкість у визначенні того, як інформація повинна бути витягнута та використана. Це робить платформи для видобування даних універсальними і відповідними різним потребам користувачів.

Легка інтеграція з іншими інструментами для аналізу та використання даних є однією з ключових переваг цих платформ. Завдяки цьому, користувачі можуть легко передавати отримані дані для подальшого дослідження, аналізу чи використання у своїх інших інструментах та системах [38].

Незважаючи на свою ефективність, платформи для автоматизованого видобування даних мають обмежену гнучкість порівняно з програмованою реалізацією парсингу. Вони можуть не задовольняти потреби користувачів, які вимагають високого рівня налаштування та індивідуалізації у процесі збору даних.

Також, залежність від функціональності та можливостей конкретного інструменту може становити обмеження для користувачів, оскільки їхні потреби можуть виявитися поза рамками можливостей даного рішення.

4. Natural Language Processing (NLP) Tools (інструменти обробки природної мови). Ці інструменти використовуються для аналізу та розуміння текстових даних, включаючи відгуки споживачів. Вони допомагають видобувати ключову інформацію, класифікувати текст, аналізувати настрої та виявляти патерни. Інструменти NLP, такі як Natural Language Toolkit (NLTK), spaCy та Stanford NLP, забезпечують широкі можливості для аналізу текстових даних та екстракції потрібної інформації.

Однією з ключових переваг Natural Language Processing (NLP) є його здатність аналізувати та розуміти текстові дані, включаючи відгуки споживачів та коментарі. Ця характеристика дозволяє використовувати NLP для

ефективного вивчення змісту текстових джерел та отримання інформації, призначеної для аналізу емоцій, настроїв та загального контексту.

Крім того, NLP надає можливість видобувати ключову інформацію, класифікувати текст та аналізувати його настрої, виявляючи патерни у великих обсягах текстових даних. Використання цієї технології дозволяє виконувати складні аналітичні завдання, такі як виявлення семантичних зв'язків та структурування текстової інформації.

Однак існують обмеження, які слід враховувати при використанні NLP. Зокрема, ця технологія фокусується переважно на аналізі тексту, що може призвести до меншої ефективності у випадках, коли дані в іншому форматі.

Також важливо враховувати, що для успішного використання NLP-алгоритмів необхідна грамотна підготовка та обробка даних перед їх застосуванням. Недостатньоочищені або неправильно підготовлені дані можуть впливати на точність та результативність NLP-аналізу.

5. Machine Learning Algorithms (алгоритми машинного навчання). Ці алгоритми можуть бути використані для автоматичного аналізу та обробки даних, включаючи парсинг. Застосування алгоритмів машинного навчання дозволяє автоматично виявляти та класифікувати дані, знаходити патерни та тренди, а також робити передбачення на основі зібраних даних. Алгоритми машинного навчання можуть бути застосовані в різних аспектах парсингу даних для отримання нових інсайтів та зроблення складних аналізів [49].

Автоматизовані системи аналізу даних відзначаються здатністю проводити автоматичний аналіз та обробку великих обсягів інформації, виявлення патернів і здійснення передбачень. Це дозволяє користувачам ефективно використовувати великий обсяг даних, щоб отримувати інсайти та приймати рішення на основі виявлених закономірностей.

Ці системи є ефективним інструментом для створення прогнозних моделей, базуючись на аналізі зібраних даних. Можливість розвивати прогностичні алгоритми дозволяє користувачам отримувати високоякісні передбачення та визначати тенденції в розвитку подій.

Однією з ключових переваг є наявність широкого спектру алгоритмів для різних видів аналізу даних. Відмінною рисою є можливість вибору оптимального методу в залежності від конкретних завдань та характеру даних.

Проте, для ефективного функціонування цих систем важлива наявність великого обсягу якісних даних для навчання моделей. Недостатня якість чи обсяг даних може впливати на точність та достовірність отриманих результатів.

Також слід враховувати, що використання автоматизованих систем аналізу даних вимагає грамотної підготовки даних та настройки параметрів алгоритмів для досягнення оптимальних результатів. Неправильна обробка або налаштування може призвести до неточностей у виведених висновках та прогнозах [50].

Кожна з цих технологій має свої переваги і обмеження, і вибір залежить від конкретних потреб та контексту. Деякі технології можуть бути більш підходящими для конкретних сценаріїв або джерел даних, тому важливо обирати той, який найкраще відповідає вашим потребам. В нашому випадку найзручнішим варіантом буде саме веб-скрапінг, зважаючи на його гнучкість та простоту у використанні.

2.2 Огляд мов програмування та бібліотек

Огляд мов програмування та бібліотек є ключовим етапом в розгляді технологій, що використовуються для написання парсерів. Світ програмування пропонує широкий спектр інструментів, і кожен з них має свої особливості та переваги. У цьому розділі детально розглянемо популярні мови програмування та бібліотеки, що використовуються для створення парсерів, розкриваючи їхні можливості та області застосування. Вивчення цього допоможе визначити оптимальний інструмент для конкретних завдань парсингу та покращить розуміння вибору мови програмування у контексті розробки парсерів [43].

Існує кілька мов програмування, які можуть бути використані для написання парсерів, але Python є однією з найпопулярніших та найефективніших

мов для цієї задачі. Ось кілька мов програмування, які часто використовуються для написання парсерів:

1. Python має багатий екосистему бібліотек та інструментів, що робить його ідеальним вибором для парсингу даних. Бібліотеки, такі як Beautiful Soup і Scrapy, забезпечують потужні функціональні можливості для роботи з HTML-сторінками, XML-даними та іншими форматами. Python має простий і зрозумілий синтаксис, що сприяє швидкому розробленню та легкості розуміння коду [33].

2. JavaScript є мовою, що виконується в браузері і дозволяє взаємодіяти з HTML-сторінками. Використання JavaScript для парсингу даних підходить, коли потрібно витягти дані, які генеруються динамічно за допомогою JavaScript на стороні клієнта. Для цього використовуються бібліотеки, такі як Cheerio та Puppeteer.

3. Ruby є ще однією популярною мовою програмування для парсингу даних. Маючи вбудовану бібліотеку Nokogiri, Ruby дозволяє зручно маніпулювати HTML- та XML-даними. Ruby також має простий і зрозумілий синтаксис, що полегшує розробку парсерів.

4. Java, як мова програмування, надає широкий та потужний інструментарій для створення парсерів. Наявність різноманітних бібліотек полегшує розробку, зокрема, Jsoup і HTMLParser стали важливими інструментами для обробки HTML-даних у Java. Jsoup дозволяє легко парсити та маніпулювати HTML-деревом, а HTMLParser забезпечує широкі можливості роботи з HTML-кодом.

5. В екосистемі Microsoft, мова програмування C# широко використовується для створення парсерів, зокрема, використовуючи потужний інструмент - HtmlAgilityPack. Ця бібліотека дозволяє ефективно взаємодіяти з HTML-кодом, забезпечуючи можливості парсингу та обробки даних з веб-сторінок. HtmlAgilityPack став невід'ємною частиною розробки парсерів в середовищі C#, завдяки своїм зручним інструментам та надійності.

Хоча всі ці мови мають свої переваги та можуть бути використані для парсингу даних, Python вирізняється деякими особливостями, що роблять його найкращим вибором:

1. Python пропонує обширний набір бібліотек для зручного парсингу та обробки даних, включаючи в себе такі інструменти, як Beautiful Soup, Scrapy, Requests, Selenium і численні інші. Це різноманіття бібліотек створює потужний інструментарій для розробників, який дозволяє їм швидко та ефективно здійснювати збір та обробку даних з різноманітних джерел. Завдяки цьому, Python є популярним вибором для завдань, пов'язаних з отриманням та аналізом інформації з веб-сайтів, API та інших джерел даних [29].

2. Python визначається його простим та легким для розуміння синтаксисом, що робить розробку і розгортання коду безпроблемними завданнями. Ця особливість має важливе значення під час роботи зі складними структурами даних, такими як HTML або XML, де зручність написання коду є ключовим фактором для ефективного парсингу та обробки інформації. Використання простого синтаксису Python сприяє швидкому розробленню та легкості у розумінні коду, що робить його ідеальним інструментом для завдань, пов'язаних із збором та обробкою даних з різних джерел [27].

3. Активна та розгалужена спільнота розробників у світі Python виступає ключовим фактором у підтримці та подальшому розвитку проектів, пов'язаних з парсингом даних. Ця спільнота не лише надає важливу підтримку в разі виникнення труднощів, але і забезпечує обширну документацію та готові приклади коду, що великою мірою спрощує роботу для розробників.

Завдяки широкому колу експертів і фахівців, готових допомагати в рішенні питань та надавати консультації, використання Python для розробки парсерів стає ще більш привабливим вибором. Наявність активної спільноти також забезпечує актуальність та вчасність інструментів і технологій, що дозволяє розробникам ефективно використовувати нові можливості та вдосконалення в галузі парсингу даних.

4. Python – універсальна мова програмування, яка володіє кросплатформенністю, що дозволяє використовувати парсери на різних операційних системах, таких як Windows, macOS і Linux. Ця особливість гарантує гнучкість та універсальність в розгортанні парсерів, незалежно від операційної системи, що використовується. Python пропонує консистентну роботу на всіх платформах, що робить його ідеальним інструментом для розробників, які вимагають переносимості та ефективності своїх парсерів.

Загалом, Python є потужним і гнучким інструментом для розробки парсерів даних завдяки своїм багатим бібліотекам, простоті використання та великій спільноті розробників. Він забезпечує ефективну та продуктивну розробку парсерів, що дозволяє збирати та аналізувати дані з різних джерел [24].

Окрім обраної мови програмування, дуже важливим є правильно підібрані бібліотеки. Було розглянуто 10 бібліотек, кожна з яких має свої особливості та переваги, і вибір залежить від конкретних потреб та вимог до проекту.

1. Бібліотека requests є однією з найпопулярніших бібліотек для виконання HTTP-запитів у програмі Python.

Requests надає простий та зрозумілий API для виконання HTTP-запитів, таких як GET, POST, PUT, DELETE і багатьох інших [21]. Вона дозволяє передавати параметри запиту, налаштовувати заголовки, виконувати аутентифікацію, використовувати сесії та куки, обробляти редирект, робити валідацію SSL-сертифікатів і багато іншого. Requests підтримує різні формати відповіді, такі як JSON, XML, текст і бінарні дані. Вона також надає зручні функції для роботи з файлами, завантаження файлів, відправлення форм і роботи з куки.

Бібліотека requests привертає увагу розробників завдяки своєму простому та легкому для розуміння інтерфейсу. Завдяки цьому, виконання HTTP-запитів та обробка отриманих відповідей стають максимально простими та доступними. Ця особливість робить її ідеальним інструментом для широкого спектру завдань, пов'язаних із здійсненням взаємодії з веб-ресурсами.

Бібліотека має ще одну значущу перевагу — високу продуктивність. Її

оптимізований код дозволяє швидко та ефективно обробляти запити, що здорово підходить для проектів з великою кількістю взаємодій з серверами.

Зазначимо також розширені можливості бібліотеки. Робота з сесіями, автоматична обробка редиректів, аутентифікація та робота з куками роблять requests потужним інструментом для взаємодії з веб-серверами. Це важливо для розробників, які потребують більш глибокого та розширеного контролю над процесом взаємодії з веб-ресурсами.

Незважаючи на велику популярність та ефективність, requests має певні обмеження. Відсутність вбудованої підтримки асинхронних запитів може стати обмеженням у деяких вимогливих застосуваннях, де важлива асинхронна обробка багатьох запитів одночасно. Це робить бібліотеку менш підходящою для застосунків, які вимагають ефективної роботи в асинхронному середовищі.

Ще однією слабкою стороною є недостатня підтримка відновлення після помилок. Коли виникає помилка від сервера, requests не має вбудованого механізму автоматичного відновлення запиту, тому доводиться вручну керувати повторним виконанням запиту у разі необхідності.

Загалом, не дивлячись на певні обмеження, бібліотека requests залишається надійним та потужним інструментом для виконання HTTP-запитів у програмі Python, надаючи простоту використання, високу продуктивність та розширені можливості взаємодії з веб-серверами.

2. Бібліотека BeautifulSoup є однією з найпопулярніших бібліотек для парсингу HTML та XML даних у програмі Python [34].

BeautifulSoup дозволяє зручно витягувати дані з HTML-або XML-структур та проводити їх обробку. Вона має простий та зрозумілий інтерфейс, що дозволяє швидко і легко навчитися використовувати бібліотеку.

BeautifulSoup автоматично розпізнає недійсний HTML і вміє коректно парсити сторінки, які містять помилки та несумісність зі стандартами. Вона підтримує різні методи пошуку та навігації по DOM-структурі, такі як пошук за тегами, класами, id, атрибутами тощо [34].

Бібліотека BeautifulSoup вирізняється простотою використання, надаючи розробникам простий та інтуїтивно зрозумілий інтерфейс для парсингу та навігації в HTML або XML документах. Завдяки цьому, робота зі структурою сторінок стає максимально зрозумілою та легкою задачею, спрощуючи витягування потрібних даних.

Більш того, бібліотека вражає гнучкістю. Розширені можливості для пошуку та фільтрації елементів дозволяють точно виділяти необхідну інформацію з веб-сторінок. Це є важливим фактором для ефективної роботи з різними типами веб-сайтів та різноманітними структурами даних.

Також варто відзначити підтримку недійсного HTML, яка є суттєвим плюсом. BeautifulSoup відмінно впорається зі сторінками, що містять неправильний або недійсний HTML-код, автоматично виправляючи помилки та забезпечуючи стабільний та надійний парсинг [34].

Незважаючи на свою ефективність, BeautifulSoup має обмеження. Один із них — швидкодія, особливо при обробці великих обсягів даних. Завдяки своєму підходу, що базується на інтерпретації та аналізі HTML-коду, бібліотека може бути менш швидкою порівняно з деякими іншими альтернативами.

Ще одним недоліком є потреба в додаткових бібліотеках для повного функціоналу. BeautifulSoup часто вимагає встановлення додаткових інструментів, таких як lxml або html5lib, що може викликати додаткові залежності та ускладнювати налаштування робочого середовища. Тим не менше, загалом, бібліотека є потужним та зручним інструментом для парсингу HTML та XML даних, наділяючи розробників зручними функціями для витягування та обробки даних [34].

3. Бібліотека time є вбудованою бібліотекою у Python і надає функції для роботи з часом та паузами.

Бібліотека time надає функції для роботи зі временем, такі як вимірювання проміжків часу, задання затримок, робота зі структурою дати та часу. Вона дозволяє отримати поточний час, виконати паузу у виконанні програми, обчислити різницю між двома часовими мітками і багато іншого. time підтримує

роботу зі структурою дати та часу у різних форматах, що дозволяє зручно оперувати часовими значеннями.

Бібліотека `time` відзначається простотою використання, маючи зрозумілий інтерфейс, що полегшує роботу з часом у програмах Python. Це особливо корисно для розробників, які шукають ефективний інструмент для обробки часових операцій у своїх програмах.

Ще однією значущою перевагою є те, що `time` є вбудованою бібліотекою в Python. Це означає, що вона автоматично присутня у будь-якій стандартній установці Python, не потребує додаткових завдань з встановлення чи управління залежностями.

Різноманітність функцій `time` дозволяє зручно виконувати різні операції, пов'язані зі змінами в часі. Це робить бібліотеку універсальним інструментом для вирішення завдань, пов'язаних із відліком часу та паузами у ваших програмах.

За всією своєю корисністю, бібліотека `time` має деякі обмеження. Наприклад, деякі функції можуть демонструвати питомо низьку точність, особливо при роботі з дуже малими проміжками часу або при точній синхронізації дій відносно системного часу [37].

Також важливо врахувати, що бібліотека `time` базується на системному часі. Це означає, що результати деяких операцій можуть піддаватися впливу системним налаштуванням чи змінами часового поясу. У деяких сценаріях це може призводити до недоліків у використанні, залежно від вимог конкретного проекту.

Загалом, бібліотека `time` залишається корисним інструментом для роботи з часом у програмах Python. Вона пропонує простоту використання та набір функцій для зручної обробки часу в різних сценаріях програмного коду.

4. Бібліотека `json` в Python є дуже корисним інструментом для роботи з форматом даних JSON (JavaScript Object Notation), який широко використовується для обміну даними в мережі [12].

Бібліотека `json` стає надійним інструментом для роботи з JSON-даними завдяки своєму простому та зрозумілому інтерфейсу. Її використання дозволяє

легко перетворювати дані між об'єктами Python та рядками JSON та навпаки. Однак привабливість цієї бібліотеки не обмежується лише простотою використання, адже JSON став стандартним форматом обміну даними в інтернеті. Завдяки цьому, багато веб-сервісів та API повертають дані у форматі JSON, роблячи цю бібліотеку необхідною для багатьох програм [16].

Бібліотека json підтримує роботу з більшістю вбудованих типів даних Python, включаючи списки, словники, рядки, числа та булеві значення. Її крос-платформенність робить її відмінним інструментом для створення та обробки JSON-даних на будь-якій платформі, де використовується Python. Бібліотека також дозволяє розширювати структури даних за допомогою вкладених списків та словників, що полегшує роботу зі складними об'єктами.

Незважаючи на свою ефективність, бібліотека json має кілька обмежень. Перш за все, відсутність підтримки кастомних класів може стати викликом, оскільки їх потрібно вручну перетворювати в структури даних JSON та навпаки [15]. Це може призвести до зайвого коду та складнощів у роботі з складними об'єктами.

Також важливо враховувати обмеження на типи даних, які підтримує JSON. Він обмежується примітивними типами, такими як числа, рядки, списки та словники, і не взаємодіє з більш складними типами, наприклад, з об'єктами типу datetime. Крім того, у випадку великих обсягів даних JSON може стати неефективним через додаткові символи, використовувані для представлення структури.

Важливим питанням також є безпека при роботі з JSON-даними з ненадійних джерел, оскільки це може викликати атаки типу "injection", наприклад, JSON Injection. В таких випадках слід бути особливо обережним при обробці даних.

5. Scrapy - це потужний фреймворк для веб-скрапінгу і парсингу даних. Він надає зручні засоби для структуризації парсерів, взаємодії зі сторінками, обробки даних та збереження результатів.

Бібліотека, яку варто врахувати при парсингу даних, - це динамічний та масштабований інструмент, що вражає своєю здатністю до паралельного парсингу. Це дозволяє ефективно обробляти великі обсяги даних, прискорюючи процес збору інформації. Ще однією його сильною стороною є автоматична навігація по посиланнях, що полегшує витягування даних з різних сторінок в інтернеті. Крім того, він підтримує використання XPath і CSS-селекторів, що додає гнучкість та точність при виборі елементів для парсингу.

Незважаючи на свою потужність, ця бібліотека може виявитися складнішою у використанні, особливо для тих, хто тільки починає занурюватися в світ парсингу даних. Вона вимагає більше коду для налаштування та використання порівняно з простішими бібліотеками. Це може створювати певні бар'єри для входження для новачків, які шукають інтуїтивно зрозумілі рішення. Однак з досвідом та знанням, цей інструмент може стати потужним засобом для складних задач парсингу та обробки даних в Інтернеті.

6. Selenium - це інструмент для автоматизованого тестування веб-додатків. Він дозволяє керувати веб-браузерами, виконувати дії на сторінках, заповнювати форми, натискати кнопки та отримувати вміст сторінок [24].

Виділений серед методів парсингу даних є можливість взаємодії з веб-сторінками, як будь-який звичайний користувач. Це надає можливість отримати дані, які генеруються динамічно через використання JavaScript. Завдяки цій можливості, парсер може отримувати доступ до інформації, яка завантажується або змінюється після завантаження сторінки. Також варто відзначити, що цей метод підтримує взаємодію з різними браузерами, що розширює його придатність для парсингу різноманітних веб-ресурсів.

Однак, незважаючи на свою гнучкість, цей метод може виявитися повільним порівняно з іншими способами парсингу. Завантаження повноцінного веб-браузера для взаємодії зі сторінкою може вимагати значних ресурсів та тривати довше, ніж у випадку інших методів. Крім того, для його використання необхідно мати додаткове програмне забезпечення, таке як веб-браузери та відповідні драйвери, що може створювати додаткові труднощі при використанні

та налаштуванні.

7. PyQuery - це бібліотека Python, яка надає інтерфейс, подібний до jQuery, для роботи з HTML- і XML-документами [21]. Вона дозволяє витягати дані, здійснювати пошук, маніпулювати елементами та виконувати інші дії.

Щодо переваг даного методу, слід відзначити його зручний синтаксис, що робить розробку та розуміння коду простим та легким завданням. Крім того, вбудована підтримка CSS-селекторів робить його особливо зручним для вибору та обробки елементів веб-сторінок. Також варто відзначити можливість використання функцій, схожих на ті, що використовуються в бібліотеці jQuery, що розширює його можливості та спрощує взаємодію з DOM-структурою.

У той час як цей метод може бути зручним та ефективним для простих завдань, його можливості обмежені порівняно з більш розширеними бібліотеками, такими як BeautifulSoup. Для більш складних або специфічних сценаріїв, де потрібна висока гнучкість та розширені функції, цей метод може виявитися менш ефективним [34].

8. Lxml - це бібліотека Python для обробки XML- і HTML-документів. Вона надає швидкий і ефективний парсер засобами C, який підтримує XPath-запити, роботу з елементами, атрибутами, текстом і іншими складовими документа.

Цей метод славиться своєю вражаючою швидкістю та потужністю парсингу, що робить його ідеальним вибором для великих обсягів даних чи завдань, де важлива ефективність. Підтримка XPath-запитів розширює можливості вибору конкретних елементів у документі, а можливість маніпулювати документом надає гнучкість у взаємодії з отриманими даними.

Незважаючи на його потужність, цей метод може бути складнішим у використанні порівняно з деякими іншими бібліотеками. Це може вимагати від розробників додаткових зусиль та часу для оволодіння його функціоналом. Також слід відзначити, що для роботи з HTML необхідно провести додаткові налаштування, що може створити певні труднощі для тих, хто шукає більш прості рішення.

9. Scrapy-Selector - це частина фреймворка Scrapy і використовується для вибору і витягування даних зі сторінок. Вона надає інтерфейс для використання CSS-селекторів, XPath-виразів та регулярних виразів для вибору елементів і витягування даних.

Цей метод вирізняється своєю простотою інтеграції з фреймворком Scrapy, створюючи ефективне середовище для парсингу. Додатково, підтримка CSS-селекторів і XPath дозволяє легко вибирати та обробляти необхідні елементи на веб-сторінках. Можливість використовувати регулярні вирази розширює гнучкість методу та дозволяє виконувати точний пошук в текстових даних.

Однак, для повноцінного використання цього методу вам потрібно виконати налаштування фреймворка Scrapy, що може вимагати додаткового часу та ресурсів. Це може бути завданням, особливо для тих, хто шукає прості та швидкі рішення без необхідності у використанні додаткових інструментів чи фреймворків.

10. Puppeteer - це Python-порт Puppeteer, бібліотеки Node.js для керування безпосередньо веб-браузерами Chrome або Chromium. Вона дозволяє виконувати скрипти на Python для автоматизованого взаємодії з веб-сторінками, заповнення форм, натискання кнопок та інших дій [22].

Метод визначається своєю унікальною здатністю імітувати взаємодію з веб-сторінками, ніби це робив користувач, завдяки вбудованій підтримці JavaScript. Однією з ключових переваг є автоматизоване керування браузером, що розширює можливості виконання різноманітних завдань.

Однак для використання цього методу необхідно мати встановлені Chrome або Chromium, що може бути обтяжливим для деяких користувачів. Додатково, через використання реального браузера, процес парсингу може бути менш ефективним у випадку великої кількості запитів, що може призвести до сповільнення процесу отримання даних.

Серед десяти перерахованих популярних бібліотек для парсингу, було обрано п'ять, а саме: requests, BeautifulSoup, time, lxml та json.

Бібліотека `requests` використовувалась для парсингу даних з веб-сторінок та для отримання HTML-коду сторінки, а `BeautifulSoup` допомагала структурувати та витягувати необхідні дані з цього коду за допомогою різних методів та CSS-селекторів [34].

Бібліотека `time` допомагала зменшити навантаження на сервер веб-ресурсу, періодично призупиняючи роботу додатка. Це зумовлено тим, що про звичайному серфінгу, людина фізично не в змозі виконати таку кількість запитів, яку виконує додаток, що може бути причиною блокування запитів на веб-сторінку, і в результаті збою роботи додатка. `Lxml` використовувався як аналізатор HTML-коду. Поміж інших бібліотек його вирізняє швидкодія, що і стало причиною його використання. `Json` використовувався для того щоб структурувати отримані дані та записати їх в файл формату `json`.

IDE (Integrated Development Environment) - це комплексне середовище розробки, яке надає зручну робочу область для програміста для написання, відлагодження та тестування програмного коду. IDE зазвичай поєднує в собі текстовий редактор з підсвічуванням синтаксису, інструменти для автодоповнення коду, систему керування версіями, візуальні компоненти для створення інтерфейсу користувача та інші допоміжні функції.

1. `Thonny` представляє собою легкий у використанні та інтуїтивно зрозумілий інтегрований середовище розробки (IDE) для мови програмування `Python`, спеціально адаптоване для новачків у програмуванні. Його простий та доступний інтерфейс спрощує процес написання та відлагодження коду, сприяючи комфортному вивченню основ `Python`.

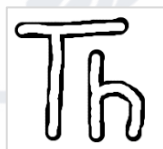


Рисунок 2.1 – Логотип `Thonny`

Thonny визначається вбудованою підтримкою автодоповнення, що дозволяє швидко та ефективно розробляти код. Це особливо корисно для початківців, що тільки ознайомлюються з мовою програмування.

Поміж його ключових переваг варто відзначити інтерактивну консоль, де можна поетапно виконувати код та спостерігати за його роботою. Це сприяє кращому розумінню основ Python та дозволяє вчитися, впроваджуючи новий код.

Крім того, Thonny обладнаний вбудованими інструментами для відлагодження, такими як крокове виконання коду, відстеження значень змінних та відображення стеку викликів функцій, що полегшує процес виправлення помилок та вдосконалення програм.

2. PyCharm визнаний як одна з найпопулярніших та високофункціональних інтегрованих середовищ розробки (IDE) для мови програмування Python. Завдяки своєму широкому спектру функцій, PyCharm став вибором багатьох розробників.



Рисунок 2.2 – Логотип PyCharm

Однією з ключових переваг є великий набір інструментів, таких як автодоповнення, аналіз коду, відлагодження та система керування версіями. Редактор коду в PyCharm вражає своєю потужністю, надаючи функції підсвічування синтаксису, можливості перетягування та впорядкування коду, а також гнучкі налаштування розширень та плагінів.

Крім того, PyCharm включає інструменти для автоматичного тестування, рефакторингу коду та розробки веб-додатків на Python. Такий інтегрований підхід робить його універсальним інструментом для розробників, які працюють з проектами різної складності та напрямку.

3. Visual Studio Code є високопопулярним та легким редактором коду, який відзначається активною підтримкою мови програмування Python завдяки великому спектру розширень. Його використовують розробники різних рівнів, від початківців до досвідчених професіоналів.



Рисунок 2.3 – Логотип Visual Studio Code

VS Code відзначається рядом корисних функцій, включаючи автодоповнення, відлагодження, керування проектами та інтеграцію з різними системами керування версіями. Його інтерфейс є зручним та налаштованим, дозволяючи розробникам адаптувати редактор коду під свої власні потреби.

Однією з ключових переваг є широка підтримка спільної роботи, особливо завдяки можливості використання розширень для спільної розробки проектів Python. Це робить VS Code популярним вибором для команд розробників, які працюють над спільними проектами та цінують зручність і продуктивність цього редактора.

4. Jupyter Notebook визначається як інтерактивне середовище для виконання коду Python та створення комплексних документів, які об'єднують код, текстові відомості та графічні представлення.



Рисунок 2.4 – Логотип Jupyter Notebook

Однією з ключових переваг є можливість виконувати код крок за кроком, дозволяючи спостерігати за результатами на кожному етапі. Це відмінно підходить для проведення експериментів та візуалізації даних, що зробило Jupyter Notebook незамінним інструментом у наукових та дослідницьких областях.

Також варто зазначити, що Jupyter Notebook підтримує форматування Markdown, що дає змогу вставляти пояснювальні тексти, формули, графіки та інші елементи безпосередньо у ноутбук. Це робить його ефективним інструментом для розробки та представлення документів, звітів та освітніх матеріалів.

5. Spyder представляє собою спеціалізоване наукове середовище для програмування на мові Python, зорієнтоване на аналіз даних, наукові обчислення та візуалізацію.



Рисунок 2.5 – Логотип Spyder

Однією з ключових особливостей є потужний редактор коду, який включає в себе функції підсвічування синтаксису, автодоповнення та інтеграцію з IPython. Це надає розробникам зручний інструментарій для написання коду, зокрема при аналізі та опрацюванні даних.

Серед інших переваг Spyder можна відзначити наявність інтегрованої консолі IPython, яка дозволяє виконувати код по чергово та проводити експерименти з даними в режимі реального часу. Додатково, Spyder пропонує інструменти для відлагодження, управління пакетами та підтримки віртуальних середовищ, що робить його комплексним інструментом для розробки в галузі аналізу даних.

Кожна з цих IDE має свої переваги та підходить для різних потреб. Вибір Thonny як найкращого варіанту обумовлюється його простотою використання,

спрямованістю на початківців та наявністю інструментів для навчання та вивчення основ Python.

2.3 Створення моделі парсеру для збору даних характеристик товару

У цьому розділі розглянемо модель для інформаційної технології, яка використовується для аналізу даних та оцінки споживчої вартості товару. Модель є ключовим елементом системи, оскільки вона дозволяє автоматизувати аналітичний процес та отримувати об'єктивні результати [18].

Модель – це концептуальне відображення системи, що включає в себе її складові, функції та взаємодії, яке визначає її структурні та функціональні аспекти. В різних галузях моделі застосовуються для проведення аналізу, передбачення майбутніх подій та прийняття обґрунтованих рішень. Ці абстракції можуть бути математичними, комп'ютерними, чи іншими формами представлення, і вони допомагають в наукових дослідженнях, інженерії, економіці та інших галузях для розв'язання завдань та вивчення складних систем.

Так як основним елементом системи є парсер, розглянемо детальніше модель роботи парсера для збору інформації про ноутбуки (рис. 2.6).



Рисунок 2.6 – Модель роботи парсера

Розглянемо детальніше етапи:

1. Перший крок – отримання вихідного HTML-коду веб-сторінки. Це може бути виконано шляхом відправлення HTTP-запиту до сервера та отримання відповіді, яка містить HTML-код сторінки.

2. Парсер аналізує HTML-код та використовує методи парсингу, такі як DOM-парсинг або використання регулярних виразів, для виділення необхідних елементів сторінки, таких як текст, посилання, зображення та інші.

3. Після розбору HTML-коду парсер визначає конкретні дані, які йому потрібно виділити. Це може включати заголовки, текст, ціни товарів, характеристики тощо [23].

4. Отримані дані організуються у структуру, таку як масиви, об'єкти або таблиці, для зручного подальшого використання.

5. Парсер може зберігати отримані дані у визначеному форматі або базі даних для подальшого використання чи аналізу.

6. Парсер повинен бути обладнаний механізмами для обробки можливих помилок чи винятків, таких як зміни у розмітці веб-сайту, недоступність даних тощо.

7. У випадку, якщо парсер використовується для регулярного оновлення даних, він може працювати у циклі, автоматично оновлюючи інформацію за певний період часу [26].

Така модель роботи парсера дозволяє систематично збирати та оновлювати інформацію про ноутбуки з різних джерел для подальшого аналізу або використання в інших інформаційних системах.

ВИСНОВКИ ДО 2 РОЗДІЛУ

У цьому розділі були описані різні аспекти розробки програмного забезпечення, включаючи середовища розробки, мови програмування, бібліотеки та інструменти.

Перш ніж розпочати розробку програмного забезпечення, важливо вибрати підходяще середовище розробки. У цьому розділі було розглянуто декілька популярних IDE. Кожне з цих середовищ має свої переваги та підходить для різних потреб розробки.

Також було описано мову програмування Python, яка є однією з найпопулярніших мов для розробки програмного забезпечення. Python має чистий синтаксис, велику кількість бібліотек та широкі можливості застосування в різних галузях. Крім того, були представлені деякі популярні бібліотеки та інструменти для розробки, такі як requests, BeautifulSoup, time, lxml та json. Ці бібліотеки дозволяють зручно виконувати різні завдання, включаючи парсинг даних, роботу з мережевими запитами та створення json-файлів.

Основним елементом системи є парсер. Парсер виконує кілька ключових етапів, починаючи від отримання HTML-коду веб-сторінки і закінчуючи зберіганням структурованих даних для подальшого використання або аналізу. На кожному етапі важливо враховувати можливість помилок та змін у розмітці веб-сайтів. Модель роботи парсера надає систематичний підхід до збору та оновлення інформації про ноутбуки. Це не лише полегшує аналіз, але й створює базу для ефективного використання отриманих даних у подальших дослідженнях чи інших інформаційних системах. Інформаційна технологія, така як ця, відкриває нові можливості для розвитку та вдосконалення сучасних методів аналізу даних та прийняття рішень в індустріальному та бізнес-середовищі.

Усі ці аспекти розробки взаємодіють між собою, створюючи унікальне середовище для кожного проекту. Вибір середовища розробки, мови програмування та інструментів повинен бути здійснений з урахуванням конкретних потреб проекту та власних уподобань розробника.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ДОДАТКУ ДЛЯ ОЦІНКИ СПОЖИВЧОЇ ЦІННОСТІ ТОВАРУ

3.1 Процес збору даних для оцінки споживчої цінності товару

Додаток для збору даних виконує важливу функцію — він реалізує парсинг даних із інтернет-магазину. Зокрема, цей код спроектований для витягування інформації із сторінок, що містять детальні дані про ноутбуки та їх характеристики. Процес парсингу включає в себе аналіз HTML-коду сторінок, визначення відповідних тегів та елементів, а також збирання цієї інформації для подальшого використання [27].

Завдяки цьому програмному рішення ми можемо отримати доступ до важливих даних про різні моделі ноутбуків, таких як їхні технічні характеристики, ціни та наявність на складі. Це відкриває безліч можливостей для аналізу ринку, порівняння пропозицій від різних виробників та визначення оптимальних варіантів для споживачів.

Важливим елементом цього коду є його гнучкість та можливість адаптації до змін у структурі веб-сайту магазину. Це забезпечує стійку та ефективну роботу додатка навіть при оновленнях чи модифікаціях у вигляді сайту. Код також враховує можливість обробки помилок та винятків, щоб забезпечити надійність та стабільність його функціонування.

Таким чином, розроблений мною код відіграє важливу роль у забезпеченні доступу до актуальної інформації про ноутбуки, сприяючи ефективному вибору та придбанню цих пристроїв для кінцевих користувачів.

Першочергово потрібно імпортувати необхідні модулі, щоб забезпечити ефективну роботу нашого додатка. Використовуючи бібліотеки `requests`, `BeautifulSoup` (псевдонім `BS`), `json` та `sleep` з модуля `time`, ми отримуємо доступ до потужного інструментарію для взаємодії з веб-ресурсами та обробки інформації.

У нашому кодї визначається заголовок `headers`, який включає в себе інформацію про користувацький агент (`User-Agent`). Цей користувацький агент є частиною HTTP-запитів і визначається, яким чином веб-сайт сприймає запити від клієнта. У нашому випадку, ми використовуємо конкретний `User-Agent` для нашого HTTP-запиту, що зображено на рисунку 3.1

```
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0"}

```

Рисунок 3.1 – Визначення інформації про користувацький агент

Цей `User-Agent` імітує взаємодію з веб-сайтом через браузер Firefox версії 47.0, який працює на операційній системі Windows. Використання такого `User-Agent` допомагає нам зробити наші HTTP-запити подібними до тих, що відправляються реальним браузером. Це важливо для того, щоб уникнути блокування або обмежень з боку веб-сайту, які можуть виникнути при виявленні неякісних або автоматизованих запитів.

Отже, використання конкретного `User-Agent` у заголовку `headers` допомагає нам створити враження, ніби наші запити генеруються реальним веб-браузером, що є важливим аспектом в роботі з веб-скрапінгом та парсингом даних.

У нашому програмному рішенні функція `get_url()` відіграє ключову роль у здійсненні отримання URL-адрес сторінок із списком ноутбуків з каталогу інтернет-магазину. Ця функція створена з метою автоматизації процесу збору даних та надає нам зручний спосіб отримати важливу інформацію.

На початку функції визначається цикл `for`, який ітерується від 1 до 22 (включно), що дорівнює кількості сторінок в інтернет магазині. Кожна ітерація цього циклу відповідає за створення URL-адреси для сторінки із списком характеристик гаджету. Це важливо, оскільки велика кількість сторінок може містити різні моделі ноутбуків та їх характеристики. Створені URL-адреси далі використовуються для виклику `requests.get()` разом із заданими заголовками.

Такий підхід дозволяє нам ефективно взаємодіяти із сервером веб-сайту та отримувати необхідні сторінки.

Отримані результати представляють собою HTML-сторінки, які в подальшій частині коду парсяться за допомогою BeautifulSoup. Це забезпечує нам зручний доступ до структурованої інформації, що міститься на кожній сторінці. Після цього, використовуючи генератор `yield`, інформація про кожен ноутбук у вигляді URL-адреси повертається, надаючи нам ітерабельний об'єкт для подальшої обробки.

Ця функція є ключовою ланкою у забезпеченні системи збору даних із магазину та дозволяє нам ефективно ітерувати через різні сторінки, збираючи важливі відомості про ноутбуки для подальшого аналізу та використання.

Функція `aggau()` відіграє критичну роль у зборі детальної інформації про кожен ноутбук, надаючи нам комплексний огляд характеристик пристроїв. Ця функція починає свою роботу, викликаючи функцію `get_url()`, яка забезпечує перелік URL-адрес сторінок із списками ноутбуків для подальшого парсингу.

Для кожного URL-адреси, отриманого від `get_url()`, виконується HTTP-запит, а отримана сторінка піддається парсингу за допомогою бібліотеки BeautifulSoup. Цей етап дозволяє ефективно аналізувати структуру HTML-коду та витягувати потрібні дані для подальшого використання.

З використанням методів `find()` та `find_all()` витягуються різні параметри, які докладно описують кожен ноутбук. Назва (`name`), ціна (`price`), модельний ряд (`laptop_model`), операційна система (`laptop_os`), діагональ дисплея (`display_diagonal`), тип матриці дисплея (`display_matrix_type`), розширення екрану (`display_extension`), частота оновлення дисплея (`display_frequency`), серія процесора (`processor_series`), модель процесора (`processor_model`), базова частота процесора (`processor_frequency`), кількість ядер процесора (`processor_core`), об'єм оперативної пам'яті (`ram_size`), тип оперативної пам'яті (`ram_type`), частота оперативної пам'яті (`ram_frequency`), тип накопичувача (`drive_type`), об'єм накопичувача (`drive_size`), тип відеокарти (`gpu_type`), виробник відеокарти

(`gpu_producer`), модель відеокарти (`gpu_model`) та об'єм відеокарти (`gpu_size`) — це параметри, які збираються для створення повної картини про кожен пристрій.

Після витягування цих даних інформація повертається за допомогою генератора `yield`, що дозволяє створити ітерабельний об'єкт, який можна легко переглядати та використовувати у різних частинах програми.

Ця функція є важливим кроком у забезпеченні повноцінного збору та обробки інформації про ноутбуки, виготовляючи дані у вигляді, що легко аналізується та використовується для подальших завдань, таких як аналіз ринку чи порівняння моделей.

У нашому програмному коді слід розглядати важливу деталь – коментарі, які вказують на кожен частину функціоналу. Ця практика не лише робить код зрозумілим для інших розробників, але і надає ясність та легкість в розумінні для тих, хто працює з кодом у майбутньому.

Вся наша програма спроектована для ефективної взаємодії з інтернет-магазином, а сам код виконує послідовний ряд операцій. Він відправляє HTTP-запити до веб-сайту, парсить отримані сторінки та вилучає необхідні дані про ноутбуки. Цей метод дозволяє отримати структуровану інформацію про ноутбуки, що подальше використовується для різноманітних завдань, таких як аналіз, обробка чи збереження в певному форматі.

Окрім функціональності, коментарі також розкривають стратегії та підходи, використані для обробки та аналізу отриманих даних. Кожен крок має відповідний коментар, що полегшує розуміння логіки роботи коду.

Останній етап – це збереження отриманих даних. За допомогою функції `array()`, дані збираються у вигляді структурованого ітерабельного об'єкту, який потім можна записати в файл JSON. Використання бібліотеки `Json` дозволяє зручно та ефективно перетворити структуровані дані у формат JSON, що ідеально підходить для подальших операцій з обробки або обміну інформацією з іншими системами.

Загалом, цей код втілює принципи чистого та структурованого програмування, що сприяє його ефективному використанню та розширенню.

3.2 Аналіз даних споживчої цінності товару

Результатом успішного виконання додатка є JSON-файл, який представляє собою збережені дані про ноутбуки. Для отримання більш глибокого розуміння цих даних, проведемо детальний аналіз. Почнемо з загального огляду структури таблиці та розглянемо її вміст для кращого усвідомлення інформації, що міститься в цьому JSON-файлі.

У файлі ми спостерігаємо структуру "списку словників" або "масиву об'єктів". Кожен елемент цього списку представляє окремий ноутбук та містить низку ключів, які визначають різні характеристики пристрою.

У першому рядку нашої таблиці, який служить введенням у світ даних про ноутбуки у JSON-файлі, виблискує розкішна картина характеристик кожного пристрою. Цінність кожного стовпця не може бути переоцінена, оскільки вони визначають важливі параметри, які визначають ноутбук і роблять його унікальним.

На першому рядку виділяються назви стовпців, які є ключами до вивчення всієї інформації про кожен ноутбук. Отже, давайте розглянемо ці стовпці детальніше:

1. Назва (Name): Визначає ім'я чи модель ноутбука.
2. Ціна (Price): Показує вартість ноутбука, важливий показник при виборі.
3. Модель ноутбука (Laptop Model): Вказує конкретну модель пристрою.
4. Операційна система (Laptop OS): Визначає операційну систему, попередньо встановлену на ноутбук.
5. Діагональ дисплея (Display Diagonal): Вказує розмір дисплея в дюймах.
6. Розширення дисплея (Display Resolution): Визначає кількість пікселів на екрані.
7. Тип матриці (Display Matrix Type): Показує вид матриці екрану, наприклад, IPS.

8. Частота оновлення дисплея (Display Refresh Rate): Визначає, як часто зображення оновлюється на екрані.
9. Серія процесора (Processor Series): Вказує на принципову лінію процесорів.
10. Модель процесора (Processor Model): Визначає конкретну модель процесора.
11. Частота процесора (Processor Frequency): Показує робочу частоту процесора в гігагерцах.
12. Кількість ядер процесора (Processor Cores): Визначає, скільки ядер в процесорі.
13. Об'єм оперативної пам'яті (RAM Size): Показує об'єм вбудованої оперативної пам'яті.
14. Тип оперативної пам'яті (RAM Type): Вказує на технологію оперативної пам'яті, наприклад, DDR4.
15. Частота оперативної пам'яті (RAM Frequency): Визначає частоту роботи оперативної пам'яті.
16. Тип накопичувача (Drive Type): Показує, чи це SSD, HDD або інший тип накопичувача.
17. Об'єм накопичувача (Drive Size): Визначає об'єм зберігання даних на накопичувачі.
18. Тип відеокарти (GPU Type): Показує, чи це вбудована чи дискретна відеокарта.
19. Виробник відеокарти (GPU Producer): Вказує на компанію-виробника відеочіпа.
20. Модель відеокарти (GPU Model): Показує конкретну модель відеокарти.
21. Об'єм пам'яті відеокарти (GPU Size): Визначає об'єм відеопам'яті.

Ці назви стовпців відповідають відповідним полям у наших даних, які були збережені під час парсингу веб-сторінок інтернет-магазину (рис. 3.2). Кожен із

цих параметрів визначає індивідуальні риси кожного ноутбука, надаючи повний обсяг інформації для здійснення обдуманих та відповідальних виборів.

```
1 {
2   "laptops": [
3     {
4       "name": "ASUS TUF GAMING F15 FX506HC-HN057 Graphite Black (90NR0724-M00X10)",
5       "price": 32999,
6       "laptop_model": "Asus TUF",
7       "laptop_os": "DOS",
8       "display_diagonal": 15.6,
9       "display_extension": "1920 x 1080",
10      "display_matrix_type": "IPS",
11      "display_frequency": 144,
12      "processor_series": "Intel® Core™ i5 (11 покоління)",
13      "processor_model": "11400H",
14      "processor_frequency": 2.7,
15      "processor_core": 6,
16      "ram_size": 16,
17      "ram_type": "DDR4",
18      "ram_frequency": 3200,
19      "drive_type": "SSD",
20      "drive_size": 512,
21      "gpu_type": "дискретна",
22      "gpu_producer": "NVIDIA",
23      "gpu_model": "GeForce RTX 3050",
24      "gpu_size": 4
25    },
26  ],
27 }
```

Рисунок 3.2 – Приклад збереження даних

У випадку коли ж певні з параметрів відсутні у конкретній моделі, замість того щоб передавати до датасету значення NULL, вони замінюються на «пусте значення» (рис. 3.3), при обчисленні ж такі значення будуть замінюватися на 0, або ж не будуть враховуватися взагалі.

```
49 {
50   "name": "Lenovo ideapad Slim 3 15IAN8 Arctic Grey (82XB002LRA)",
51   "price": 17999,
52   "laptop_model": "Lenovo IdeaPad",
53   "laptop_os": "DOS",
54   "display_diagonal": 15.6,
55   "display_extension": "1920 x 1080",
56   "display_matrix_type": "IPS",
57   "display_frequency": 60,
58   "processor_series": "Intel® Core™ i3 (13 покоління)",
59   "processor_model": "N305",
60   "processor_frequency": 1.8,
61   "processor_core": 8,
62   "ram_size": 8,
63   "ram_type": "LPDDR5",
64   "ram_frequency": 4800,
65   "drive_type": "SSD",
66   "drive_size": 512,
67   "gpu_type": "інтегрована",
68   "gpu_producer": [],
69   "gpu_model": [],
70   "gpu_size": []
71 }
```

Рисунок 3.3 – Приклад збереження відсутніх параметрів

В аналізованій таблиці ми маємо доступ до даних про 889 різноманітних ноутбуків. Кожен рядок цієї таблиці є вірним відображенням окремого гаджету, відтак, сума всіх рядків відображає точну кількість ноутбуків, які були зібрані завдяки парсингу даних.

Деякі з них вміщують кількісні показники, такі як вартість ноутбука, кількість ядер у процесорі, обсяг оперативної пам'яті та інші. Ці дані не лише числові, а й насичені кількісним контекстом, що робить їх вельми інформативними для кінцевого користувача.

Ще однією важливою частиною цього датасету є якісна інформація. Клітинки, які вміщують назви ноутбуків, серії компонентів, виробників та інші характеристики, не тільки завантажені текстом, але й несуть в собі характер та унікальність кожного з пристроїв. Ця якісна інформація є ключем для розуміння контексту і застосування даних у практичних сценаріях. Дані відображаються відповідно до установлених форматів, що сприяє зручній інтерпретації та використанню накопиченої інформації.

Отримані дані відкривають різні можливості в їхньому застосуванні. Зокрема, ця база інформації про ноутбуки може служити ключовим інструментом для проведення різноманітних аналізів та вивчення специфічних аспектів ринку електроніки. Величезний обсяг даних відкриває можливості для глибокого розуміння і використання в різних галузях. Нижче розглянемо деякі конкретні сфери застосування цих даних:

1. Аналіз споживчої цінності. Отримані дані про ноутбуки надають змогу провести детальний аналіз їхньої споживчої цінності. Залежно від конкретних вимог та вподобань споживачів, можна визначити, які характеристики найбільше впливають на ціну ноутбуків, і як можна оптимізувати цю вартість для кінцевих користувачів.

2. Порівняння цін та інших характеристик. За допомогою цих даних можна провести широке порівняння цін та інших ключових характеристик між різними компаніями, брендами чи моделями ноутбуків. Це дозволяє споживачам зробити інформований вибір, враховуючи різноманіття пропозицій на ринку.

3. Розрахунок середньої ціни. Інтерес до середньої ціни ноутбуків може бути задоволений завдяки обробці отриманих даних. Це дозволяє отримати узагальнену картину цінової динаміки на ринку та розуміти, як різні фактори впливають на ціну в цілому.

4. Подальший аналіз та обробка. Отримані дані, які вже є цінною інформацією, можуть служити як вихідна точка для подальших аналітичних операцій та обробки. Використовуючи різні методи аналізу даних, можна виявити тенденції, розуміти споживчі уподобання та забезпечити корисний контекст для бізнес-рішень.

Узагальнюючи, отримані дані від інтернет-магазину, відображені у таблиці, становлять не лише інформацію про конкретні ноутбуки, але й відкривають нові можливості для стратегічного розуміння ринку електроніки, її динаміки та споживчих уподобань.

Аналіз отриманих даних відбувається шляхом обчислення комплексного критерію, що будується на співставленні обраних показників в сукупності агрегованого індексу.

Для обрахування були обрані 4 основні показники, такі як: ціна, частота процесора, кількість оперативної пам'яті та об'єм пам'яті дискретної відеокарти, оскільки пам'ять інтегрованої відеокарти напряду залежить від кількості оперативної пам'яті. Тож можна сказати що ці критерії були обрані для того щоб вирахувати співвідношення ціни та продуктивності.

Комплексний критерій є рейтингом, що повинен надати загальне уявлення про ринок лептопів з урахуванням частки кожного об'єкта в сукупному визначенні показника.

Комплексний критерій будується з урахуванням нормалізованих значень часткових критеріїв та їх вагових коефіцієнтів. Часткові критерії будуються у відповідності до основних категорій оцінювання, вагові коефіцієнти визначаються на основі оцінки багатофакторного зв'язку (лінійних коефіцієнтів кореляції та кореляційної матриці). Комплексний критерій розраховується за формулою:

$$I = \frac{\sum_{i=1}^N \sum_{j=1}^M R_{Hij} \times \lambda_j}{\sum_{j=1}^M \lambda_j} \quad (3.1)$$

де I - значення споживчої цінності товару;

R_{Hij} - нормалізоване значення j -го критерію оцінювання, що обчислюється по i -му гаджету за формулою:

$$R_H = \frac{(R - R_{min})}{(R_{max} - R_{min})} \quad (3.2)$$

Ця формула працює таким чином, що чим більше R , тим більше буде нормалізоване значення. Тож більші R вважаються "кращими" або "більш бажаними", оскільки вони призводять до більших нормалізованих значень. Це правило працює з усіма обраними показниками, окрім показника ціни. Це пов'язано з тим, що зі сторони споживача, його цікавить як за найменшу суму отримати найпотужніший гаджет, тому в цьому випадку при нормалізації значень ціни буде використовуватися інша формула:

$$R_H = \frac{(R_{max} - R)}{(R_{max} - R_{min})} \quad (3.3)$$

де R - поточне значення оціночного показника;

R_{max} - максимальне значення оціночного показника;

R_{min} - мінімальне значення оціночного показника;

λ_j - ваговий коефіцієнт відповідного j -го критерію оцінювання;

t - поточний період оцінювання (1 рік);

j - змінна, яка характеризує множину кількості часткових критеріїв, що утворюють комплексний критерій ($j \in [1, M]$, $M=4$);

i - змінна, яка характеризує множину гаджетів, що підлягають рейтинговій оцінці ($i \in [1, N]$, $N=889$);

Процедура нормалізації передбачає приведення всіх часткових критеріїв до одного масштабу з урахуванням принципу максимальної ефективності. При цьому вагові коефіцієнти враховують рівень тісноти між факторами по всій кореляційній матриці, що утворює вибірка:

$$\lambda_j = \frac{\sum_{i=1}^m r_{ij} - 1}{m - 1} \quad (3.4)$$

де λ_j – ваговий коефіцієнт відповідного j -го критерію оцінювання;

j – змінна, яка характеризує множину кількості часткових критеріїв, що утворюють комплексний критерій ($j \in [1, M]$, $M=4$);

r_{ij} - лінійний коефіцієнт кореляції між i -тим та j -тим частковими критеріями.

Процес нормалізації даних та розрахування їх вагових коефіцієнтів є автоматизованим, та в результаті були отримані наступні значення, що зображені на рисунку 3.4.

num1: 0.30141456718903553	ціна
num2: 1.4125396491147988	частота процесора
num3: 0.8456777730663017	ОЗУ
num4: 1.2927899691465412	пам'ять відеокарти

Рисунок 3.4 – Вагові коефіцієнти

З отриманих даних, можна зробити висновок, що найбільшу вагу мають пам'ять відеокарти та частота процесора. Це є логічним значенням, адже процесор та відеокарта є одними з основних та найдорожчих складових, через що на пряму впливають на ціну девайса.

Після проведення оцінки споживчої вартості товару для 889 ноутбуків, гаджетом з найбільшим рейтингом став GIGABYTE AERO 16 OLED Silver. Його ціна складає 106999 гривень і його характеристики зображені на рисунку 3.5.

```

15275 {
15276   "name": "GIGABYTE AERO 16 OLED Silver (BSF-A3KZ964SQ)",
15277   "price": 106999,
15278   "laptop_model": "Gigabyte AERO",
15279   "laptop_os": "Windows 11",
15280   "display_diagonal": 16.0,
15281   "display_extension": "3840 x 2400",
15282   "display_matrix_type": "IPS",
15283   "display_frequency": 60,
15284   "processor_series": "Intel® Core™ i7 (13 покоління)",
15285   "processor_model": "13900H",
15286   "processor_frequency": 4.1,
15287   "processor_core": 14,
15288   "ram_size": 32,
15289   "ram_type": "LPDDR5",
15290   "ram_frequency": [],
15291   "drive_type": "SSD",
15292   "drive_size": 1024,
15293   "gpu_type": "дискретна",
15294   "gpu_producer": "NVIDIA",
15295   "gpu_model": "GeForce RTX4070",
15296   "gpu_size": 8
15297 },

```

Рисунок 3.5 – Характеристики ноутбука GIGABYTE AERO 16 OLED Silver

Ціна даного ноутбука представляє собою вигідну пропозицію, яка залишається у прийнятних межах, особливо у порівнянні з середніми цінами на ноутбуки в цілому. Загалом, середня вартість ноутбуків складає 52550.34 грн. Докладніше, якщо розглядати підкатегорії ноутбуків, то середні ціни становлять 37800.30 грн. для ноутбуків із інтегрованою відеокартою та 80981.58 грн. для ноутбуків із дискретною відеокартою, як це відображено на рисунку 3.6.

Ці контекстні середні значення стають важливим фактором при оцінці конкретної цінової політики даного ноутбука. Варто відзначити, що, не дивлячись на вигідність ціни обраного пристрою, ринок ноутбуків може варіюватися, і це важливо враховувати при порівнянні з загальною динамікою цін.

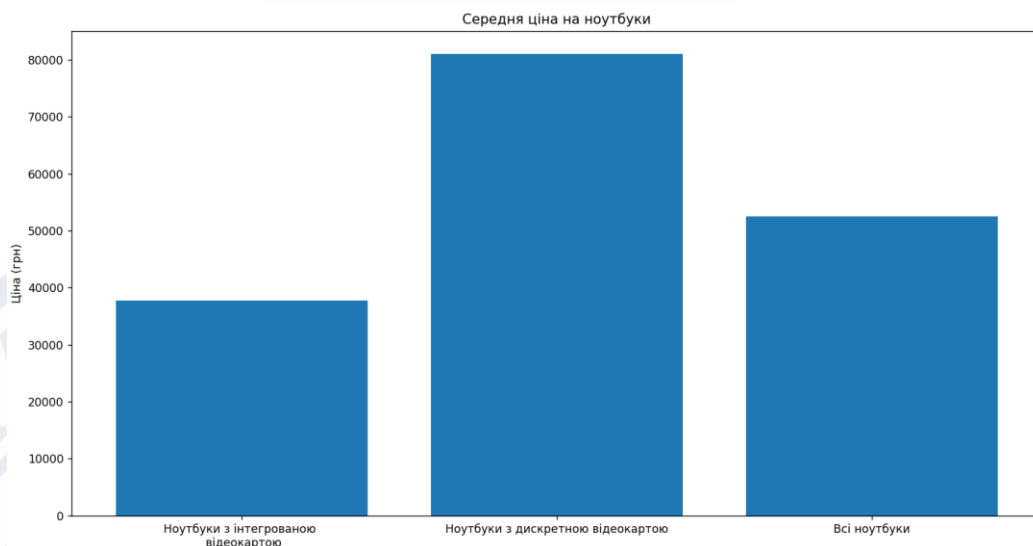


Рисунок 3.6 – Графік середніх цін на ноутбуки з дискретною та інтегрованою відеокартою

Максимальна ціна на ноутбук в даному асортименті становить 229999 грн., тоді як мінімальна ціна складає всього 6999 грн. Цей широкий діапазон вартості свідчить про різноманітність ринку та різні стратегії ціноутворення виробників ноутбуків.

Зобразивши графічно зростання показника споживчої вартості (рис. 3.7), можна помітити, що більша частина навіть не перетнула значення в 0.4. Це напряму пов'язано з кількістю гаджетів з дискретною відеокартою. Їх кількість складає 300 штук з 889, в свою чергу ноутбуків з інтегрованою відеокартою всього є в кількості 589 штук.

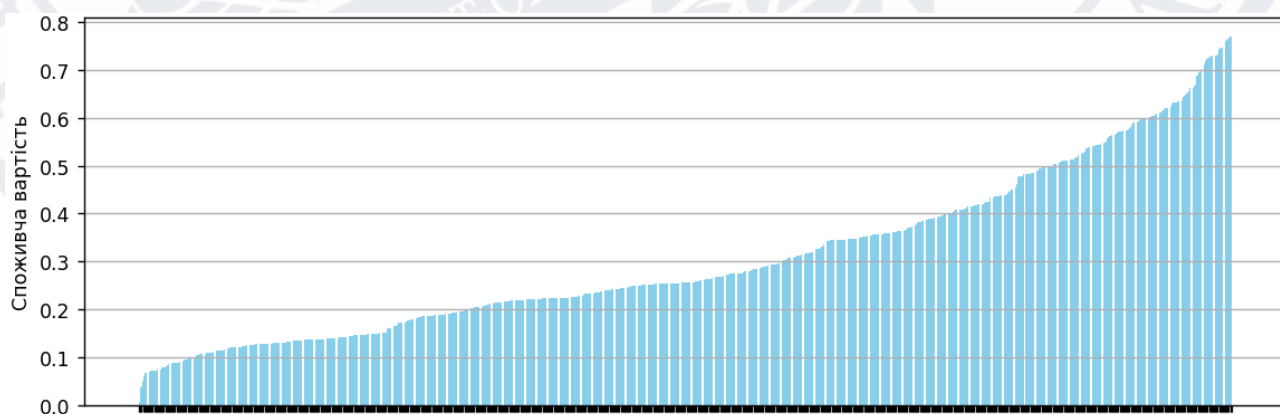


Рисунок 3.7 – Графік споживчої вартості для всіх ноутбуків

Також графік споживчої вартості можна співставити з графіком середніх цін по кожній з моделей ноутбуків, що зображено на рисунку 3.8.

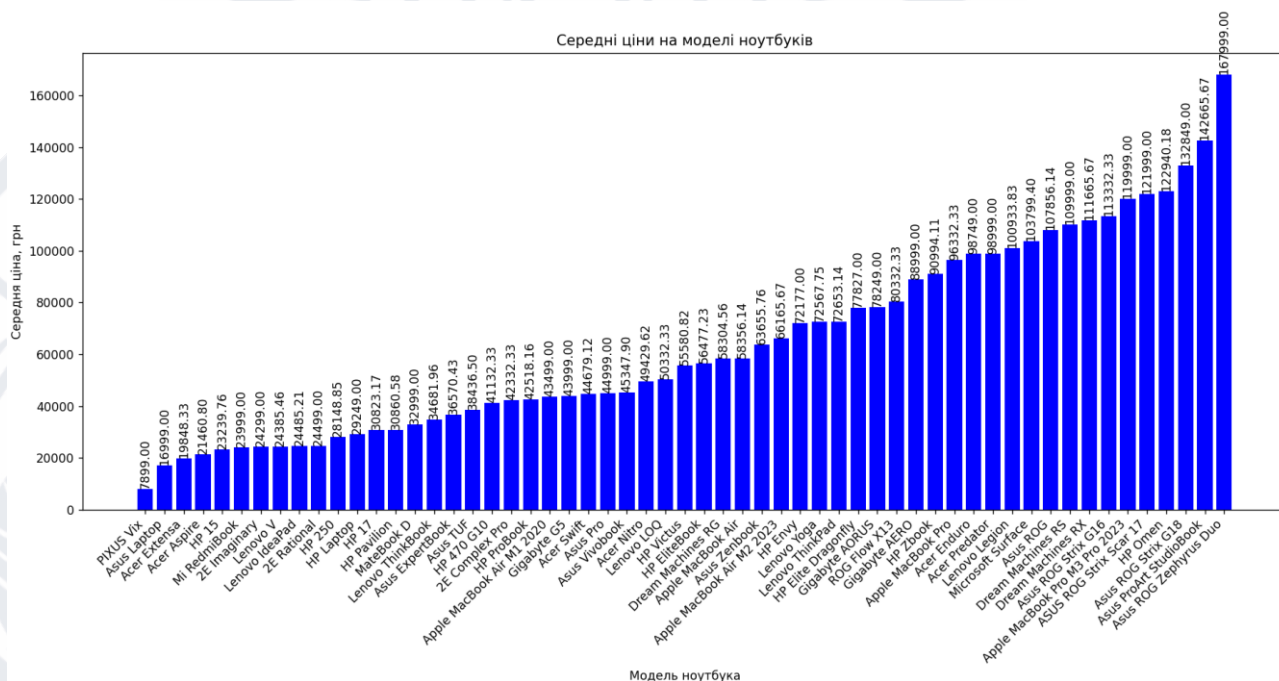


Рисунок 3.8 – Графік середньої ціни на моделі ноутбуків

На основі зібраних даних також можна виділити найпопулярніші значення в різних стовпцях таблиці:

1. Найпопулярніше значення у стовпці 'name': Lenovo ideapad Slim 5 16ABR8 Cloud Grey (82XG0058RA)
2. Найпопулярніше значення у стовпці 'price': 29999.0
3. Найпопулярніше значення у стовпці 'laptop_model': HP Zbook
4. Найпопулярніше значення у стовпці 'laptop_os': DOS
5. Найпопулярніше значення у стовпці 'display_diagonal': 15.6
6. Найпопулярніше значення у стовпці 'display_extension': 1920 x 1080
7. Найпопулярніше значення у стовпці 'display_matrix_type': IPS
8. Найпопулярніше значення у стовпці 'display_frequency': 60
9. Найпопулярніше значення у стовпці 'processor_series': Intel® Core™ i7 (13 покоління)
10. Найпопулярніше значення у стовпці 'processor_model': 1355U

11. Найпопулярніше значення у стовпці 'processor_frequency': 1.7
12. Найпопулярніше значення у стовпці 'processor_core': 10
13. Найпопулярніше значення у стовпці 'ram_size': 16
14. Найпопулярніше значення у стовпці 'ram_type': DDR4
15. Найпопулярніше значення у стовпці 'ram_frequency': 3200
16. Найпопулярніше значення у стовпці 'drive_type': SSD
17. Найпопулярніше значення у стовпці 'drive_size': 512
18. Найпопулярніше значення у стовпці 'gpu_type': інтегрована
19. Найпопулярніше значення у стовпці 'gpu_producer': Nvidia
20. Найпопулярніше значення у стовпці 'gpu_model': GeForce RTX4060
21. Найпопулярніше значення у стовпці 'gpu_size': 8

Ці найпопулярніші значення вказують на типові характеристики ноутбуків у даному наборі даних і можуть служити важливою інформацією для подальших аналітичних висновків та порівнянь.

3.3 Перспективи вдосконалення технології аналізу даних споживчої цінності товару

Розроблена інформаційна технологія аналізу даних для оцінки споживчої вартості ноутбуків має величезний потенціал для подальшого вдосконалення та розвитку. У межах даної роботи був реалізований продукт з базовим набором функцій, спрямованих на забезпечення стабільної роботи та демонстрації результатів магістерського дослідження.

Процес покращення та розширення функціональності може включати наступні етапи:

1. Впровадження механізмів обробки текстової інформації, зокрема відгуків користувачів. Це включатиме в себе використання технологій обробки природної мови (NLP), які дозволять системі розуміти та аналізувати текстові відгуки, враховуючи не лише їхню кількість, але й суттєві аспекти, висловлені споживачами. Цей підхід допомагатиме розширити рівень деталізації та об'єктивності при оцінці споживчої вартості товарів.

Для ефективного врахування впливу різних факторів на споживчу вартість ноутбуків, можна розробити систему вагових коефіцієнтів. Ці коефіцієнти визначатимуть важливість кожного параметра в оцінці загальної споживчої цінності. Наприклад, якщо для певного споживача важливіше є здатність пристрою до продуктивної роботи, ніж його зовнішній вигляд, то ваговий коефіцієнт для технічних характеристик може бути вищим. Такий підхід дозволяє адаптувати систему до індивідуальних уподобань різних користувачів та враховувати їхні особисті пріоритети при оцінці споживчої цінності ноутбуків.

2. Застосування передових методів машинного навчання з метою створення моделей, які здатні ефективно моделювати та прогнозувати динаміку споживчої вартості ноутбуків. Основною ідеєю є використання історичних даних щодо різних параметрів та показників, що впливають на споживчу цінність, для тренування алгоритмів машинного навчання. Алгоритми будуть орієнтовані на автоматичне розпізнавання основних змін та патернів у споживчому попиті, що дозволить системі реагувати на динамічні зміни у вимогах споживачів. Важливою частиною цього етапу буде використання розуміння історичних тенденцій для навчання моделей, які можуть ефективно адаптуватися до змін у споживчому поведінці та прогнозувати майбутні тенденції.

Застосування методів машинного навчання в контексті моделювання та прогнозування споживчої вартості ноутбуків сприятиме створенню точних та гнучких систем, які можуть враховувати складні та динамічні фактори на споживчому ринку.

3. Важливим інструментом для представлення результатів аналізу та оцінки споживчої вартості ноутбуків є інтерфейс. Основною метою цього етапу є створення інтуїтивного та зрозумілого інтерфейсу, який дозволить користувачам легко отримувати інсайти з отриманих даних.

У процесі розробки інтерфейсу приділяється увага визначенню оптимального способу візуалізації результатів аналізу. Це може включати в себе використання графіків, діаграм, теплових карт, та інших інтерактивних

елементів. Важливо, щоб інтерфейс забезпечував зручний доступ до ключових показників та дозволяв користувачам проводити власний аналіз даних.

4. Додатково, можна розробити засоби для зручного представлення даних, графіків та звітів. Це включатиме в себе можливість створення інтерактивних звітів, де користувачі зможуть детально вивчати важливі параметри та проводити порівняльний аналіз різних аспектів споживчої вартості. Засоби візуалізації стануть ключовим елементом для ефективного використання отриманих результатів та допоможуть користувачам приймати обґрунтовані рішення на основі аналізу зібраних даних.

5. Ще одним етапом у реалізації інформаційної технології аналізу даних щодо оцінки споживчої вартості ноутбуків є впровадження та підтримка системи в операційному середовищі. Цей етап передбачає введення розробленої технології в роботу, щоб вона стала необхідним інструментом для аналізу та оцінки споживчої вартості ноутбуків. Паралельно з впровадженням, надається необхідна технічна та консультаційна підтримка користувачам для ефективного використання системи.

6. Додатково, важливим аспектом є моніторинг та оновлення системи з урахуванням змін в ринкових умовах та технологічних новаціях. Система повинна бути адаптивною та готовою до змін, що відбуваються в галузі споживчої електроніки, щоб продовжувати надавати актуальні та точні результати. Постійний моніторинг ринку та впровадження оновлень допомагатиме системі залишатися відповідною до поточних тенденцій та потреб користувачів.

Завершенням цього етапу є встановлення процедур та механізмів для ефективного взаємодії з користувачами, вирішення їхніх питань та врахування їхніх фідбеків для подальшого вдосконалення системи. Впровадження та підтримка є важливою фазою, що створює основу для стійкої та ефективної експлуатації розробленої інформаційної технології.

ВИСНОВКИ ДО 3 РОЗДІЛУ

У даному розділі докладно розглянуто процес створення та вдосконалення парсера, який дозволив сформувати структуровані дані про різноманітні ноутбуки, доступні на інтернет-магазині. Описано взаємодію із веб-сайтом, використання бібліотек для парсингу та обробки отриманих даних, а також організацію цих даних у вигляді датасету.

Під час аналізу отриманого датасету були виявлені ключові характеристики ноутбуків, такі як модель, ціна, технічні характеристики компонентів, розмір оперативної та накопичувальної пам'яті, інформація про дисплей та відеокарту. Цей аналіз надав нам важливі висновки щодо різноманітності та розподілу цін на ноутбуки різних моделей і технічних характеристик.

Зокрема, було виявлено, що на ринку існує широкий спектр цін на ноутбуки, і розроблений парсер надає можливість зібрати інформацію для подальшого детального аналізу цінних тенденцій. Додатково, робочі зразки з інтегрованими та дискретними відеокартами показали різницю в цінах і дозволяють споживачам зробити більш обґрунтований вибір.

Завершуючи цей розділ, важливо вказати на перспективи подальшого розвитку цієї роботи. Можливості удосконалення включають розширення функціоналу парсера для збору додаткової інформації, вдосконалення методів аналізу даних та можливість автоматизованого оновлення датасету для відображення змін в цінах і характеристиках нових ноутбуків.

Отже, наша інформаційна технологія дозволяє не лише отримувати актуальну інформацію про ринок ноутбуків, але і має значний потенціал для подальшого розвитку та вдосконалення з урахуванням змін в сфері технологій та споживацьких уподобань.

ВИСНОВКИ

Проведене дослідження присвячено питанням розробки та практичної реалізації інформаційної технології аналізу даних оцінки споживчої вартості товару. За результатами дослідження можна зробити наступні висновки:

1. Обґрунтованість використання технологій аналізу даних споживчої цінності товару обумовлена системним підходом у формуванні політики проєктування інформаційних систем, збору даних та їх структуризації для аналізу конкурентного середовища.

2. Виявлено, що найбільш доцільним є використання подібних технологій для аналізу конкурентних середовищ розгалужених систем, якими є інтернет-магазини з розгалуженою структурою філіалів та доступом до широкого спектру різноманітних товарів для клієнтів.

3. Створення інформаційної технології, що охоплює процеси збору та аналізу даних по категоріях оцінки споживчої вартості товарів (на прикладі ноутбуків), включаючи технічні характеристики, відгуки користувачів, цінову інформацію та інші важливі параметри, будується на використанні інструментів та бібліотеки, таких як requests, BeautifulSoup, time, lxml та json.

4. При проведенні порівняльного аналізу інструментів, таких як веб-скрапінг, API, Data Extraction Tools, Natural Language Processing та Machine Learning Algorithms, визначено їх переваги та недоліки у контексті збору та аналізу даних. Обґрунтовано вибір мови програмування Python для написання парсера, а також представлено результати огляду популярних інтегрованих середовищ розробки (IDE) для Python, таких як PyCharm, Visual Studio Code, Jupyter Notebook та інші, з наголосом на їх можливості та переваги.

5. Реалізована інформаційна технологія включає в себе методи парсингу даних, обробки і аналізу, для отримання повного і об'єктивного огляду споживчої вартості ноутбуків. Результати перевірки розробленого додатку отримані на вибірці даних про ноутбуки з різними характеристиками ключових факторів, які впливають на споживчу вартість товару.

6. Проведений аналіз отриманих даних для визначення ключових факторів, які впливають на споживчу вартість товару та характеризують вплив технічних характеристик, бренду та ін. включає оцінку комплексного інтегрованого критерію з урахуванням цінності локальних характеристик в скупній оцінці споживчої цінності товару (ноутбуків).

7. Розроблена інформаційна технологія становить інноваційний інструмент для аналізу та оцінки споживчої вартості ноутбуків, що дозволяє забезпечити комплексний підхід до вивчення ринкових тенденцій і прийняття стратегічних рішень в області товарної стратегії та маркетингу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Porter M.E. The Competitive Advantage: Creating and Sustaining Superior Performance. 1985. 558 с.
2. Потапова Н.А. Логістика онлайн-торгівлі в контексті проявів глобалізації цифрової економіки. Економіка. Фінанси. Менеджмент: актуальні питання науки і практики. № 3. 2019. С. 62 – 77.
3. Потапова Н.А. Інформаційна логістика в сучасній ексабайтовій економіці. Маркетинг і логістика в системі менеджменту: тези доповідей XIV Міжнародної науково-практичної конференції. Львів: Видавництво Львівської політехніки. 2022. С. 183-184.
4. Потапова Н.А. Електронна логістика в системі інформаційної логістики підприємства. Маркетингове забезпечення продуктового ринку. Збірник тез XV Міжнародної науково-практичної конференції (м. Полтава, 16 травня 2023 року). Полтава: ПДАУ. 2023. С. 135-137.
5. Потапова Н.А., Денисюк В.О., Крохмалюк В.В. Парсинг як метод обробки даних при оцінці споживчої цінності товарів. Наука і техніка сьогодні. 2023. №13(27). С.819-827.
6. Зелінська О.В., Потапова Н.А., Волонтир Л.О. Інформаційні системи та технології в галузі. Вінниця, 2020. 253 с.
7. Обозна М. Д., Сопельник В. О. Споживча вартість: від міфів до реальності." проблеми розвитку економіки підприємства: погляд молоді. 2020. С. 38.
8. Sabda M.A., Suhardi Suhardi. Implementasi Data Mining Dalam Memprediksi Penjualan Parfum Terlaris Menggunakan Metode K-Nearest Neighbor. Jurnal Sistem Komputer dan Informatika. 2023. С. 415-422.
9. Ліщинський П. С. Створення моделі взаємодії для управління даними ІОТ. Організаційний комітет конференції. С. 34.
10. Сірик М. О. Програмний інтерфейс формування каталогу медичних препаратів. MS thesis. Сумський державний університет. 2023. С. 63

11. Jeff Friesen. Java XML and JSON: Document Processing for Java SE. 2019. 519 p.
12. Sai S Sriparasa, Bruno Joseph D'mello. JavaScript and JSON Essentials. 2018. 203 p.
13. Khetan Rajeev Kumar. An analysis of differences in clinical demographics, risk factors, proportion, in-hospital outcomes, and angiographic findings between elderly and young adults with myocardial infarction. Student's Journal of Health Research Africa 4.12. 2023.
14. Peres de Oliveira Luiz. Parsing gigabytes of JSON per second with parallel bit streams. 2023. 66 p.
15. Revathi K. Auto JSON: An Automatic Transformation Model for Converting Relational Database to Non-relational Documents." International Journal of Advanced Computer Science and Applications 14.3. 2023. pp. 661-668.
16. Nuijten Alex, Patrick Barel. Storing JSON in the Database. Modern Oracle Database Programming: Level Up Your Skill Set to Oracle's Latest and Most Powerful Features in SQL, PL/SQL, and JSON. Berkeley, CA: Apress. 2023. 283-309 p.
17. Береговой К. В. Аналітичний сервіс для вибору і купівлі товарів. 2023. 76 с.
18. Mitchell R. Web Scraping with Python: Collecting Data from the Modern Web. O'Reilly Media. 2015. 253 p.
19. Heydt M. Python Web Scraping Cookbook. Packt Publishing. 2018. 364 p.
20. Munzert S., Rubba C., Meißner P., Nyhuis D. Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. John Wiley & Sons, 2015. 310 p.
21. Jarmul K., Lawson R. Python Web Scraping: Hands-on data scraping and crawling using PyQT, Selenium, HTML and Python. Packt Publishing, 2017. 220 p.
22. Heydt M. Python Web Scraping Cookbook: Over 90 proven recipes to get you scraping with Python, micro services, Docker and AWS. Packt Publishing, 2018. 364 p.

23. Lawson R. Web Scraping with Python: Step by Step Guide to Easy Web Scraping, Data Extraction, and Web Crawling in Python. Packt Publishing, 2020. 179 p.
24. Ryan Mitchell. Web Scraping with Python: A Comprehensive Guide. 2018. 308 p.
25. Michael Heydt. Python Web Scraping Cookbook. 2018. 366 p.
26. Simon Munzert, Christian Rubba, Peter Meißner, Dominic Nyhuis. Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. 2015. 310 p.
27. Kim Okkonen. Python Web Scraping: Hands-on data scraping and crawling using PyQT, Selenium, HTML and Python. 2018. 174 p.
28. Richard Lawson. Web Scraping with Python: Step by Step Guide to Easy Web Scraping, Data Extraction, and Web Crawling in Python. 2020. 179 p.
29. Vasilis Vryniotis. Python Web Scraping: Hands-on Techniques and Tools to Automate Data Extraction. 2020. 366 p.
30. Laurence Moroney. Web Scraping with Python: Collecting More Data from the Modern Web. 2018. 210 p.
31. Andrew Bird. Mastering Python Web Scraping: Get Your Data Back. 2015. 306 p.
32. Richard Lawson. Python Web Scraping Cookbook: Over 90 proven recipes to get you scraping with Python, micro services, Docker and AWS. 2017. 249 p.
33. Martin Broadhurst. Python Web Scraping: Second Edition. 2018. 270 p.
34. Ryan Mitchell. Web Scraping with Python and BeautifulSoup. 2018. 382 p.
35. Paul Bradshaw. Scraping for Journalists: How to grab data from hundreds of websites. 2012. 257 p.
36. Alexandru Vladutu. Web Scraping for Beginners: How to Extract Data From Any Website With Python. 2020. 122 p.
37. John Watson. Web Scraping with Python: A Concise Guide to Data Collection and Web Scraping in Python. 2019. 110 p.

38. Samuel Robert. Python Web Scraping: The Bible of the Modern Web Scraping Masters. 2020. 154 p.

39. Солодченко Р. К. Веб-довідкова система аналізу продажу товарів. Сучасні комп'ютерні та інформаційні системи і технології: матеріали. 2021. С. 131.

40. Донцова О. Є. Інформаційна технологія рекомендації товарів для платформи онлайн продажів. MS thesis. Сумський державний університет, 2020. 73 с.

41. Митчелл Райан. Скрапинг веб-сайтов с помощью Python. Litres. 2022. 271 с.

42. Heydt Michael. Python Web Scraping Cookbook: Over 90 proven recipes to get you scraping with Python, microservices, Docker, and AWS. Packt Publishing Ltd. 2018. 335 p.

43. Lawson Richard. Web scraping with Python. Packt Publishing Ltd. 2015. 147 p.

44. Hajba, Gábor László. Website Scraping with Python. Berkeley: Apress. 2018. 223 p.

45. Chapagain Anish. Hands-On Web Scraping with Python: Perform advanced scraping operations using various Python libraries and tools such as Selenium, Regex, and others. Packt Publishing Ltd. 2019. 320 p.

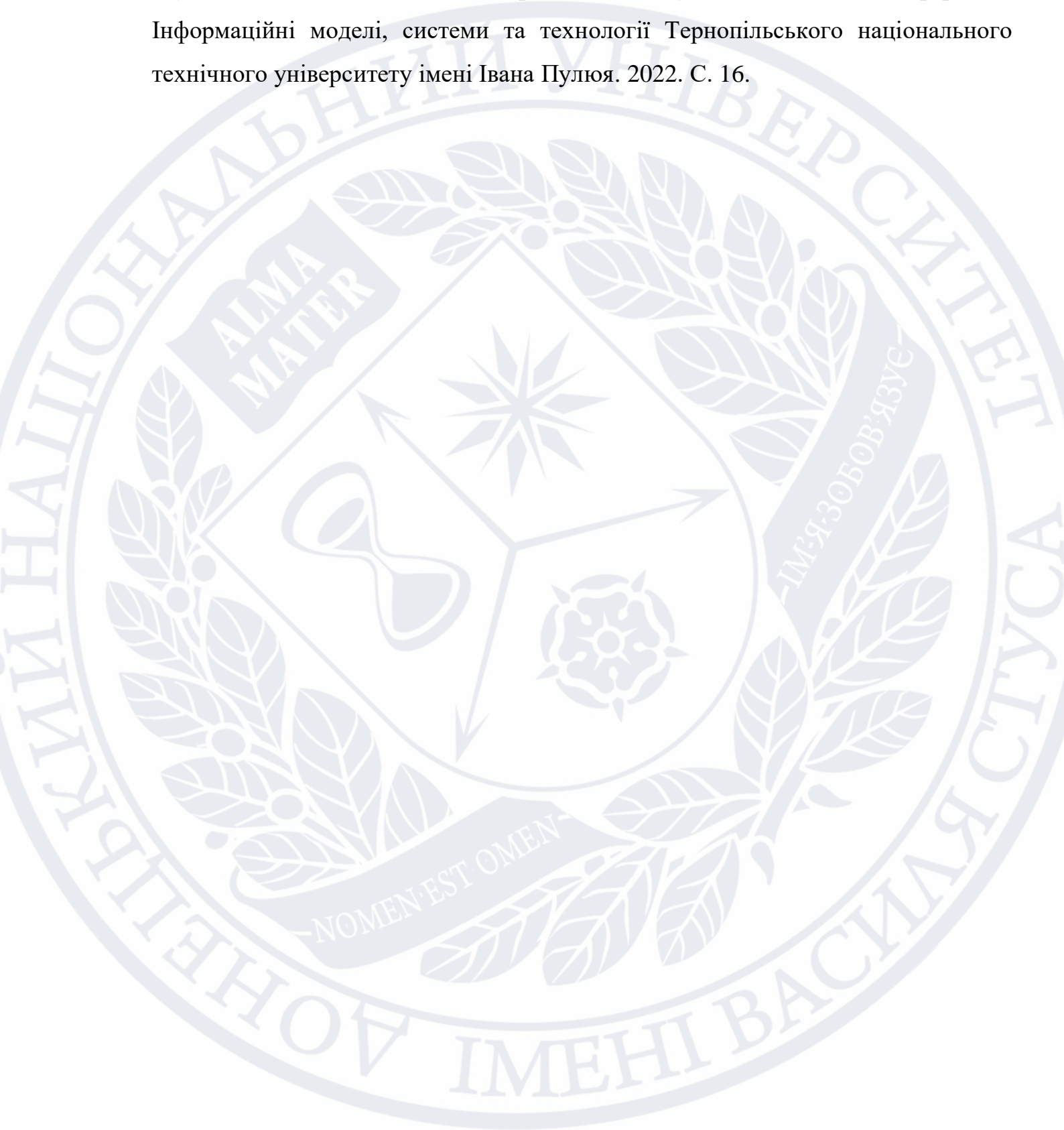
46. Mitchell Ryan. Web scraping with Python: Collecting more data from the modern web. O'Reilly Media, Inc. 2018.

47. Broucke Seppe Vanden, Bart Baesens. Practical Web Scraping for Data Science: best practices and examples with Python. Apress. 2018.

48. Ляпандра І. А., Івахів В. В., Білоус. В. С. Методи та засоби обробки великих даних. Збірник тез доповідей X Міжнародної науково-практичної конференції молодих учених та студентів. Актуальні задачі сучасних технологій. 2021. С. 105-106.

49. Прийма П. В. Дослідження засобів та методів машинного навчання для аналітичного опрацювання великих даних. MS thesis. 2023. 78 с.

50. Белоусов, К., Маєвський. Т. Великі дані та аналітичне опрацювання в наукових дослідженнях. Матеріали X науково-технічної конференції. Інформаційні моделі, системи та технології Тернопільського національного технічного університету імені Івана Пулюя. 2022. С. 16.



ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписалась/підписався, розуміючи та підтримуючи загально визнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»;
що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації;
що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

(дата)_____
(підпис)