КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
ЕКОНОМІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

Самостійна робота на тему:

«Застосування технології динамічного скрепінгу даних на сайтах електронної комерції»

Студента 1 курсу магістратури
напряму підготовки
«Економічна кібернетика»
Атояна Андрія

Викладач:
д.е.н., проф. Затонацька Т.Г.

Київ - 2020

ANNOTATION

Today, e-commerce has strong positions in the structure of world trade. Conducting trades and related financial transactions on the Internet can benefit all participants in a commercial relationship. Statistics published by a number of global financial institutions show a quantitative increase in digital buyers and sales worldwide. In the context of the current Covid-19 pandemic, the issue of the relevance of this type of trade is especially acute. The increase in the number of diseases, government measures related to restrictions on the operation of infrastructure facilities have led to a sharp increase in the fundamental indicators of e-commerce. The consequence of this surge is the emergence of the need to develop specific programs and algorithms for the convenience of shopping on the Internet.

Thus, **the purpose of this article** is to study the possibility of using dynamic data scraping technologies to form your own database of online stores.

This project is based on testing **the hypothesis** about the decency of online store owners. As a rule, during the holidays, prices for some goods are reduced in order to make the buyer want to buy them. But whether the indicated price is a discount, or a fiction, we can check by developing a bot to collect data.

Key words: e-commerce, web scraping, data analysis, bot, commercial websites.

ЗМІСТ

ВСТУП

Сьогодні, сфера електронної комерції займає міцні позиції в структурі світової торгівлі. Проведення торгів і відповідних фінансових операцій в інтернеті дозволяє отримати вигоду для всіх учасників комерційних відносин. Статистичні дані, опубліковані низкою глобальних фінансових інститутів, показують кількісне зростання цифрових покупців і продажів у всьому світі. В умовах поточної пандемії Covid-19 питання актуальності даного виду торгівлі є особливо гострою. Збільшення кількості захворюваності, заходи урядів країн, пов'язані з обмеженнями роботи звичайної інфраструктури призвели до різкого збільшення базових показників електронної комерції. Наслідком даного сплеску є виникнення потреби розробки окремих програм і алгоритмів для зручності здійснення покупок в інтернеті.

Метою написання цієї роботи є вивчення можливості застосування технологій динамічного скрепінгу даних для формування власної бази даних онлайн-магазинів.

Відповідно до мети, **завдання роботи** полягає в:

- Дослідженні розвитку сфери електронної комерції, її ролі у структурі світової торгівлі.
- Огляді існуючих програмних забезпечень та інструментів скрепінгу даних.
- Дослідженні методів та підходів для написання автоматизованого алгоритму збору даних комерційних веб-сайтів.
- формуванні власної бази даних та аналізі результатів.

В основі даного проекту лежить перевірка **гіпотези** про порядність власників інтернет-магазинів: «як правило, в періоди свят ціни на деякі товари зменшують, з метою викликати у покупця бажання придбати їх. Проте, чи є зазначена ціна знижкою, або фікцією ми зможемо перевірити, розробивши бот для збору даних».

Об’єктом дослідження є дані товарів компаній на електронних ресурсах.

Предметом дослідження є автоматизація процесу збору відповідної інформації для її використання у подальшому аналізі.

Практичне значення полягає в можливості використання даного автоматизованого алгоритму для збору інформації щодо товарів компанії та ведення персональної бази даних. **Теоретична цінність** обумовлена аналізом і порівнянням методів та підходів скрепінгу, виявлення їх недоліків та переваг.

ДОСЛІДЖЕННЯ РОЗВИТКУ СФЕРИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ В СВІТІ

Як ми вказали раніше, кожен з учасників торгівлі отримує вигоду, здійснюючи покупки онлайн. Наприклад, покупці можуть значно заощадити свій час, розширюється географія їх покупок, безготівкова оплата забезпечує прозорість. Також покупці можуть самі стати продавцями, розмістивши відповідні оголошення на електронній платформі (C2C). У свою чергу, компанії отримують плату за розміщення, надаючи продавцеві таку можливість (B2C). В добавок, платформа може заробляти шляхом розміщення реклами. Таким чином, всі операції є прозорими і з кожної сплачено податок. В даному випадку, виграє держава.

Такий сприятливий клімат, призводить до зростання впливу електронної комерції в міжнародних торгах. Згідно зі статистикою, зібраною компанією «Statista»[1], ми можемо розглянути динаміку частки електронної комерції в структурі роздрібних продаж глобальних роздрібних продажів через інтернет в 2014-2023 рр. Дані представлені в мільярдах доларів США.

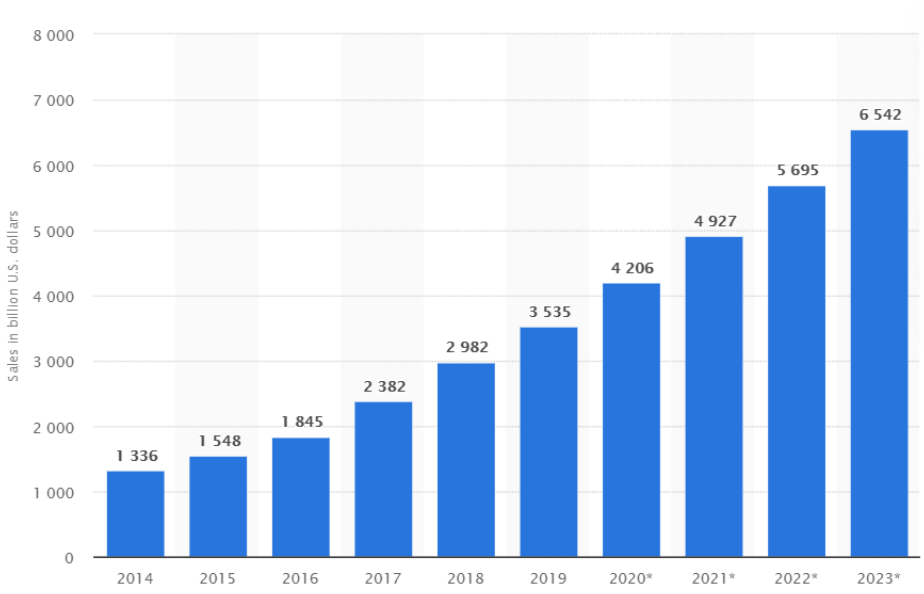

Рис. 1.1. Частка електронної комерції в структурі роздрібних продаж з 2014 по 2023 рік (у мільярдах доларів США)

Джерело: Розроблено автором на основі [1]

Як ми можемо бачити, в період з 2014 по 2019 року спостерігається поступове збільшення фінансового обсягу продажів. Однак починаючи з 2020 року темпи

зростання значно змінилися. У першою чергу це обумовлено поточною пандемією, яка внесла значні корективи в усі сфери економічної діяльності.

У той час як одні компанії зазнають збитків, які можуть привести до банкрутства, інші – стають сильнішими ніж будь-коли. Яскравими прикладами є Amazon, Alibaba, Prosus, ринкові капіталізації яких, на вересень 2020 становлять 1597,2; 777,5 і 146,1 мільярдів доларів відповідно [2].

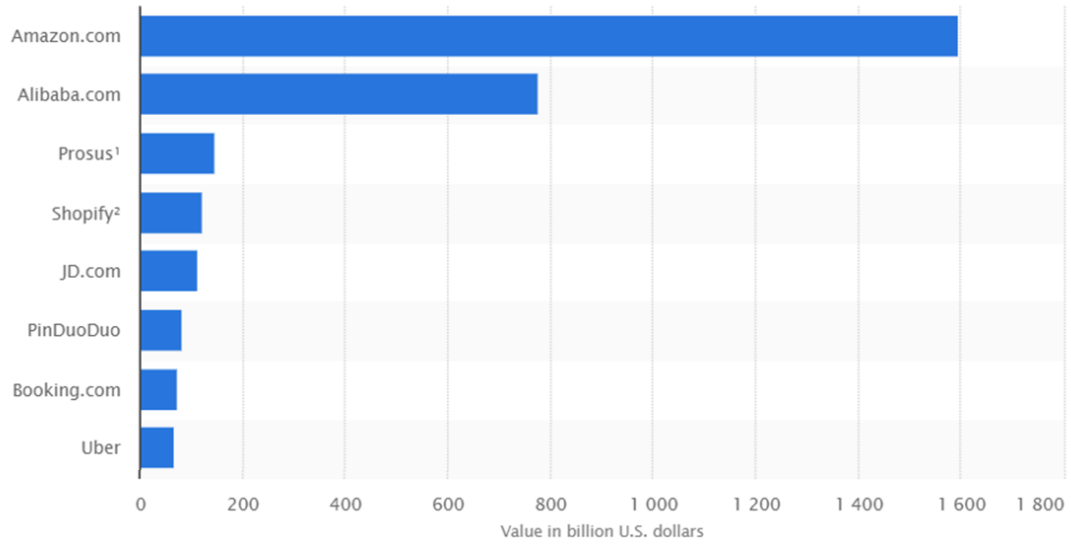

Рис. 1.2. Ринкова капіталізація провідних світових компаній в області електронної комерції станом на вересень 2020 р.

Джерело: Розроблено автором на основі [2]

Особливо вдалим рік видався для компанії Amazon, яка внаслідок пандемії і пзбільшення частки роздрібних продажів в інтернеті, змогла заробити 609 мільярдів з початку року (64,5% від початкової суми) [3]. Такі показники свідчать про потенціал збільшення частки електронної комерції в загальносвітовому обсязі роздрібних продаж. За оцінками експертів, на період 2021 вона становитиме 18,1% [4].

Таким чином, значне розширення сфери електронної комерції в поточному і майбутньому призводить до виникнення попиту на створення різнопланових програм. Головним завданням розробників є підвищення якості послуг, що надаються і зручності здійснення покупок в інтернеті.

Серед наукових статей присвячених цій темі, можна виділити ряд робіт де основний акцент робиться на моніторингу руху цін товарів на електронних ресурсах.

Серед таких – дослідження З. Уллаха та Х. Уллаха [7], щодо створення власного програмного продукту для збору інформації. В статті розглядається підходи скрепінгу даних, особлива увага приділяється типам пошукових механізмів, а саме: методам відстані Хеммінга і Левенштейна, їх порівнянню.

Для проведення тесту використовувався фреймворк Scrapy для парсинга в writecrawler. Сканер був налаштований і протестований для вилучення даних з різних веб-сайтів. Результатом роботи є вилучення 15000 записів продуктів, а приблизний час скрепінгу склав 11 хвилин. На основі проведеного дослідження перевага була надана методу відстані Левенштейна. Критеріями оцінки були: швидкість, складність та пропускна здатність.

Також акцент робився на оптимальності використання ресурсів комп'ютера. На основі проведених досліджень, був обраний підхід для розробки власного програмне забезпечення. Його особливістю є надання можливість пошуку цільового продукту в одному консолідованому місці замість пошуку по різним веб-ресурсам. Дана стаття має сильну математичне обґрунтування та розкриває безліч технічних аспектів.

Як ми підкреслили з минулої статті, процес вибору інструментів для збору даних передбачає ряд досліджень та застосування різних підходів. У своїй статті Сауркар А.В. та інші [8], детально розглянули основні принципи збору інформації в інтернеті. Особлива увага, була зосереджена на вже існуючих програмах та компонентах, зіставлення та порівняння яких лягло в основу написання практичної частини. Слід зазначити, що веб-скрепінг також досить поширений в академічних базах даних, таких як Scopus, Web of Science і Inspec [7].

Отже, ми можемо бачити широку популярність використання даного підходу у розробці окремих програм, їх використання як у повсякденній роботі, так і в наукових дослідженнях.

ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ ЗАБЕЗПЕЧЕНЬ ТА ІНСТРУМЕНТІВ СКРЕПІНГУ ДАНИХ

На сьогоднішній день існує величезна кількість як окремих інструментів для скрепінгу даних, так і повноцінних програмних забезпечень. Як правильно, використання даних послуг - не дешеве задоволення, що особливо відчутно для середнього та малого бізнесу. Оскільки, акцент у даній роботі робиться на моніторингу цін товарів певних електронних ресурсів, то у першу чергу розглянемо програми які широко використовуються для задоволення даної мети. Серед таких можна виділити «Mozenda», «Octoparse» та «Data Crops» [9].

Програмне забезпечення «Mozenda» представляє собою зручний інструмент, який дозволяє користувачам без відповідних технічних знань легко отримувати дані з сайтів та аналізувати їх. Структурно програма складається з «конструктора агентів» і «веб-консолі».

Перший елемент представляє собою зручний інтерфейс за допомогою якого користувач, ввівши посилання та використовуючи інтерактивну складову, створює свого роду набір інструкцій, якими пізніше керується програма. Також, конструктор агентів надає можливість побачити одержані в результаті роботи дані - preview (Рис 2.1).

Фактично, процес встановлення відповідності між діями користувача та автоматизацією процесу збору даних, можна описати фразою «наведи і натисни».

Фінальний результат - сформований файл інструкцій, називається «агентом». Користувач має можливість запустити його роботу, використовуючи другий елемент програми - «веб-консоль». Після виконання, дані можуть бути експортовані, або використані для формування звітності.

Слід зазначити, що «Mozenda» це хмарний сервіс, отож всі агенти працюють на високо оптимізованих серверах збору та оброки інформації. Перевагами даного продукту є наявність простого інтерфейсу, зручний процес автоматизації та багатозадачність: можливість не тільки збирати дані, але й розміщувати їх на електронних ресурсах, обробляти їх всередині середовища та формувати звітність. Підтвердження цього є його популярність та широке застосування.

Ще одним відомим інструментом для скрепінгу є «Octoparse». Як і в попередньому прикладі використання даного продукту виключає необхідність володіння певними технічними знаннями щодо процесу збору та мов програмування.

Дана можливість обумовлена головною розробкою компанії, а саме шаблонів скрепінгу. Вони представляють собою рішення, з допомогою якого, користувач може отримати контент практично з будь-якого веб-сайту і зберегти чисті структуровані дані у вказаному форматі.

Octoparse імітує роботу людини через вбудований браузер. Боти виконують роботу по перегляду, пошуку та вилучення даних. Розширені налаштування, включаючи прокрутку веб-сторінок, очікування перед виконанням і т. д., роблять весь процес вилучення даних більш ефективним (рис. 2.2).

Щоб запобігти захисту веб-сайтів, де вбудовані відповідні методи захисту від скрепінгу, Octoparse надає проксі-сервер, ротацію IP-адрес, призначених для користувача агентів, обхід Captcha, очищення файлів cookie, щоб запобігти перериванню збору інформації з веб-сторінок.

Таким чином, як досвідчені, так і звичайні користувачі можуть легко використовувати Octoparse для масового вилучення інформації з веб-сайтів, де для більшості завдань персональне кодування не потрібне.

Останньою технологією в розрізі даного розділу є DataCrops. Дане програмне забезпечення, витягує інформацію за допомогою вдосконаленої технології, яка може автоматично сканувати веб-сайти, збирати, трансформувати і завантажувати дані із вказаною періодичністю і часом.

Багато в чому підхід даної платформи, щодо накопичення та обробки інформації схожий з попередніми. Таким чином, аби уникнути повторень ми перейдемо до фінальних висновків цього розділу.

Як ми можемо бачити, дані платформи володіють потужним та ефективним арсеналом. Вони дозволяють швидко та якісно проводити вилучення на накопичення даних для їх аналізу. Також основною перевагою є зменшення порогу входу для користувачів, а саме відсутність необхідності у технічних навичках та знань певних процесів.

Причиною, по якій компанії шукають альтернативу їх застосування – є ціна. Оскільки використання даного продукту є дорогим задоволенням. Для прикладу, використання «Mozenda» передбачає щомісячну оплату у 250 доларів. Дані витрати є доволі чуттєвими для малого і середнього бізнесу.

Саме тому виникає необхідність у розробці власного підходу для задоволення даної потреби. Одним з таких – є використання мов програмування із застосування певних програм. Даний підхід спроможний не тільки зменшити витрати компанії, а й виконувати вузько спеціалізовані завдання.

МЕТОДОЛОГІЯ

Web scraping є технологією сканування даних з веб-сторінок. Шляхом визначення ряду ключових тегів та їх атрибутів, ми маємо можливість вилучити наявну в них інформацію. Залежно від виду інформації: статичної або тієї, що змінюється під впливом JavaScript, web scraping поділяється на статичний і динамічний.

В ході даної роботи, для отримання даних з веб-ресурсів буде використовуватися програмна бібліотека «Selenium WebDriver». Часом, її помилково називають «драйвером браузера», але насправді це ціле сімейство драйверів для різних браузерів. Також туди входить набір клієнтських бібліотек на різних мовах програмування.

Доволі часто для цих потреб використовується python, оскільки він більш популярний і має широку сферу застосування. Однак мова R володіє явною перевагою в кількості бібліотек для data science, що корисно для подальшої обробки інформації. Тому, написання автоматизованого алгоритму буде проводитися саме на ньому.

Зв'язка R з «Selenium WebDriver» здійснюється за допомогою використання бібліотеки «RSelenium». Важливим компонентом для здійснення роботи є використання Docker.

Дана програма являє собою засіб для упаковки, доставки і запуску додатків. Його використання дозволяє уникнути ряду помилок, які можуть виникнути під час запуску програми або коду на різних комп'ютерах. Така властивість зумовлено насамперед структурою самого Docker і його складових: «image» і «container». Для того що б наочно бачити те, що відбувається в контейнері Докера використовується програма TightVNC.

Слід зазначити, що виконання даної роботи передбачає володіння рядом мов, власне: R, JavaScript, HTML, CSS та xPath. Останній елемент є представляє собою мову запитів, за допомогою яких, процес знаходження та вилучення бажаних об’єктів стає простішим та більш ефективним.

Табл. 1.1 Технічний перелік необхідних програм і компонентів

R language	https://cran.r-project.org/bin/windows/base/
R Studio	https://rstudio.com/products/rstudio/download/
Docker Desktop	https://docs.docker.com/docker-for-windows/install/
ChromeDriver	https://chromedriver.chromium.org/downloads
TightVNC	https://www.tightvnc.com/download.php

Джерело: Розроблено автором

Список необхідних програм та компонентів знаходиться у Табл. 1.1. Отож перед тим, я к приступити до практичного виконання завдання, налаштуємо базові елементи програми Docker.

Для роботи з Docker необхідно налаштувати його базові елементи. Оскільки в даній роботі скрепінг даних відбувається у браузері Google Chrome, нам необхідно встановити відповідний шаблон інструкцій – «image» (рис. 3.1).

```
PS C:\Users\Work> docker pull selenium/standalone-chrome-debug
```

Рис. 3.1 Завантаження «image» для контейнеру.

Джерело: розроблено автором.

На його основі ми маємо створити «container» - середовище, де відбуватиметься робота нашого боту (рис. 3.2). Після створення контейнеру, він автоматично буде запущений. Для того щоб зупинити/почати його у майбутньому роботу необхідно ввести наступні команду: docker stop/start (container id).

```
PS C:\Users\Work> docker run --name malishka -d -p 4445:4444 -p 5901:5900 selenium/standalone-chrome-debug
```

Рис. 3.2 Створення Docker container

Джерело: розроблено автором.

У разі необхідності ми можемо спостерігати за середовищем контейнеру, використовуючи програму TightVNC. Для цього достатньо ввести наступну команду в поле Remote Host: localhost:5901.

Таким чином, налаштувавши всі необхідні компоненти ми можемо приступити до виконання практичної частини.

ПРАКТИЧНА ЧАСТИНА

Для збору інформації ми скористаємося електронним ресурсом «Wine & Food», де акцент робимо на винному асортименті. Такий вибір обмовлений періодом «Чорної п'ятниці» – всесвітній днем розпродажів. Датасет формувався з 26.11.20 по 30.11.20.

Першим етапом автоматизації є знайомство зі структурою сайту. Це можна зробити натиснувши клавіші «Ctrl+Shift+I». Після цього відкриється вкладка «інструменти розробника», або простіше кажучи Chrome DevTools (рис. 4.1). З його допомогою, ми можемо переглянути код та будову сайту, прослідкувати взаємозв’язки окремих елементів, дослідити зміни які відбуваються під дією JavaScript, визначитися з тегами для їх подальшого використання.

Загалом, структура сайту є доволі зручною та інтуїтивно зрозумілою. Даними, які привернули нашу увагу стали назви брендів та інформація винного асортименту. Для збору перших, ми вирисовували комбінацію пакетів статичного та динамічного скрепінгу - «rvset» та «RSelenium» відповідно. Слід зазначити, що основним інструментом звернення до елементів сторінки була мова запитів xpath. Даний підхід дозволяє перебачити низку проблем, вказавши прямий шлях до групи об'єктів.

Отже, запустивши docker container, та перейшовши на необхідну сторінку, нам необхідно було вилучити інформацію про бренди, яка міститься у випадаючому списку. Трудність полягала в тому, що даний список входить до блоку «div id="mCSB_1_container"», який в свою чергу є дочірнім елементом «div class = "bx-filter-parameters-box"». Останній змінюється під дією скрипту, що робить доступ до інформації можливим лише у випадку, коли даний блок є активним. Цей стан супроводжується відповідним записом в кінці класу "*bx-active".

Використовуючи інструменти «Selenium», нам частково вдалося вирішити цю проблему. Проте, вилучення інформації стало можливим лише для об’єктів, які потрапляють у поле нашого зору. Інші – незважаючи на наявність інформації у коді сторінки, залишаються недоступними. Даний ефект обумовлений дією скрипту батьківського вузла.

Отже, автоматизація підхожу поступового прокручування списку є вкрай не ефективним та дуже вразливим до помилок підходом. Таким чином, постала потреба у знаходженні альтернативного рішення даної проблеми.

Обійти даний аспект вдалося за допомогою комбінації двох вище зазначених пакетів. Перейшовши на сайт, ми імпортували код сторінки в R, за допомогою функції getPageSource(). Дану інформацію ми зберегли у змінній списку, який пізніше перетворили на веб-елемент, для подальшої роботи з інструментами пакету «rvset». Записавши шлях, з використання xpath, в структурі функції, нам вдалося вилучити назви усі брендів.

Наступним етапом є вилучення інформації винного асортименту. Відповідні записи знаходяться у блоках «div class = "product-container"», які місять в собі опис кожного з товарів. Це легко перевірити, скориставшись інструментом пошуку в Chrome DevTools. Як ми можемо бачити кількість блоків відповідає кількості об’єктів на сторінці.

Питання постає лишу у способі збору цих даних. Найпростіший з них – це пряме звернення до кожного блоку та вилучення наявного там тексту. Результатом цього э сформований список з текстових векторів (рис. 4.2). Проте, окрім назви бренду, товару та цін (звичайних та знишкових), вектор міститиме зайву інформацію. Звичайно, визначивши структуру, даний патерн запису можна розбити та вилучити необхідну інформацію. Проте використання даного підходу є доволі сумнівним.

Саме тому, для вирішення даного питання ми знову скористалися мовою xpath, застосувавши осі – важливу складову мови, яка дозволяє значно скоротити шлях до об’єкту, дозволяє досягти тієї ж ефективності при менших витратах ресурсів (рис. 4.3).

Таким чином, сформувавши універсальних шлях до об’єктів, нам вдалось автоматизувати даний процес. Також, даний блок коду був написаний у тілі функції TryCatch(), з метою запобігання можливих помилок в процесі скрепінгу.

Завершальним етапом автоматизації є налаштування процесу переходу на інші сторінки. Фактично, для виконання цього завдання використовувався схожий підхід.

Середній час роботи боту складає 4 хвилини 50 секунд. Результатами є вилучення 481 запису брендів та 1360 винного асортименту. Слід зазначити, що в тілі коду передбачений рядок, що дозволяє зупинити виконання алгоритму з переходом на іншу сторінку. Дана властивість можлива за рахунок використання функції Sys.sleep(), значення якої становить 5 секунд. Такий підхід дозволяє передбачити ряд помилок які можуть виникати при неповному завантаженні сторінки.

Отже, описавши основні аспекти роботу алгоритму, ми можемо перейти до діагности отриманих результатів. Дані ресурсу збиралися з 26.11.20 по 30.11.20 в період «Чорної п'ятниці». Нище представлена таблиця розбіжностей цін на певні товари.

	wine	price	price_2
4	Вино Parras Alfacinha Tinto 0,75 L	Цена: 139.99 грн. 177 грн.	Цена: 189 грн.
5	Вино D'Adimant Rose 0,75 L	Цена: 299 грн. 399 грн.	Цена: 444 грн.
76	Вино Chateau Pigoudet Classic Rose 0,75 L	Цена: 555 грн.	Цена: 499 грн. 599 грн.
90	Вино Paco Lola №12 Albarino Rias Baixas DO 0,75 L	Цена: 399.99 грн. 460 грн.	Цена: 440 грн.
92	Вино Chateau Pigoudet Premiere Rose 0,75 L	Цена: 498 грн.	Цена: 399 грн. 479 грн.
110	Вино Pata Negra Viura Rioja DO 0.75 L	Цена: 239 грн.	Цена: 258 грн.
129	Вино Lynx Chardonnay 0,75 L	Цена: 498 грн. 555 грн.	Цена: 399.99 грн. 555 грн.
253	Вино 242 Pinot Noir Cabernet Sauvignon 0,75 L	Цена: 199 грн. 244 грн.	Цена: 228 грн.
274	Вино Giacondi Chianti DOCG 0,75 L	Цена: 199 грн. 239 грн.	Цена: 199 грн.
297	Вино Carlos Serres Viura - Tempranillo blanco 0.75 L	Цена: 129 грн. 177 грн.	Цена: 139 грн.
298	Вино Carlos Serres Tempranillo Rioja 0.75 L	Цена: 129 грн. 177 грн.	Цена: 139 грн.
304	Вино Torrevento Bolonero DOC Castel Del Monte 0,75 L	Цена: 333 грн. 399 грн.	Цена: 299 грн. 399 грн.
368	Вино L'interrogant 0,75 L	Цена: 699 грн. 798 грн.	Цена: 760 грн.
429	Вино Vinas Del Vero Gewurztraminer 0,75 L	Цена: 299 грн. 399 грн.	Цена: 399 грн.
519	Вино Domaine du Cleray Chardonnay 0,75 L	Цена: 299 грн. 399 грн.	Цена: 399 грн.
699	Вино Noval Porto Tawny 0,75 L	Цена: 498 грн. 599 грн.	Цена: 555 грн.
762	Вино Domaine de Pellehaut Harmonie Blanc 0,75 L	Цена: 219 грн. 269 грн.	Цена: 239 грн. 269 грн.

Рис. 4.4 Таблиця розбіжностей цін на певні товари за період 26.11.20 - 30.11.20. Джерело: Розроблено автором.

Як ми можемо бачити, існує певна розбіжність. Ціноутворення деяких з них дійсно має справедливий характер, проте більшість спостережень свідчать про

наміри власників заробити з повітря. Отож, гіпотеза, яку ми сформулювали на початку цієї статті знайшла своє підтвердження на практиці.

Також, ми перевірили наявність нових брендів. Їх фактична кількість у 30.11.20 складає 481 шт., проти 480 на початок дослідження. Брендом, який поповнив склад вже існуючих, став: "Leleka Wines".

Певних змін зазнав і асортимент вин, кількість яких, зросла з 1351 до 1360. Перелік відповідної продукції представлений нижче.

```
[1] "Вино Venica e Venica Ronco Bernizza Chardonnay 0.75 L"
[2] "Вино Venica e Venica Friuliano Collio DOC 0.75 L"
[3] "Вино Cantina Menhir Primitivo Di Manduria 0.75 L"
[4] "Вино Leleka Wines Chardonnay 0,75 L"
[5] "Вино Leleka Wines red semi-sweet 0,75 L"
[6] "Вино Leleka Wines Pinot Gris 0.75 L"
[7] "Вино Leleka Wines Merlot 0,75 L"
[8] "Вино Leleka Wines Cabernet Sauvignon 0,75 L"
[9] "Вино Leleka Wines white semi-sweet 0,75 L"
```

На рис. 4.5 представлений відповідний список нової продукції.

Джерело: розроблено автором.

ВИСНОВКИ

Як ми можемо бачити, сфера електронної комерції являє собою одну з найбільш популярних форм торгівлі на сьогоднішній день. Показники які ми маємо потягом останніх років є прямим підтвердженням цього факту. Свої корективи також внесла пандемія Covid-19, яка збільшила частку ecommerce в структурі світової торгівлі. Найбільші компанії цього сектору, як Amazon, Alibaba та Prosus істотно наростили свої фінансові ресурси, що дозволило їм не тільки уникнути труднощів, а й відкрити нові горизонти.

Такий сприятливий клімат, призводить до виникнення попиту на створення різнопланових програм та інструментів, головною метою яких є покращення умов здійснення покупок в інтернеті. Одним з перспективних напрямків є розробка програм для збору даних з веб-ресурсів з метою моніторингу ціни.

Однак, незважаючи на багату функціональність й їх значну популярність, дані розробки залишаються недоступними для цілого ряду компаній. Ціна використання таких програм є вкрай чутливою для малого та середнього бізнесу. Отож, виникає потреба у знаходженні іншого підходу для задоволення власних проблем. Одним з таких – є написання власного автоматизованого алгоритму.

Відповідно до мети, була досліджена можливість застосування технологій динамічного скрепінгу даних, а також, представлена її практична реалізація. Для автоматизації процесу збору даних був використаний ресурс «Wine & Food», дані якого збиралися в період «Чорної п'ятниці» з 26.11.20 по 30.11.20.

Згідно з результатів аналізу, був виявлений ряд розбіжностей ціни товарів, розміщених на сайті компанії. Ціноутворення деяких з них дійсно має справедливий характер, проте більшість спостережень свідчать про наміри власників заробити, створюючи фіктивні знижки. Даний факт лише підтверджує нашу гіпотезу.

Також, були виявлені кількісні зміни, щодо брендів та винного асортименту. Відтепер, перелік брендів складає 481щт., проти 480 на початок дослідження, тоді як кількість вин зросла на 9 одиниць.

Отож, застосування динамічного скрепінгу даних для написання автоматизованого алгоритму є доволі ефективним. Звичайно, даний підхід потребує

технічних навичок: розуміння певних процесів та знань мов програмування, що відчутно підвищує поріг використання даної технології на практиці. Проте, оволодіння цим інструментом дозволяє вирішувати вузько спеціалізовані задання та відчутно зменшити витрати компаній. Адже використання існуючих аналогів є доволі не дешевим задоволенням.

ДЖЕРЕЛА

1. Retail e-commerce sales worldwide from 2014 to 2023 (in billion U.S. dollars). Global Business Data Platform «Statista» [An electronic resource]. – URL: https://bit.ly/3m5SMt2
2. Market cap of leading consumer internet and online service companies worldwide as of September 2020 (in billion U.S. dollars). Global Business Data Platform «Statista» [An electronic resource]. – URL: https://bit.ly/3pX1PyO
3. Amazon.com Inc (AMZN) Market Cap. Research platform «YCharts» [An electronic resource]. – URL: https://ycharts.com/companies/AMZN/market_cap
4. E-commerce share of total global retail sales from 2015 to 2023. Global Business Data Platform «Statista» [An electronic resource]. – URL: https://bit.ly/2KDFJ4o
5. Number of digital buyers worldwide from 2014 to 2021(in billions). Global Business Data Platform «Statista» [An electronic resource]. – URL: https://bit.ly/3pZu6oE
6. Top 10 Price Monitoring tool in 2020. Octoparse web scraping blog [An electronic resource]. – URL: https://www.octoparse.com/blog/top-10-price-monitoring-tool
7. Ullah Z., Ullah H. Web Scraper Revealing Trends of Target Products and New Insights in Online Shopping Websites. [An electronic resource]. – URL: https://bit.ly/3qAUrty
8. Saurkar A.V., Pathare K.G., Gode S.A. An Overview on Web Scraping Techniques and Tools. [An electronic resource]. – https://bit.ly/3owXKzS

ДОДАТКИ

Додаток А

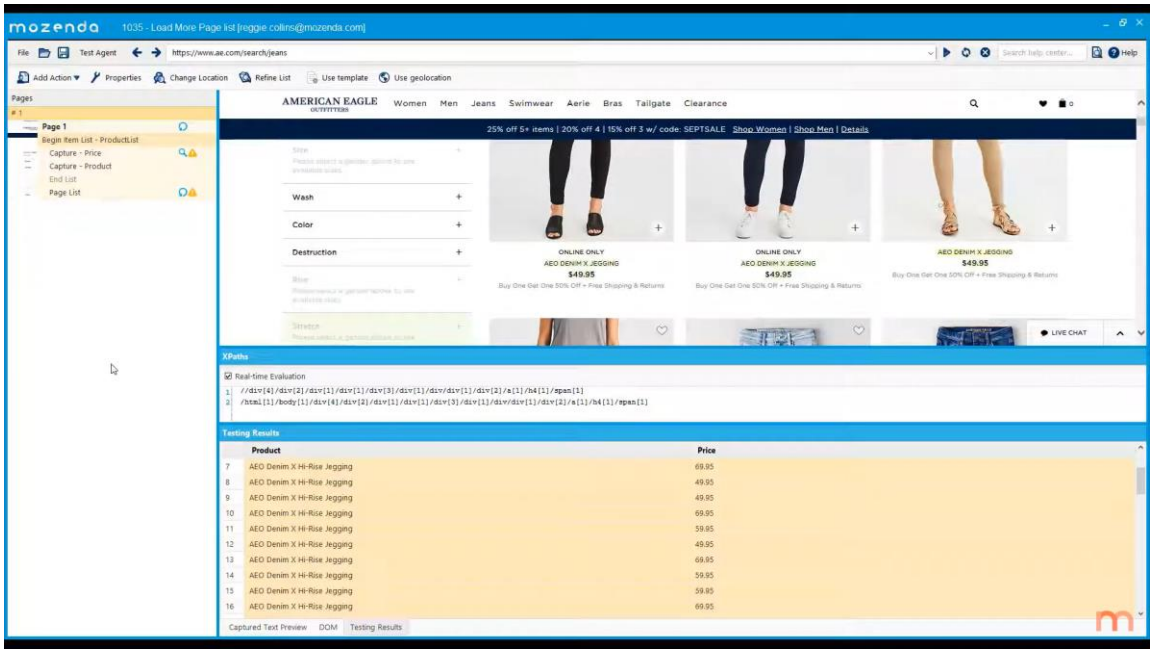

Рис. 2.1 Середовище «конструктора агентів» програми «Mozenda».

Джерело: Mozenda, official website. URL: https://www.mozenda.com/

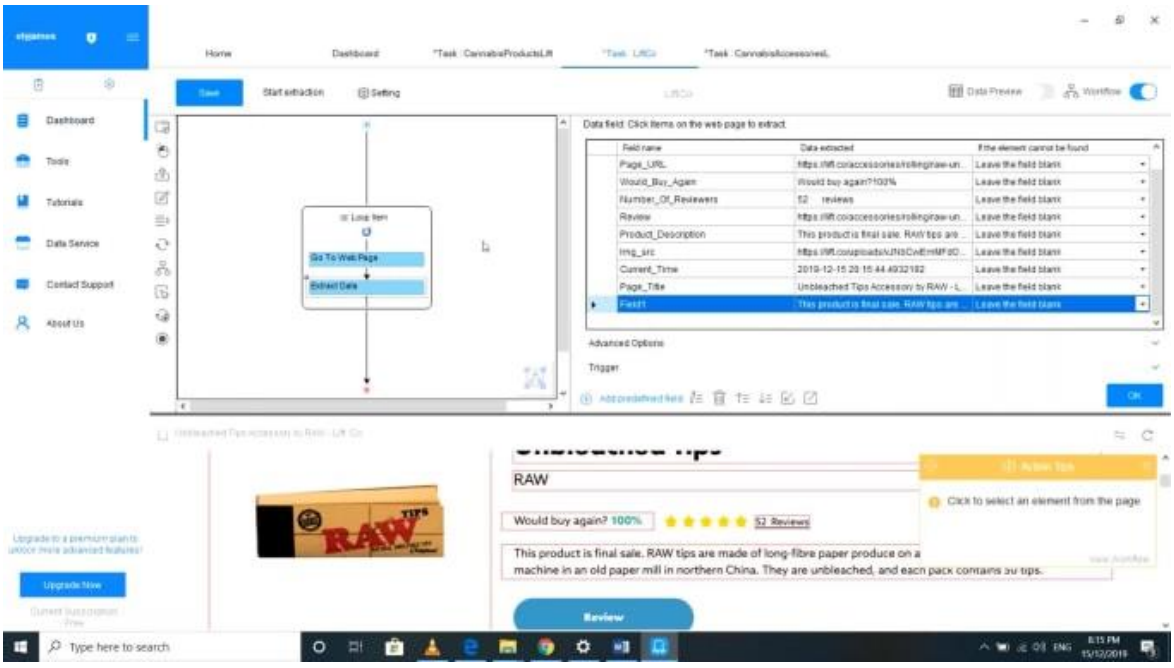

Рис. 2.2 Середовище програми «Octoparse»

Джерело: Octoparse, official website. URL: https://www.octoparse.com/

Додаток В

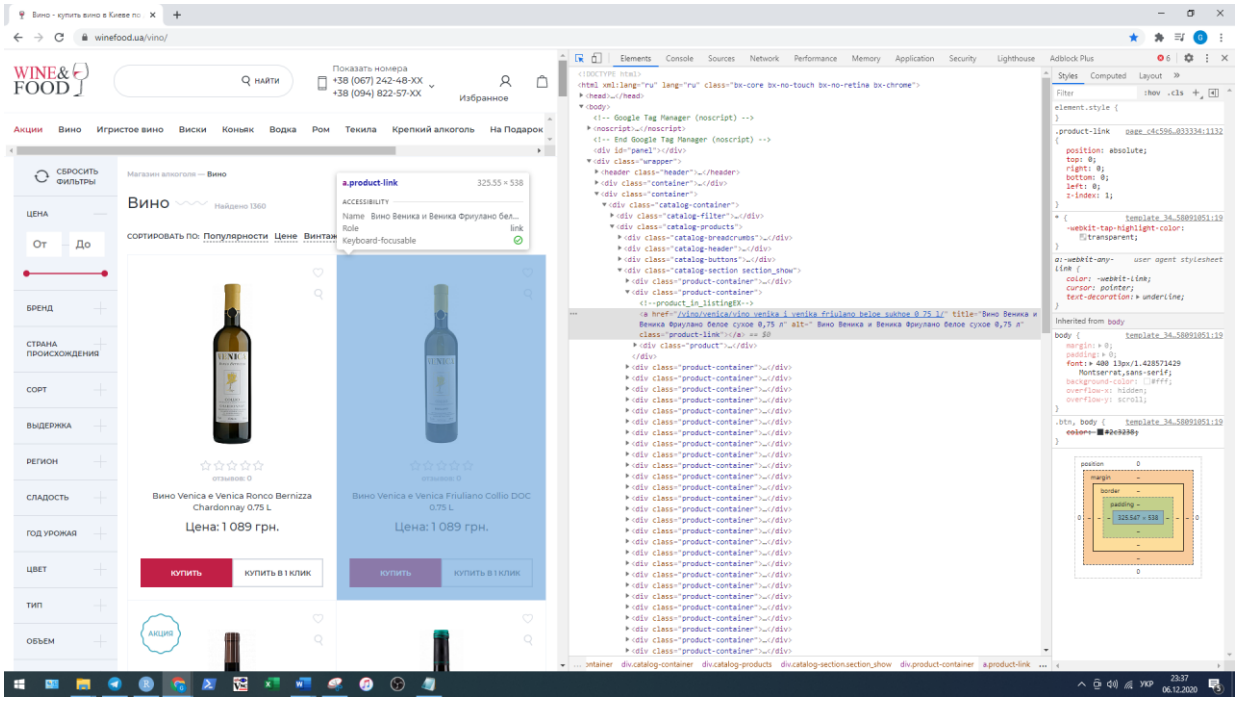

Рис. 4.1. Використання інструменту Chrome DevTools для вивчення структури сайту

Джерело: Розроблено автором.

```
> head(unlist(wineInfo))
[1] "отзывов: 0\nВино Venica e Venica Ronco Bernizza Chardonnay 0.75 L\nцена: 1 089 грн.\nкупить\nкупить в 1 клик"
[2] "отзывов: 0\nВино Venica e Venica Friuliano Collio DOC 0.75 L\nцена: 1 089 грн.\nкупить\nкупить в 1 клик"
[3] "акция\nотзывов: 0\nВино Cantina Menhir Primitivo Di Manduria 0.75 L\nцена: 899 грн.\n999 грн.\nкупить\nкупить в 1 клик"
[4] "отзывов: 0\nВино Leleka Wines Chardonnay 0,75 L\nцена: 189 грн.\nкупить\nкупить в 1 клик"
[5] "акция\nотзывов: 0\nВино Leleka Wines red semi-sweet 0,75 L\nцена: 155 грн.\n189 грн.\nкупить\nкупить в 1 клик"
[6] "отзывов: 0\nВино Leleka Wines Pinot Gris 0.75 L\nцена: 189 грн.\nкупить\nкупить в 1 клик"
```

Рис. 4.2. Сформований список з текстових векторів.

Джерело: Розроблено автором.

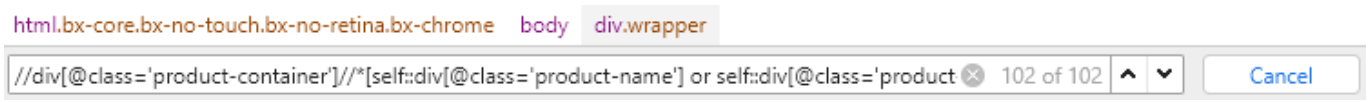

Рис. 4.3. Формування посилання на об’єкти шляхом використання xPath.

Джерело: Розроблено автором.

Додаток С

Код автоматизованого алгоритму:

```
library(RSelenium)
library(rvest)
library(magrittr)
library(tidyverse)
library(readr)
library(dplyr)

#Data scraping ----

remDr <- remoteDriver(
  remoteServerAddr = "localhost",
  port = 4445L,
  browserName = "chrome"
)

remDr$open()
remDr$maxWindowSize()

link<-"https://winefood.ua/vino/"

remDr$navigate(link)

pageState<-remDr$getPageSource()

pagehtml<-read_html(unlist(pageState))
```

```
WineBrand <- pagehtml %>% html_nodes(xpath =
"//div[@id='mCSB_1_container']/div[@class='checkbox']//span[@class='bx-filter-param-
text']/a") %>% html_text()
WineBrand

list1<-list()

for(i in 1:27){

  tryCatch({
    m2<-remDr$findElements("xpath","//div[@class='product-
container']//*[self::div[@class='product-name'] or self::div[@class='product-price']]")
    m2_text<-m2 %>% sapply(., function(x) x$getElementText())

    if(i<27){
      nextPage<-remDr$findElement("xpath", "//li[@class='bx-pag-next']/a")
      nextPage$clickElement()
      Sys.sleep(5)
    }

    list1<-c(list1, m2_text)



  },
  error = function(c) {
    i<<-i-1
    #remDr$deleteAllCookies()
    remDr$refresh()
    Sys.sleep(5)
  },
```

```
  finally={gc()}
  )

}


#Data processing ----

index<-seq(2,length(list1), by=2)

data <-  tibble('wine'=unlist(list1[-c(index)]), 'price'=unlist(list1[index]))

row1 <-  grepl("\n",data[[2]])

discout_p <-data[row1,]
regular_p <-data[!row1,]

regular_p$price<-parse_number(regular_p$price) #
https://winefood.ua/search/index.php?q=+Sileni+%D0%A1hardonnay&s=

#Split a column with both prices into two separate ones
discout_p<-discout_p %>% separate(price,c("disc", "reg"), '\n')

discout_p$disc<-parse_number(discout_p$disc)
discout_p$reg<-parse_number(discout_p$reg)


#Save data in working directory ----

setwd("C:/Users/Work/Documents/RData")
```

```
listdock<-list(list1, data, discout_p, regular_p, WineBrand)
saveRDS(listdock, file = "winecatalogue_06.rds")



#Comparison ----

# (1) Check the existence of new brands
dataset_26<-readRDS("winecatalogue_26.rds")
dataset_06<-readRDS("winecatalogue_06.rds")

setdiff(dataset_06[[5]], dataset_26[[5]]) #[1] "Leleka Wines"

# (2) Check the existence of new types of wine

list_26<-dataset_26[[2]]
list_06<-dataset_06[[2]]

#As we can see, the length of the 2 sets is different, which means that a new wine has been
added.

setdiff(list_06$wine, list_26$wine) #Here is the list of those nine.

# (3) Check the price changes

v<-setdiff(list_06$wine, list_26$wine)

#removed new components
list_06_new<-list_06[!(list_06$wine %in% v),]
```

```
#compared order and presence of all fields
all(list_06_new$wine == list_26$wine)

Union<-cbind(list_26, "price_2"=list_06_new$price)

difference<-Union[Union$price!=Union$price_2, ]
```