SECP3133-02 High Performance Data Processing

Project 1 Final Report

14/5/2025

FACULTY OF COMPUTING

Prepared By: Group 2 Hepatitis

LIM JING YONG A22EC0182

LEE SOON DER A22EC0065

JASLENE YU A22EC0171

NIK ZULAIKHAA A22EC0232

Lecturer:

DR.ARYATI BINTI BAKRI

Table of Content

1.0 Introduction **3**
1.1 Project Background 3
1.2 Objectives 3
1.3 Target Website & Data to be Extracted 3
2.0 System Design and Architecture **6**
2.1 Architecture 6
2.2 Tools and Framework used 7
2.3 Roles of team members 8
3.0 Data Collection **9**
3.1 Crawling method 9
3.2 Number of records collected 9
3.3 Ethical considerations 10
4.0 Data Processing **11**
4.1 Cleaning methods 11
4.2 Data Structure 11
4.3 Transformation and formatting 13
5.0 Optimization Techniques **15**
5.1 Methods used 15
5.2 Code Overview 16
6.0 Performance Evaluation **18**
6.1 Before optimization 18
6.2 Comparison to Optimized Solution 19
6.3 Graph Visualization 20
7.0 Challenges & Limitations **23**
7.1 Challenges 23
7.2 Limitations 23
8.0 Conclusion **25**
8.1 Our Findings 25
8.2 Improvement to be done 26
Appendix **27**

1.0 Introduction

1.1 Project Background

As technology of modern days grows, speed and effectiveness of computer processes has become significant to those who require data to work, which practically includes almost every existing field in the market. High performance data processing, being one of the more recent technology innovations meant to accelerate compute rate, has become an important topic in data analytics. Hence, from this project, we would like to evaluate the practicality of said innovation:

1. Are high performance computing (HPC) solutions truly accelerating the computing processes, specifically the data cleaning process?
2. How does each python library (Modin, Dask, Joblib etc) differ in the ways they handle HPC?
3. Which library delivers the best performance when dealing with 100k+ datasets?

Through this project, we want to make a conclusion from the listed questions above, and potentially help answer questions by more IT users or business users, whom may be interested in discovering pros and cons of each optimization library.

1.2 Objectives

- To develop a web scraping system using several libraries, such as BeautifulSoup, Selenium and more to extract up to 100,000 real estate data from the iProperty Malaysia website.
- To perform heavy loaded data cleaning on said dataset.
- To utilize four libraries with optimization techniques including multithreading, multiprocessing, Spark, etc. to enhance data processing efficiency
- To evaluate and compare performance before and after optimization based on:
    - Time
    - Memory usage
    - CPU usage
    - Throughput
- To visualize insights and performance metrics using chart and graphs for clear comparison

1.3 Target Website & Data to be Extracted

The target website for data scraping in this project is iProperty Malaysia, a well-known online real estate platform that provides listings for various residential properties that are available for sale across Malaysia. The site offers detailed information on each property, including its location, price, size, furnishing status, type and the contact agent responsible for the listing.

The data extracted from iProperty includes the following key attributes:

1. Property title (e.g., apartment, condominium, terrace house)
2. Location (state and city)
3. Price
4. Built-up size (in square feet)
5. Furnishing status (furnished, partially furnished or unfurnished)
6. Agent name

This information allows for valuable insights into the Malaysian property market and also provides a robust dataset to evaluate the performance of different high-performance data processing techniques in later stages of the project.

Note: There exist 3 different card styles for properties based on time of publish

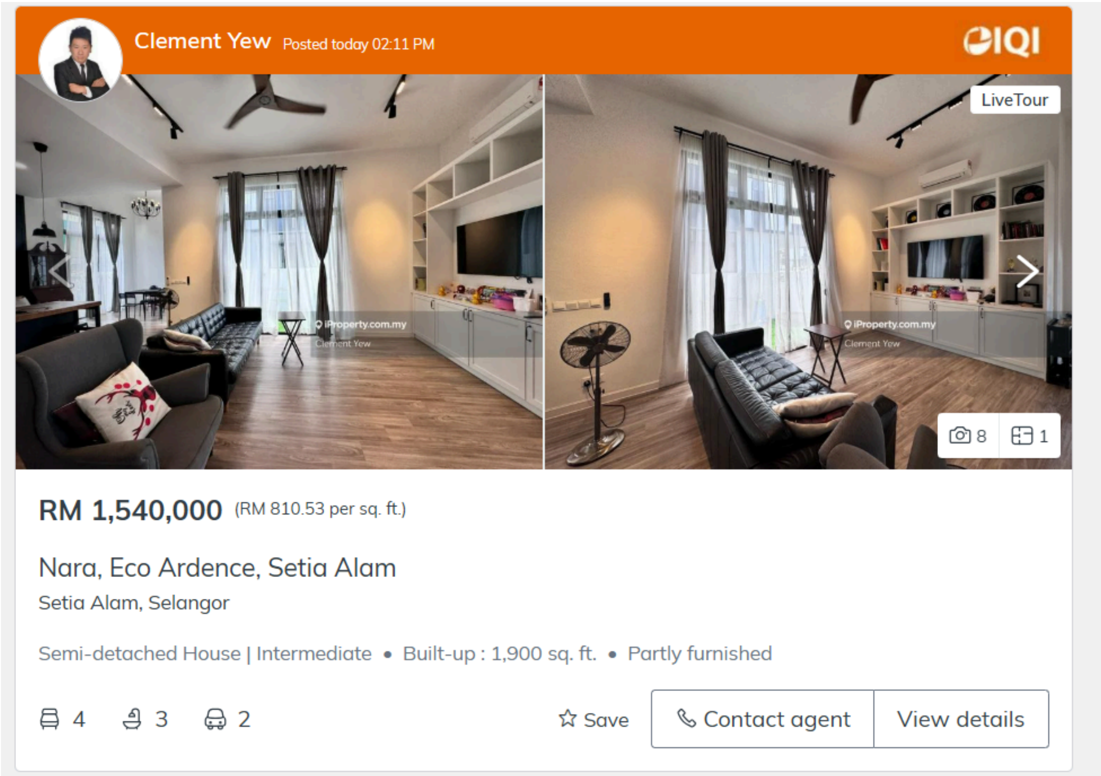


Figure 1.3.1 Example premium style containers in iProperty website

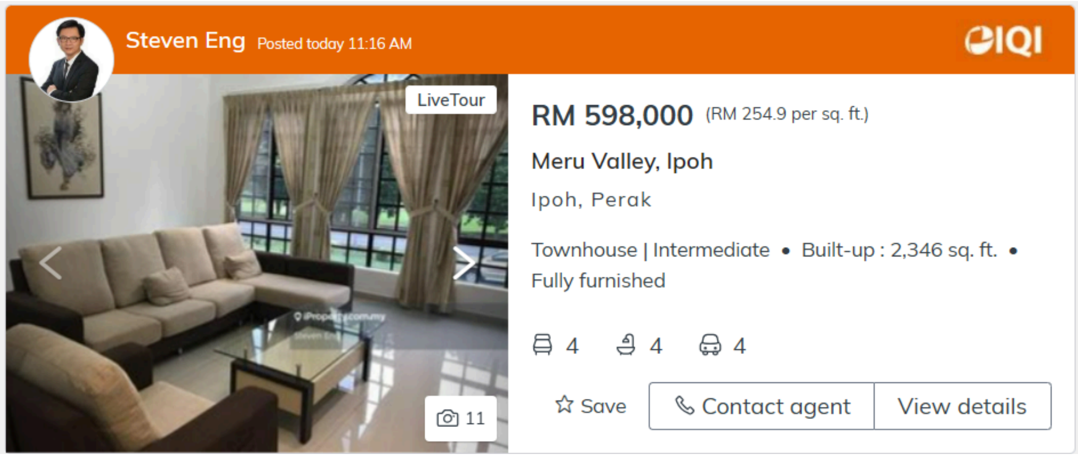


Figure 1.3.2 Example featured style containers in iProperty website

Figure 1.3.3 Example basic style containers in iProperty website

2.0 System Design and Architecture

2.1 Architecture

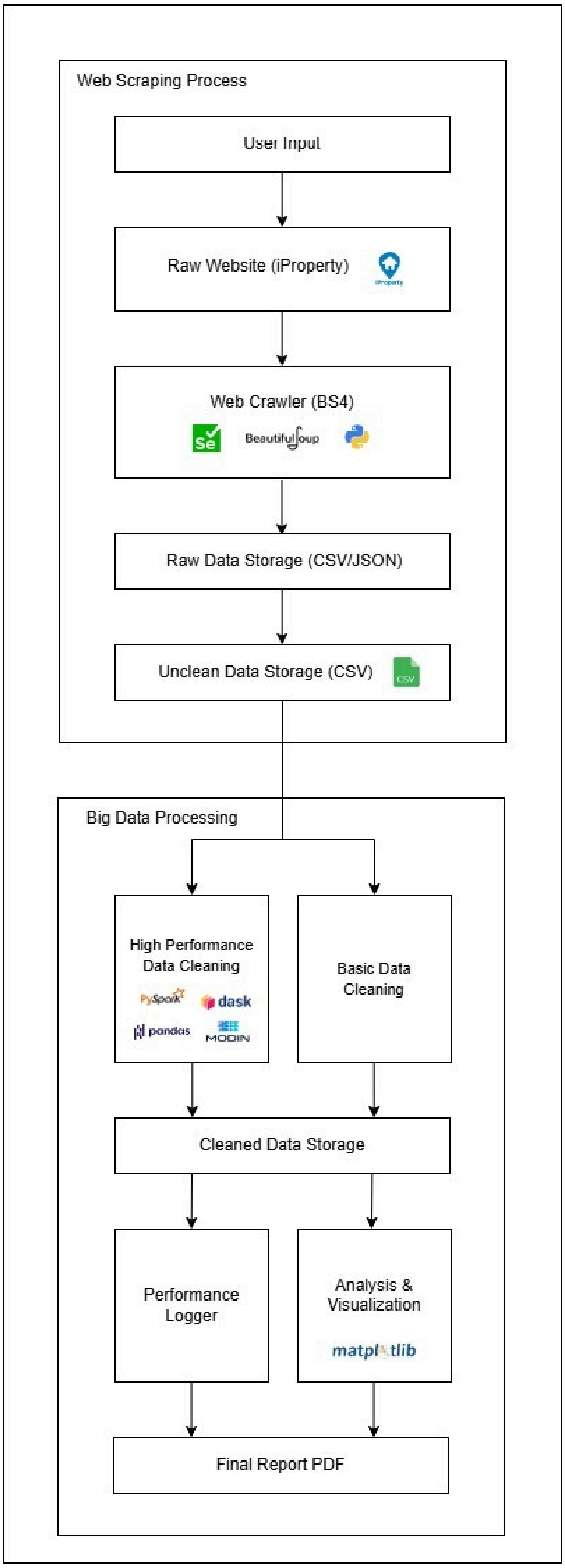


Figure 2.1 Architecture Design of our System

2.2 Tools and Framework used

Category	Software/Application/Library/Website
Documentation	Google Doc
Progression Monitoring	Github
Architecture Design	Draw.io
Web Scraping Library	BeautifulSoup, Selenium
Initial Dataset Form	Excel file (.csv)
IDE	Google Colab
Coding Language	Python
Data Visualization	Matplotlib Library (Python)
Basic Data Cleaning	Pandas Library (Python)
Optimization Libraries	Modin & Ray, Polars, Spark, Joblib, Multiprocessing
Database Management System	SQLite

Table 2.2 Tools used in this Project

2.3 Roles of team members

Member	Role	Responsibility
Lim Jing Yong	Lead Crawler Developer	• Leading the implementation of the web crawler using BeautifulSoup and Selenium • Handle pagination, data structure identification, and crawl delay management • Ensure ethical scraping practices and data completeness • Optimization library used: **Modin & Ray**
Lee Soon Der	Data Cleaning and Storage Specialist	• Leading the cleaning and preprocessing of the raw data • Standardize fields and manage datasets for further processing • Optimization library used: **Polars**
Jaslene Yu	Performance and Optimization Lead	• Leading the process applying high performance computing techniques • Monitor and document CPU/memory usage and system throughput • Optimization library used: **Multiprocessing & Spark**
Nik Zulaikhaa	Performance Analyst and Documentation Lead	• Leading the comparison of optimization result • Generate visualizations and compile the final technical report • Optimization library used: **Joblib**

Table 2.3 Member Roles

3.0 Data Collection

3.1 Crawling method

To collect the required property data from iProperty Malaysia, we implemented a **web crawling system** by using **Python** by combining the **Selenium** library with **BeautifulSoup** for HTML parsing. Selenium was used to simulate a real browser environment that handles dynamic content loading. BeautifulSoup was implemented to extract and parse the HTML structure of the web pages.

The crawling process follows a **dynamic web scraping approach**, as the targeted website, the iProperty requires JavaScript execution to fully render the property listings. We automated **Chromium** using a **headless** mode setup, enhanced with **fake user agents** to mimic real browsing behavior and avoid bot detection.

During the crawling process, **CSS selectors** with partial class matching were used to accurately scrape the property title, price, location, details, and agent name. The data was collected from **multiple pages** across **different areas**, and the final output was compiled into a CSV file.

3.2 Number of records collected

A total of **160,000+ records** of properties are recorded.

1	Area	Property Title	Property Price	Property Location	Property Details	Property Agent	Source URL
2	perlis-zop7y	Semi D 2 Tingkat - Taman Jaya Diri - Seriab -	RM 775,776	Kangar, Perlis	Semi-detached House | Intermediate Â â€¢Â Built-u	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
3	perlis-zop7y	Semi D 1 Tingkat - Taman Nyu Indah 2 - Arau	RM 398,000	Arau, Perlis	Semi-detached House Â â€¢Â Built-up : 1,021 sq. ft.	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
4	perlis-zop7y	Taman Seri Manis Dua, Kangar	RM 306,000	Kangar, Perlis	Semi-detached House Â â€¢Â Built-up : 1,938 sq. ft.	Joyce Chan	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
5	perlis-zop7y	Teres 1 Tingkat - Bandar Baharu Putra Heigh	RM 185,000	Arau, Perlis	1-sty Terrace/Link House | Intermediate Â â€¢Â Built	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
6	perlis-zop7y	Teres 1 Tingkat - Bandar Baharu Putra Heigh	RM 210,000	Arau, Perlis	1-sty Terrace/Link House | Intermediate Â â€¢Â Built	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
7	perlis-zop7y	Taman Seri Bintong Maju, Kangar	RM 247,000	Kangar, Perlis	1-sty Terrace/Link House | Corner lot Â â€¢Â Built-u	Leslie Low	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
8	perlis-zop7y	Taman Tengku Budriah, Arau	RM 416,000	Arau, Perlis	Semi-detached House Â â€¢Â Land area : 4,884 sq. ft	Leslie Low	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
9	perlis-zop7y	Rumah Berkembar Setingkat di Jejawi, Perlis	RM 441,000	Kangar, Perlis	Semi-detached House Â â€¢Â Built-up : 1,150 sq. ft.	Lim	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
10	perlis-zop7y	Taman Kuala Perlis, Kuala Perlis	RM 220,000	Kuala Perlis, Perlis	2-sty Terrace/Link House | Intermediate Â â€¢Â Built	Adam Wong	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
11	perlis-zop7y	Teres (1) Setingkat di Irama Residensi - Kang	RM 298,000	Kangar, Perlis	1-sty Terrace/Link House Â â€¢Â Built-up : 990 sq. ft.	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
12	perlis-zop7y	Taman Sri Pauh, Arau	RM 171,000	Arau, Perlis	1-sty Terrace/Link House Â â€¢Â Land area : 1,195 sc	Joyce Chan	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
13	perlis-zop7y	Taman Empiang Ceria, Kangar	RM 409,920	Kangar, Perlis	Semi-detached House | Intermediate Â â€¢Â Built-u	Shafina	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
14	perlis-zop7y	Padang Biak, Arau	RM 3,147,210	Arau, Perlis	Residential Land Â â€¢Â Land area : 2.89 acres	Dickson Lum	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
15	perlis-zop7y	Teres 2 Tingkat - Taman Bunga Padi - Kanga	RM 180,000	Kangar, Perlis	2-sty Terrace/Link House | Intermediate Â â€¢Â Built	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
16	perlis-zop7y	Rumah Berkembar 1 Tingkat - Taman Sri Gu	RM 385,000	Kangar, Perlis	Semi-detached House | Intermediate Â â€¢Â Built-u	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
17	perlis-zop7y	Teres 2 Tingkat - Taman Kemajuan - Kangar,	RM 250,000	Kangar, Perlis	2-sty Terrace/Link House | End lot Â â€¢Â Built-up :	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
18	perlis-zop7y	Taman Padang Behor, Kangar	RM 370,000	Kangar, Perlis	Semi-detached House Â â€¢Â Built-up : 960 sq. ft.	Nur Suhaini	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
19	perlis-zop7y	Kangar	RM 450,000	Kangar, Perlis	Semi-detached House | Corner lot Â â€¢Â Built-up :	Nur Suhaini	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
20	perlis-zop7y	Kangar	RM 385,000	Kangar, Perlis	Semi-detached House | Intermediate Â â€¢Â Built-u	Nur Suhaini	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
21	perlis-zop7y	(BAHARU) Taman Enyum Indah - Arau - Perli	RM 355,000	Arau, Perlis	1-sty Terrace/Link House Â â€¢Â Built-up : 880 sq. ft.	Haneef	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
22	perlis-zop7y	Taman Utara Jaya 1,, Kuala Perlis	RM 398,000	Kuala Perlis, Perlis	Semi-detached House | Intermediate Â â€¢Â Built-u	Elwin	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
23	perlis-zop7y	Pauh, Arau	RM 6,970,000	Arau, Perlis	Residential Land Â â€¢Â Land area : 9.7 acres	Sally Chin	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
24	perlis-zop7y	Kangar,Perlis., Kangar	RM 750,000	Kangar, Perlis	Semi-detached House | Intermediate Â â€¢Â Built-u	Michael Lee	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
25	perlis-zop7y	Jalan Padang Katong, Kangar	RM 750,000	Kangar, Perlis	Semi-detached House | Intermediate Â â€¢Â Built-u	Michael Lee	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
26	perlis-zop7y	Rumah Berkembar Setingkat di Jejawi Perlis,	RM 415,000	Kangar, Perlis	Semi-detached House | Intermediate Â â€¢Â Built-u	Mk Sangaran	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
27	perlis-zop7y	Taman Hijrah, Kangar	RM 350,000	Kangar, Perlis	1-sty Terrace/Link House | Corner lot Â â€¢Â Built-u	Muaz Azizan	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/
28	perlis-zop7y	Taman Temak Jaya, Seriab	RM 395,000	Seriab, Perlis	2-sty Terrace/Link House | Intermediate Â â€¢Â Built	Faizals	https://www.iproperty.com.my/sale/perlis-zop7y/all-residential/

new_iproperty_dataset

Figure 3.2.1 First 27 rows of data in uncleaned dataset

163512	simpang-ampat-41p	Bandar Tasek Mutiara, Simpang Ampat	RM 6,546,428	Simpang Ampat, Penang	Residential Land Â â€¢Â Land area : 96,271 sq. ft.	Crystal Tee	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163513	simpang-ampat-41p	Bandar Tasek Mutiara, Simpang Ampat	RM 9,980,292	Simpang Ampat, Penang	Residential Land Â â€¢Â Land area : 146,769 sq. ft.	Crystal Tee	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163514	simpang-ampat-41p	Bandar Tasek Mutiara, Simpang Ampat	RM 9,149,536	Simpang Ampat, Penang	Residential Land Â â€¢Â Land area : 134,552 sq. ft.	Crystal Tee	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163515	simpang-ampat-41p	Bandar Tasek Mutiara, Simpang Ampat	RM 12,579,252	Simpang Ampat, Penang	Residential Land Â â€¢Â Land area : 184,989 sq. ft.	Crystal Tee	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163516	simpang-ampat-41p	Royale Infinity, Simpang Ampat	RM 380,000	Simpang Ampat, Penang	Condominium Â â€¢Â Built-up : 1,480 sq. ft.	Jass Kong	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163517	simpang-ampat-41p	Simpang Ampat	RM 570,000	Simpang Ampat, Penang	Semi-detached House Â â€¢Â Built-up : 1,585 sq. ft.	Jetson Yip	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163518	simpang-ampat-41p	Hijauan Hill, Simpang Ampat	RM 650,000	Simpang Ampat, Penang	Semi-detached House Â â€¢Â Built-up : 2,280 sq. ft.	Anders Ong	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163519	simpang-ampat-41p	Batu Kawan, Simpang Ampat	RM 858,000	Simpang Ampat, Penang	2-sty Terrace/Link House Â â€¢Â Built-up : 2,290 sq.	Chris Huah	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163520	simpang-ampat-41p	Bandar Tasek Mutiara, Simpang Ampat	RM 261,000	Simpang Ampat, Penang	1-sty Terrace/Link House Â â€¢Â Land area : 1,195 sc	Koay CK	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163521	simpang-ampat-41p	Taman Lembah Indah, Simpang Ampat	RM 1,400,000	Simpang Ampat, Penang	Bungalow Â â€¢Â Built-up : 4,500 sq. ft.	Steven Hng	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163522	simpang-ampat-41p	Royale Infinity, Simpang Ampat	RM 680,000	Simpang Ampat, Penang	Condominium | Intermediate Â â€¢Â Built-up : 1,400	Kelly Yee	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163523	simpang-ampat-41p	Taman Lembah Indah, Simpang Ampat	RM 660,000	Simpang Ampat, Penang	3-sty Terrace/Link House Â â€¢Â Built-up : 2,524 sq.	Winson Chong	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163524	simpang-ampat-41p	Taman residensi alma, Simpang Ampat	RM 700,000	Simpang Ampat, Penang	2-sty Terrace/Link House Â â€¢Â Built-up : 1,640 sq.	Suzzane Lee	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163525	simpang-ampat-41p	Taman Idaman, Simpang Ampat	RM 810,000	Simpang Ampat, Penang	2-sty Terrace/Link House | Corner lot Â â€¢Â Built-u	CH Ong	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163526	simpang-ampat-41p	Taman Perwira Indah, Simpang Ampat	RM 430,000	Simpang Ampat, Penang	2-sty Terrace/Link House Â â€¢Â Built-up : 2,000 sq.	Fionne Seah	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163527	simpang-ampat-41p	Taman Perwira, Simpang Ampat	RM 315,000	Simpang Ampat, Penang	1-sty Terrace/Link House Â â€¢Â Built-up : 1,200 sq.	Louis Ng	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163528	simpang-ampat-41p	Bandar Tasek Mutiara, Simpang Ampat	RM 550,000	Simpang Ampat, Penang	2-sty Terrace/Link House | Intermediate Â â€¢Â Buil	Alex Ho	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163529	simpang-ampat-41p	Taman Lembah Indah, Simpang Ampat	RM 866,000	Simpang Ampat, Penang	3-sty Terrace/Link House | Corner lot Â â€¢Â Built-u	CH Ong	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163530	simpang-ampat-41p	Raintree Park, Simpang Ampat	RM 480,000	Simpang Ampat, Penang	2-sty Terrace/Link House | Intermediate Â â€¢Â Buil	Francis Neow	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163531	simpang-ampat-41p	Taman Tambun Emas, Simpang Ampat	RM 699,000	Simpang Ampat, Penang	3-sty Terrace/Link House | Corner lot Â â€¢Â Built-u	Nicol Tan	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163532	simpang-ampat-41p	Taman Murai Jaya, Simpang Ampat	RM 680,000	Simpang Ampat, Penang	Semi-detached House | Intermediate Â â€¢Â Built-up	CH Ong	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163533	simpang-ampat-41p	Bandar Tasek Mutiara, Simpang Ampat	RM 950,000	Simpang Ampat, Penang	Semi-detached House | Corner lot Â â€¢Â Built-up :	CH Ong	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163534	simpang-ampat-41p	Taman Lembah Indah, Simpang Ampat	RM 608,000	Simpang Ampat, Penang	2-sty Terrace/Link House | Intermediate Â â€¢Â Buil	Jetson Yip	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163535	simpang-ampat-41p	Taman Eco Meadow, Simpang Ampat	RM 850,000	Simpang Ampat, Penang	2-sty Terrace/Link House | Intermediate Â â€¢Â Buil	Jass Ooi	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163536	simpang-ampat-41p	Taman Tambun Indah, Simpang Ampat	RM 1,600,000	Simpang Ampat, Penang	Bungalow | Intermediate Â â€¢Â Built-up : 3,016 sq.	Francis Neow	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163537	simpang-ampat-41p	Bandar Tasek Mutiara, Simpang Ampat	RM 690,000	Simpang Ampat, Penang	Semi-detached House | Intermediate Â â€¢Â Built-up	Nicol Tan	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163538	simpang-ampat-41p	Eco Bloom, Simpang Ampat	RM 470,000	Simpang Ampat, Penang	Condominium | Intermediate Â â€¢Â Built-up : 901 s	Andy Chuah	https://www.iproperty.com.my/sale/simpang-ampat-41pqd/all
163539							

new_iproperty_dataset

Figure 3.2.2 Last 27 rows of data in uncleaned dataset (until 163538th row)

The dataset has the following fields as below:

1. Area - The state or city where the property is located.
2. Property Title - The title or headline of the property listing.
3. Property Price - The listed selling price of the property.
4. Property Location - A more specific address of the property.
5. Property Details - Detailed information about the property include size, type and furnishing.
6. Property Agent - The name of the real estate agent or agency responsible for selling or listing the property.
7. Source URL - The website link to the original source page.

3.3 Ethical considerations

To ensure ethical web scraping, we followed these key practices:

- Polite Crawling: Crawl delays and request throttling were applied to avoid overloading the server.
- No Sensitive Data: Only publicly available property listing information was collected. No personal or confidential data was accessed.
- Academic Use Only: All data is used strictly for educational purposes within the course scope and will not be used commercially.
- Secure Data Handling: The dataset is securely stored and not publicly shared.

4.0 Data Processing

4.1 Cleaning methods

Several cleaning methods are utilized to transition our newly built dataset of 160k+ data into a more readable format to help ease the process of analysis. More details regarding data cleaning can be referred to in section 4.3.

Method	Involved Field	Description
Data Splitting	Property Title	By comma
	Property Location	By comma
	Property Detail	By separator
Title Casing	Property Title	First character uppercase
	Property Agent	First character uppercase
Punctuation Removal	Property Price	"RM", comma
Data Type Conversion	Property Price	string to int
Special character replacement	Property Details	Multiple characters to "|"
Numeric Value Extraction	Property Details	Extract property size
Filling Missing Values	Property Details	As "None"
Row Filtering	Property Agent	Remove non-agent name data
Null Data Removal	Entire Dataset	Remove data based on condition

Table 4.1 Data Cleaning Methods and related Fields

4.2 Data Structure

The cleaned property dataset was stored in an **SQLite database** for efficient data management.

A copy of the original DataFrame was created to preserve the raw data. The cleaned data was then saved into the SQLite database using the **to_sql()** method, where it was stored in a table named **cleaned_data**. If the table already existed, it was replaced with the new data.

The database file (**iproperty.db**) was then exported and made available for download in Google Drive. This allows easy access to the cleaned data for further analysis in SQLite-compatible applications.

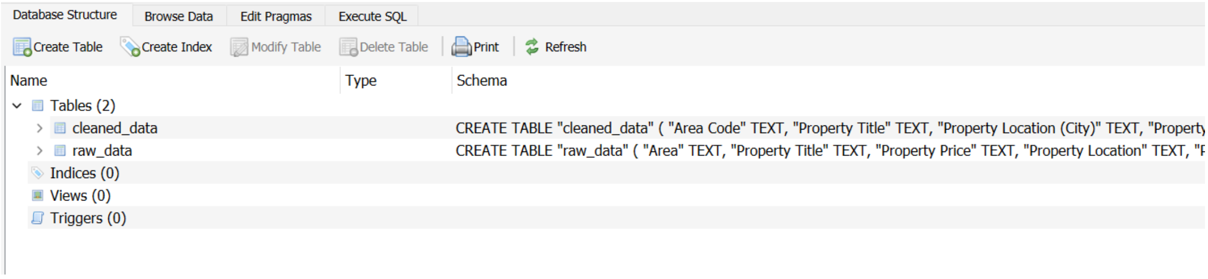
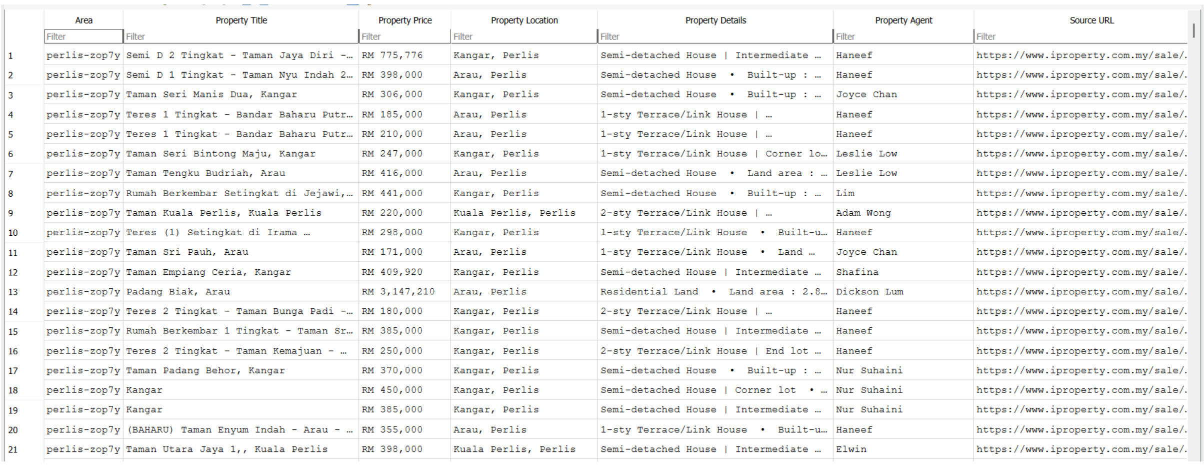
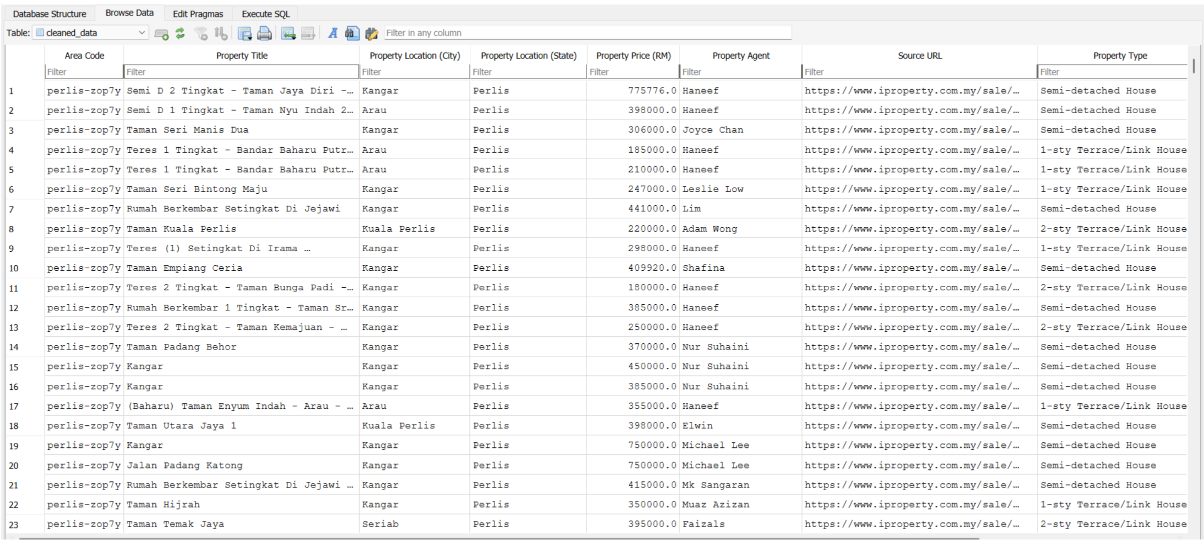


Figure 4.2.1 Database Structure in SQLite



	Area	Property Title	Property Price	Property Location	Property Details	Property Agent	Source URL
1	perlis-zop7y	Semi D 2 Tingkat - Taman Jaya Diri -…	RM 775,776	Kangar, Perlis	Semi-detached House | Intermediate …	Haneef	https://www.iproperty.com.my/sale/.
2	perlis-zop7y	Semi D 1 Tingkat - Taman Nyu Indah 2…	RM 398,000	Arau, Perlis	Semi-detached House • Built-up : …	Haneef	https://www.iproperty.com.my/sale/.
3	perlis-zop7y	Taman Seri Manis Dua, Kangar	RM 306,000	Kangar, Perlis	Semi-detached House • Built-up : …	Joyce Chan	https://www.iproperty.com.my/sale/.
4	perlis-zop7y	Teres 1 Tingkat - Bandar Baharu Putr…	RM 185,000	Arau, Perlis	1-sty Terrace/Link House | …	Haneef	https://www.iproperty.com.my/sale/.
5	perlis-zop7y	Teres 1 Tingkat - Bandar Baharu Putr…	RM 210,000	Arau, Perlis	1-sty Terrace/Link House | …	Haneef	https://www.iproperty.com.my/sale/.
6	perlis-zop7y	Taman Seri Bintong Maju, Kangar	RM 247,000	Kangar, Perlis	1-sty Terrace/Link House | Corner lo…	Leslie Low	https://www.iproperty.com.my/sale/.
7	perlis-zop7y	Taman Tengku Budriah, Arau	RM 416,000	Arau, Perlis	Semi-detached House • Land area : …	Leslie Low	https://www.iproperty.com.my/sale/.
8	perlis-zop7y	Rumah Berkembar Setingkat di Jejawi,…	RM 441,000	Kangar, Perlis	Semi-detached House • Built-up : …	Lim	https://www.iproperty.com.my/sale/.
9	perlis-zop7y	Taman Kuala Perlis, Kuala Perlis	RM 220,000	Kuala Perlis, Perlis	2-sty Terrace/Link House | …	Adam Wong	https://www.iproperty.com.my/sale/.
10	perlis-zop7y	Teres (1) Setingkat di Irama …	RM 298,000	Kangar, Perlis	1-sty Terrace/Link House • Built-u…	Haneef	https://www.iproperty.com.my/sale/.
11	perlis-zop7y	Taman Sri Pauh, Arau	RM 171,000	Arau, Perlis	1-sty Terrace/Link House • Land …	Joyce Chan	https://www.iproperty.com.my/sale/.
12	perlis-zop7y	Taman Empiang Ceria, Kangar	RM 409,920	Kangar, Perlis	Semi-detached House | Intermediate …	Shafina	https://www.iproperty.com.my/sale/.
13	perlis-zop7y	Padang Biak, Arau	RM 3,147,210	Arau, Perlis	Residential Land • Land area : 2.8…	Dickson Lum	https://www.iproperty.com.my/sale/.
14	perlis-zop7y	Teres 2 Tingkat - Taman Bunga Padi -…	RM 180,000	Kangar, Perlis	2-sty Terrace/Link House | …	Haneef	https://www.iproperty.com.my/sale/.
15	perlis-zop7y	Rumah Berkembar 1 Tingkat - Taman Sr…	RM 385,000	Kangar, Perlis	Semi-detached House | Intermediate …	Haneef	https://www.iproperty.com.my/sale/.
16	perlis-zop7y	Teres 2 Tingkat - Taman Kemajuan - …	RM 250,000	Kangar, Perlis	2-sty Terrace/Link House | End lot …	Haneef	https://www.iproperty.com.my/sale/.
17	perlis-zop7y	Taman Padang Behor, Kangar	RM 370,000	Kangar, Perlis	Semi-detached House • Built-up : …	Nur Suhaini	https://www.iproperty.com.my/sale/.
18	perlis-zop7y	Kangar	RM 450,000	Kangar, Perlis	Semi-detached House | Corner lot • …	Nur Suhaini	https://www.iproperty.com.my/sale/.
19	perlis-zop7y	Kangar	RM 385,000	Kangar, Perlis	Semi-detached House | Intermediate …	Nur Suhaini	https://www.iproperty.com.my/sale/.
20	perlis-zop7y	(BAHARU) Taman Enyum Indah - Arau - …	RM 355,000	Arau, Perlis	1-sty Terrace/Link House • Built-u…	Haneef	https://www.iproperty.com.my/sale/.
21	perlis-zop7y	Taman Utara Jaya 1,, Kuala Perlis	RM 398,000	Kuala Perlis, Perlis	Semi-detached House | Intermediate …	Elwin	https://www.iproperty.com.my/sale/.

Figure 4.2.2 Raw data table in SQLite



	Area Code	Property Title	Property Location (City)	Property Location (State)	Property Price (RM)	Property Agent	Source URL	Property Type
1	perlis-zop7y	Semi D 2 Tingkat - Taman Jaya Diri -…	Kangar	Perlis	775776.0	Haneef	https://www.iproperty.com.my/sale/…	Semi-detached House
2	perlis-zop7y	Semi D 1 Tingkat - Taman Nyu Indah 2…	Arau	Perlis	398000.0	Haneef	https://www.iproperty.com.my/sale/…	Semi-detached House
3	perlis-zop7y	Taman Seri Manis Dua	Kangar	Perlis	306000.0	Joyce Chan	https://www.iproperty.com.my/sale/…	Semi-detached House
4	perlis-zop7y	Teres 1 Tingkat - Bandar Baharu Putr…	Arau	Perlis	185000.0	Haneef	https://www.iproperty.com.my/sale/…	1-sty Terrace/Link House
5	perlis-zop7y	Teres 1 Tingkat - Bandar Baharu Putr…	Arau	Perlis	210000.0	Haneef	https://www.iproperty.com.my/sale/…	1-sty Terrace/Link House
6	perlis-zop7y	Taman Seri Bintong Maju	Kangar	Perlis	247000.0	Leslie Low	https://www.iproperty.com.my/sale/…	1-sty Terrace/Link House
7	perlis-zop7y	Rumah Berkembar Setingkat Di Jejawi	Kangar	Perlis	441000.0	Lim	https://www.iproperty.com.my/sale/…	Semi-detached House
8	perlis-zop7y	Taman Kuala Perlis	Kuala Perlis	Perlis	220000.0	Adam Wong	https://www.iproperty.com.my/sale/…	2-sty Terrace/Link House
9	perlis-zop7y	Teres (1) Setingkat Di Irama …	Kangar	Perlis	298000.0	Haneef	https://www.iproperty.com.my/sale/…	1-sty Terrace/Link House
10	perlis-zop7y	Taman Empiang Ceria	Kangar	Perlis	409920.0	Shafina	https://www.iproperty.com.my/sale/…	Semi-detached House
11	perlis-zop7y	Teres 2 Tingkat - Taman Bunga Padi -…	Kangar	Perlis	180000.0	Haneef	https://www.iproperty.com.my/sale/…	2-sty Terrace/Link House
12	perlis-zop7y	Rumah Berkembar 1 Tingkat - Taman Sr…	Kangar	Perlis	385000.0	Haneef	https://www.iproperty.com.my/sale/…	Semi-detached House
13	perlis-zop7y	Teres 2 Tingkat - Taman Kemajuan - …	Kangar	Perlis	250000.0	Haneef	https://www.iproperty.com.my/sale/…	2-sty Terrace/Link House
14	perlis-zop7y	Taman Padang Behor	Kangar	Perlis	370000.0	Nur Suhaini	https://www.iproperty.com.my/sale/…	Semi-detached House
15	perlis-zop7y	Kangar	Kangar	Perlis	450000.0	Nur Suhaini	https://www.iproperty.com.my/sale/…	Semi-detached House
16	perlis-zop7y	Kangar	Kangar	Perlis	385000.0	Nur Suhaini	https://www.iproperty.com.my/sale/…	Semi-detached House
17	perlis-zop7y	(Baharu) Taman Enyum Indah - Arau - …	Arau	Perlis	355000.0	Haneef	https://www.iproperty.com.my/sale/…	1-sty Terrace/Link House
18	perlis-zop7y	Taman Utara Jaya 1	Kuala Perlis	Perlis	398000.0	Elwin	https://www.iproperty.com.my/sale/…	Semi-detached House
19	perlis-zop7y	Kangar	Kangar	Perlis	750000.0	Michael Lee	https://www.iproperty.com.my/sale/…	Semi-detached House
20	perlis-zop7y	Jalan Padang Katong	Kangar	Perlis	750000.0	Michael Lee	https://www.iproperty.com.my/sale/…	Semi-detached House
21	perlis-zop7y	Rumah Berkembar Setingkat Di Jejawi …	Kangar	Perlis	415000.0	Mk Sangaran	https://www.iproperty.com.my/sale/…	Semi-detached House
22	perlis-zop7y	Taman Hijrah	Kangar	Perlis	350000.0	Muaz Azizan	https://www.iproperty.com.my/sale/…	1-sty Terrace/Link House
23	perlis-zop7y	Taman Temak Jaya	Seriab	Perlis	395000.0	Faizals	https://www.iproperty.com.my/sale/…	2-sty Terrace/Link House

Figure 4.2.3 Cleaned data table in SQLite

4.3 Transformation and formatting

Reformatting Property Titles:

```
df['Property Title'] = df['Property Title'].str.split(',').str[0]
df['Property Title'] = df['Property Title'].str.title()
```

- Capitalized the first letter of each word.
- Separated titles by commas for improved readability.

Dividing Property Locations:

```
split_cols = df['Property Location'].str.split(', ', expand=True)
split_cols.columns = ['Property Location (City)', 'Property Location (State)']
df.drop(columns=['Property Location'], inplace=True)
df.insert(2, 'Property Location (City)', split_cols['Property Location (City)'])
df.insert(3, 'Property Location (State)', split_cols['Property Location (State)'])
```

- Split into two components: city and state.

Handling Property Prices:

```
df['Property Price'] = df['Property Price'].str.replace('RM', '', regex=False)
df['Property Price'] = df['Property Price'].str.replace(',', '', regex=False)
df['Property Price'] = pd.to_numeric(df['Property Price'], errors='coerce')
df.rename(columns={'Property Price': 'Property Price (RM)'}, inplace=True)
```

- Removed entries with NaN values to ensure data integrity.
- Converted prices from string format to integers.
- Renamed the column to "rm" and separated data by commas.

Standardizing Property Agent Names:

```
df['Property Agent'] = df['Property Agent'].str.title()
df['Property Agent'] = df['Property Agent'].apply(
    lambda x: None if re.search(r'\bsdn\.?\s*bhd\.?\b', str(x), re.IGNORECASE) else x
)
```

- Capitalized the first letter of each agent's name.
- Filtered out non-agent names (e.g., company names).

Splitting Property Details:

```
def parse_property_details(detail):
    if pd.isnull(detail):
        return pd.Series([None, None, None])

    # clean messy code (eg: Â â€¢Â )
    clean_detail = re.sub(r'[^\x00-\x7F]+', ' ', str(detail))

    # extract Type (all character before Built-up)
    type_match = re.search(r"^(.*?)Built-up", clean_detail, re.IGNORECASE)
    if type_match:
        raw_type = type_match.group(1).strip()
        property_type = re.split(r'\s*\|\s*', raw_type)[0]
    else:
        property_type = None

    # extract Area
    area_match = re.search(r'Built[-\s]*up[^0-9]*([\d,]+)\s*sq\.?\s*ft', clean_detail, re.IGNORECASE)
    area = area_match.group(1).replace(',', '') if area_match else None

    # extract Furnishing status
    furnishing = "Unknown"
    if re.search(r'\bUnfurnished\b', clean_detail, re.IGNORECASE):
        furnishing = 'Unfurnished'
    elif re.search(r'\bPartially Furnished\b', clean_detail, re.IGNORECASE):
        furnishing = 'Partially Furnished'
    elif re.search(r'\bFully Furnished\b', clean_detail, re.IGNORECASE):
        furnishing = 'Fully Furnished'
    elif re.search(r'\bFurnished\b', clean_detail, re.IGNORECASE):
        furnishing = 'Furnished'

    return pd.Series([
        property_type.strip() if property_type else None,
        float(area) if area else None,
        furnishing
    ])
```

- Divided into three categories: type, size (in sqft), and furnishing status using the separator “ | ”.
- Converted size from string to float for quantitative analysis.
- Removed entries with sizes below 70 sqft to eliminate illogical data.
- Replaced NaN values in furnishing status with "Unknown."

5.0 Optimization Techniques

5.1 Methods used

Person In Charge	Library
Lee Soon Der	**Polars**: A high performance DataFrame library designed for faster data processing, especially for large volumes of dataset. It is built with rust as its core, offers multi-threaded executions and supports lazy evaluation, making it more memory-efficient than pandas.
Lim Jing Yong	**Modin & Ray**: Modin is a high-performance drop-in replacement for pandas that speeds up data processing by automatically distributing operations across all CPU cores. Ray, which is a general purpose distributed execution engine, is used by Modin to perform the parallel operation efficiently.
Jaslene Yu	**Multiprocessing**: A library in Python allows a program to utilize multiple CPU cores simultaneously, enabling parallel execution of tasks. This is particularly useful for CPU-bound data processing operations where processing can be split across processes to reduce runtime.
	Spark: Apache Spark is a distributed computing framework designed for large-scale data processing across clusters. It optimizes performance using in-memory computation, lazy evaluation, and task parallelism. Spark's DataFrame and SQL APIs allow developers to express complex data workflows that Spark automatically optimizes under the hood using the Catalyst optimizer and Tungsten engine.
Nik Zulaikhaa	**Joblib**: Joblib is a parallel processing library in Python that simplifies running tasks concurrently using multiple CPU cores. It is ideal for speeding up repetitive, CPU-bound operations like data parsing by distributing workloads efficiently with minimal code changes.

5.2 Code Overview

Person In Charge	Code

Lee Soon Der	Use CLI command to download Polars
	`[ ] pip install polars`
	
	Load csv in lazy mode
	`df_lazy = pl.read_csv("new_iproperty_dataset.csv").lazy()`
	
	Execute lazy plan to collect all previous cleaning processes and execute everything in parallel
	`df = df_lazy.collect()`

Lim Jing Yong	Use CLI command to download Modin with Ray
	`!pip install -U modin[ray] ray`
	
	Implement Modin into Pandas
	`import pandas as pd`
	to
	`import modin.pandas as pd`
	
	Import Ray library, initiate the engine and set it to the OS's environment for Modin to use it
	`import ray`
	
	`ray.init(ignore_reinit_error=True)`
	`os.environ["MODIN_ENGINE"] = "ray"`
	
	Confirming which engine Modin is using (another option is Dask)
	`import modin.config as modin_cfg`
	`print(modin_cfg.Engine.get())`
	
	`Ray`

Jaslene Yu	**Multiprocessing** Import Multiprocessing library `import multiprocessing as mp`

	mp.Pool.map() divides the tasks into smaller chunks in order to processes them in parallel using all available CPU cores mp.cpu_count()
	`def parallel_parse(series, func, processes=None):`
	`    with mp.Pool(processes or mp.cpu_count()) as pool:`
	`        results = pool.map(func, series)`
	`    return pd.DataFrame(results, columns=['Property Type',`
	`                                          'Property Size (sqft)',`
	`                                          'Property Furnishing Status'])`
	To avoid the infinite loop of processes by ensuring that the multiprocessing pool is only created when the script is executed directly
	`if __name__ == "__main__":`

	Spark
	
	Import SparkSession class from PySpark's sql module
	`from pyspark.sql import SparkSession`
	Using Spark SQL API for further operation
	`spark = SparkSession.builder \`
	`    .appName("iProperty Data Cleaning") \`
	`    .getOrCreate()`
	UDF is applied to the DataFrame in parallel across the partitions of the data and PySpark will distribute the UDF tasks across worker nodes
	`df = df.withColumn("parsed", parse_udf(col("Property Details")))`

Nik Zulaikhaa	Use CLI command to download JobLib
	`!pip install joblib`
	
	Imported Parallel and delayed from JobLib
	`from joblib import Parallel, delayed`
	
	Joblib is used to parallelize parsing tasks across CPU cores
	`results = Parallel(n_jobs=-1)(delayed(parse_property_details)(d) for d in df['Property Details'])`

6.0 Performance Evaluation

6.1 Before optimization

Below shows several screenshots of the basic data cleaning process's performance without optimization techniques. Session is restarted after every run to ensure memory usage is started from the initial state.

Run 1:

```
📊 Performance Summary
➡️ Total Records Processed: 144214
⏱ Total Processing Time: 113.38 seconds
⚙️ Memory Used (Before → After): 163.79 MB → 1241.14 MB
🔺 Peak Memory Usage: 456.80 MB
📈 Throughput: 1271.91 records/second
```

Run 2:

```
📊 Performance Summary
➡️ Total Records Processed: 144214
⏱ Total Processing Time: 107.69 seconds
⚙️ Memory Used (Before → After): 163.89 MB → 1242.30 MB
🔺 Peak Memory Usage: 461.97 MB
📈 Throughput: 1339.11 records/second
```

Run 3:

```
📊 Performance Summary
➡️ Total Records Processed: 144214
⏱ Total Processing Time: 107.36 seconds
⚙️ Memory Used (Before → After): 164.23 MB → 1243.98 MB
🔺 Peak Memory Usage: 461.96 MB
📈 Throughput: 1343.25 records/second
```

From the three runs:

Average Processing Time	109.48 seconds
Average Memory Used	1242.47 MB
Average Peak Memory Usage	460.24 MB
Average Throughput	1318.09 records/second

Table 6.1 Average Performance of 3 runs of unoptimized data cleaning

6.2 Comparison to Optimized Solution

Person In Charge	Performance
Lee Soon Der	Elapsed Time: 5.15 sec Memory Used (Start → End): 575.35 MB → 746.41 MB Peak Memory (tracemalloc): 99.14 MB Throughput: 27,990.23 records/sec Total Records Cleaned: 144214 Processing Time Difference : 104.33 seconds Memory Usage Difference : 496.06 MB Peak Memory Usage Difference : 361.1 MB Throughput Difference : 26672.14 records/second
Lim Jing Yong	**Modin & Ray** Performance Summary Total Records Processed: 144214 Total Processing Time: 26.41 seconds Memory Used (Before → After): 223.07 MB → 393.36 MB Peak Memory Usage: 71.74 MB Throughput: 5460.38 records/second Processing Time Difference : 83.07 seconds Memory Usage Difference : 849.11 MB Peak Memory Usage Difference : 388.5 MB Throughput Difference : 4142.29 records/second
Jaslene Yu	**Multiprocessing** Performance Summary Total Records Processed: 144214 Total Processing Time: 19.57 seconds Memory Used (Before → After): 164.45 MB → 456.23 MB Peak Memory Usage: 107.99 MB Throughput: 7368.01 records/second Processing Time Difference: 89.91 seconds Memory Usage Difference: 786.24 MB Peak Memory Usage Difference: 352.25 MB Throughput Difference: 6049.92 records/second

	Spark: 📊 Performance Summary ➡️ Total Records Processed: 144211 ⏱ Total Processing Time: 36.92 seconds ⚙️ Memory Used (Before → After): 106.32 MB → 106.51 MB 🔺 Peak Memory Usage: 0.15 MB 📈 Throughput: 3906.31 records/second Processing Time Difference: 72.56 seconds Memory Usage Difference: 1135.96 MB Peak Memory Usage Difference: 460.09 MB Throughput Difference: 2,588.22 records/second
Nik Zulaikhaa	**Joblib:** 📊 Performance Summary 📦 Total Records Processed: 144214 ⏱️ Total Processing Time: 20.19 seconds 🛠️ Memory Used (Before ➜ After): 182.28 MB ➜ 567.96 MB 🔺 Peak Memory Usage: 385.68 MB 📈 Throughput: 7141.52 records/second Processing Time Difference: 89.29 seconds Memory Usage Difference: 674.51 MB Peak Memory Usage Difference: 74.56 MB Throughput Difference: 5823.43 records/second

6.3 Graph Visualization

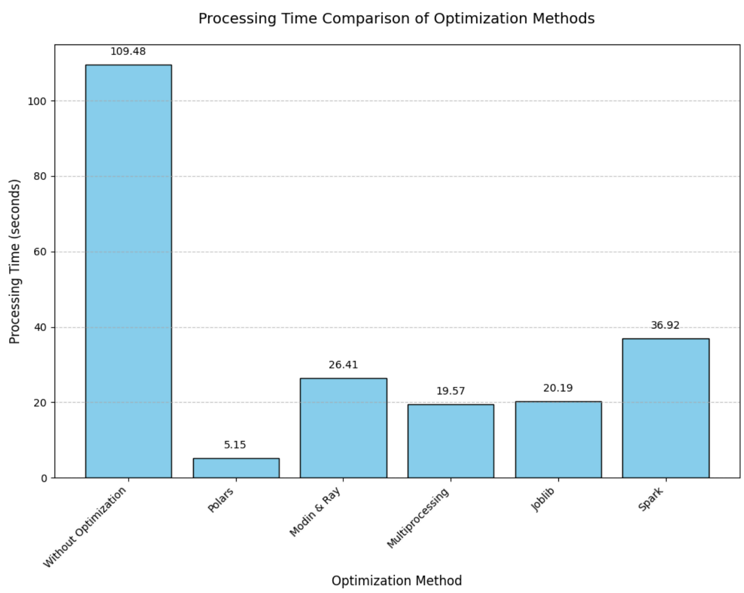


Figure 6.3.1 Graph of Comparison between Libraries (Processing Time)

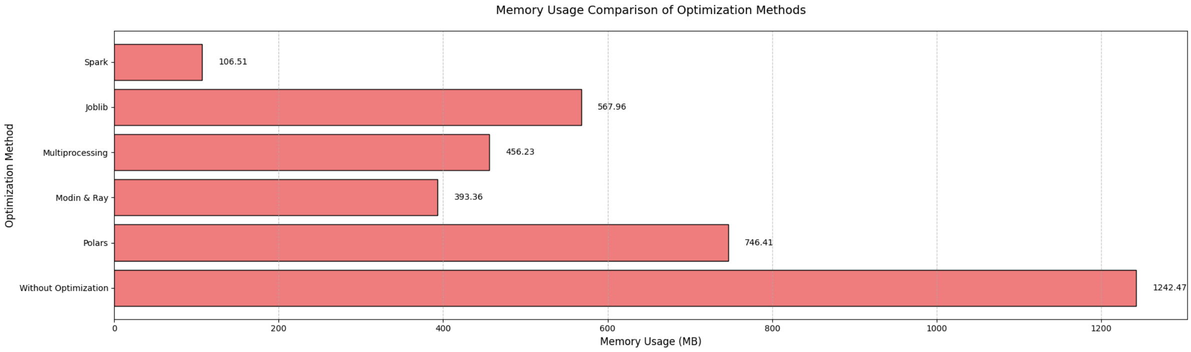


Figure 6.3.2 Graph of Comparison between Libraries (Memory Usage)

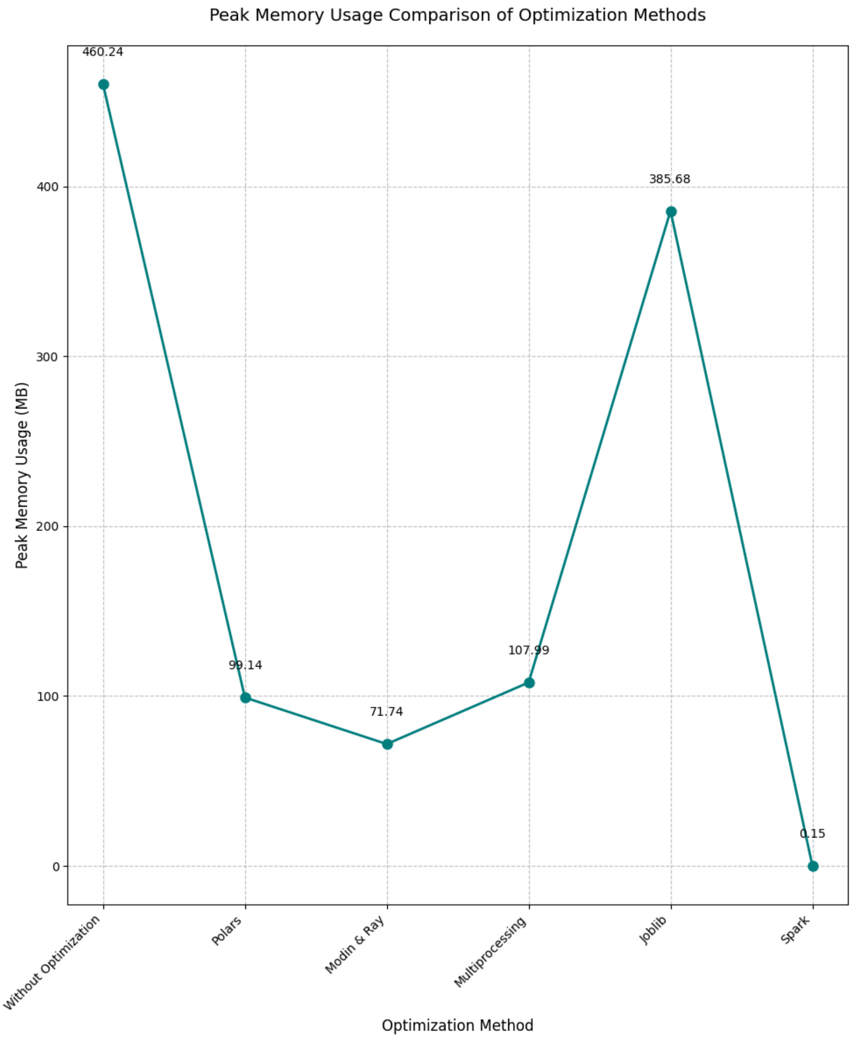


Figure 6.3.3 Graph of Comparison between Libraries (Peak Memory Usage)

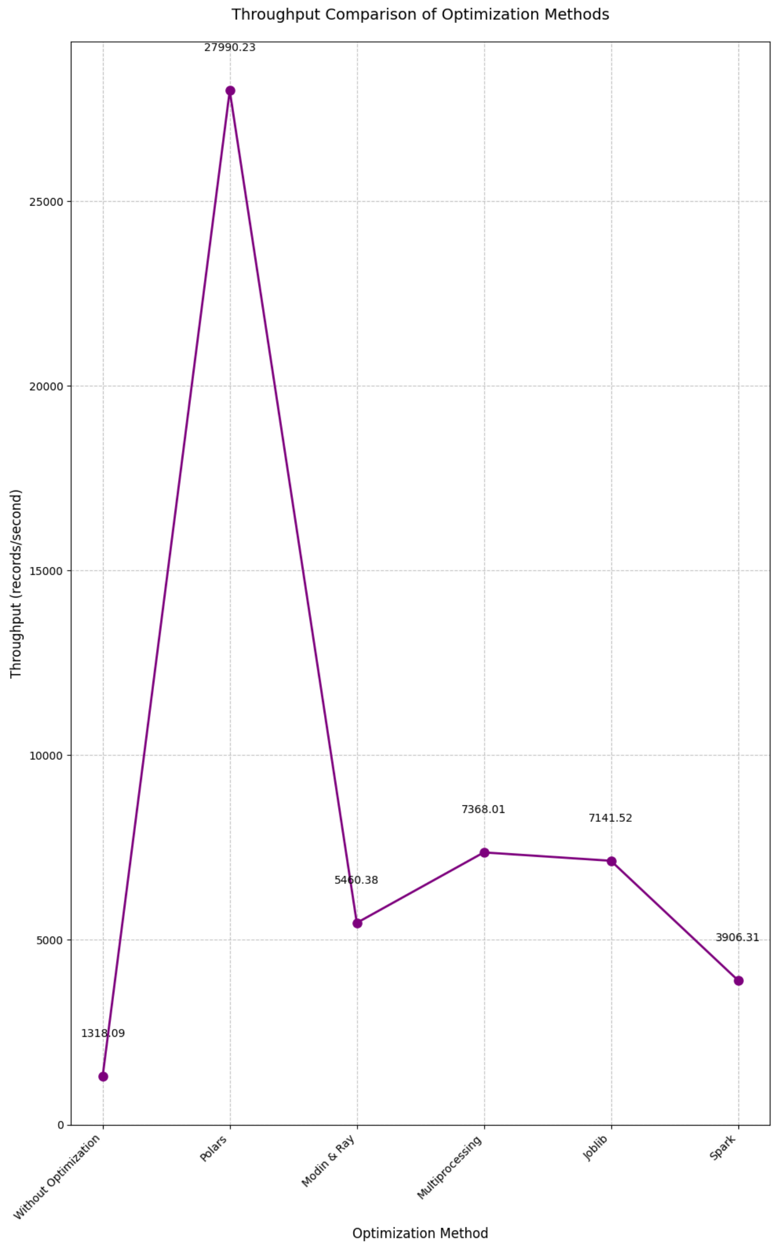


Figure 6.3.3 Graph of Comparison between Libraries (Throughput)

7.0 Challenges & Limitations

7.1 Challenges

- Iproperty website provides **multiple wrapper styles** (see section 1.3) which was unexpected initially. Due to BeautifulSoup's method of extracting data, additional modification must be done to our data scraping coding to scrap all wanted data.

- Iproperty website only has a maximum of 100 pages per filter, hence **area codes** must be listed individually in order to iterate between areas to scrap more than 100k data.

- During the data scraping process, Iproperty website's **CAPTCHA system** prevented us from extracting data. This issue was solved through usage of a user agent and an inconsistent (randomly timed) WebDriverWait() function.

- The property data in Iproperty website has **multiple blank columns and incorrect measurements** (exp. Property Size 1 sqft, NULL data for Property Furnishing Status, weird symbols in Property Name). This issue is addressed through criteria-based row dropping.

- Implementing Apache Spark for optimization required translating the existing data cleaning logic into Spark's **specific terminology and syntax**. For instance, operations like filtering and column manipulation, while conceptually similar to Pandas, necessitate the use of Spark's DataFrame API and functions (e.g., filter(), withColumn()), leading to a significant rewrite of the cleaning code.

7.2 Limitations

- An attempt was made to use Modin + Dask library for advanced optimization, however due to usage of **apply() function** in our data cleaning method, coding becomes too complicated and requires big modifications.

- Some high-performance libraries are **designed for processing massive volumes** of datasets with millions of rows such as Dask, Vaex and Apache Flink. When applied to smaller datasets like ours (~160k rows), the optimization performance will be lower than the lightweights tools like Pandas due to unnecessary overhead which results in slower execution and reduced efficiency.

- Although Joblib enabled parallel cleaning of the ”Property Details” column, the task was lightweight, and multiprocessing overhead outweighed the benefits for small datasets (~240 rows), resulting in minimal performance gains.

- A key inconsistency with the Apache Spark implementation was a lower final row count compared to other methods, even with identical cleaning logic, likely due to differences in regex handling.

8.0 Conclusion

8.1 Our Findings

Throughout this project, we tested multiple Python libraries to perform high-performance data processing and compared their effectiveness in terms of processing time, memory usage, peak memory, and throughput. Based on our evaluations, we can rank the libraries from best to least efficient as follows:

1. Polars

- Best performance overall in terms of throughput (26,672.14 records/sec).
- Very fast due to Rust core, multi-threading, and lazy evaluation.
- Ideal for large-scale structured datasets.

2. Modin + Ray

- Great improvement over baseline, no need to change much code from pandas.
- Good CPU utilization with automatic parallelization.
- Slightly less efficient than Polars due to overhead from Ray.

3. Multiprocessing

- Effective CPU-bound task handling with custom control.
- Very flexible but requires careful management to avoid bugs.
- High throughput and memory savings.

4. Joblib

- Easy implementation for parallelizing loops and repetitive tasks.
- Performed well for parsing tasks, moderate improvements across metrics.

5. Apache Spark

- Strong for distributed processing, but showed relatively lower throughput (2,588.22 records/sec) in our use case.
- High memory overhead due to Spark's architecture.
- More suitable for truly massive datasets (millions of rows or distributed clusters).

In terms of the full process (scraping → cleaning → optimization), we observed:

- Web scraping was stable after implementing Selenium with anti-bot tactics.
- Cleaning pipeline was highly customizable; however, apply() usage limited compatibility with some optimization frameworks.
- Optimization techniques drastically reduced processing time from ~109s to as low as ~5s (Polars), proving the value of choosing the right tool for scale.

8.2 Improvement to be done

For future enhancements, the following improvements are recommended:

- Refactor cleaning code to avoid apply(): The current usage of apply() limits compatibility with high-performance libraries (e.g., Dask, Vaex). Refactoring using vectorized operations or built-in functions would enhance performance and reduce processing time.
- Mixing libraries for faster web scraping: Currently, Selenium and BeautifulSoup were used sequentially, which can be slow for large-scale scraping. Combining libraries like Scrapy (for high-speed crawling), Playwright (for better dynamic rendering), or aiohttp with asyncio (for concurrent HTTP requests) can significantly improve scraping speed and efficiency.
- Use Dask with proper refactoring: Though Dask was skipped due to apply() limitations, refactoring the code would allow its integration, offering strong parallelism and scalability for very large datasets.
- Benchmark with larger datasets: Testing on a dataset with millions of rows would provide more realistic insights into how libraries like Spark and Dask perform under true high-load conditions.
- Explore hybrid optimization pipelines: Combining strengths of multiple libraries—e.g., using Polars for initial fast processing and Spark for distributed querying—could balance speed and scalability.
- Automate and scale scraping pipelines: Tools like Airflow or Kafka could help automate and manage continuous scraping tasks for real-time or large-scale applications.

Implementing these improvements would enable the system to handle more data at higher speeds, with better compatibility and efficiency across different processing stages.

Appendix

Iproperty Website: https://www.iproperty.com.my/sale/all-residential/

Data Scraping Google Colab:
https://colab.research.google.com/drive/13rqFmOfF7uuyUb1AsjxuVXrj2sjHaPgW?usp=sharing

SQLite + Data Cleaning Google Colab:
https://colab.research.google.com/drive/1_jnPhfKX0OOMcsSbrDYIxRPJyHmyBOxP?usp=sharing

LogBook Github: https://github.com/users/Jingyong14/projects/3

Lim Jing Yong Optimization (Modin + Ray) Google Colab:
https://colab.research.google.com/drive/1WthqdbUkXWVnRG9hnmNKKdJNTeTsuCX0?usp=sharing

Lee Soon Der Optimization (Polars) Google Colab:
https://colab.research.google.com/drive/1j-R06GOCs3sAsXz6-0qcLT0PFf6Ee_jL?usp=sharing

Jaslene Yu Optimization (Multiprocessing + Spark) Google Colab:
https://colab.research.google.com/drive/1a91UY1zZKkCBuA-a6MSroimJ8VQv5TTN?usp=sharing

Nik Zulaikhaa Optimization (Joblib) Google Colab:
https://colab.research.google.com/drive/1SvnCFwGiOnF4mSEyzNy58V2jCziaZILW?usp=sharing

Visualization Google Colab:
https://colab.research.google.com/drive/1ETyMtF0aS4UgERg_PK4iGFIvApjYB_8o?usp=sharing