HUTECH
Đại học Công nghệ Tp.HCM

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

ĐỒ ÁN MÔN HỌC

THU THẬP VÀ PHÂN TÍCH DỮ LIỆU NHÀ ĐẤT TỪ TRANG WEB BATDONGSAN.VN SỬ DỤNG SELENIUM VÀ MONGODB

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **MÃ NGUỒN MỞ KHOA HỌC DỮ LIỆU**

Giảng viên hướng dẫn : ThS.Lê Nhật Tùng

Sinh viên thực hiện :

2386400039 - Phạm Trường Phát

2386400022 - Đinh Quốc Khánh

2386400982 - Lê Đình Nhật Thắng

TP. Hồ Chí Minh, 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

ĐỒ ÁN MÔN HỌC

THU THẬP VÀ PHÂN TÍCH DỮ LIỆU NHÀ ĐẤT TỪ TRANG BATDONGSAN.VN SỬ DỤNG SELENIUM VÀ MONGODB

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **MÃ NGUỒN MỞ KHOA HỌC DỮ LIỆU**

Giảng viên hướng dẫn : ThS.Lê Nhật Tùng

Sinh viên thực hiện :

2386400039 - Phạm Trường Phát

2386400022 - Đinh Quốc Khánh

2386400982 - Lê Đình Nhật Thắng

TP. Hồ Chí Minh, 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TPHCM, Ngày….. Tháng …... Năm 2025

Giáo viên hướng dẫn

(Ký tên, đóng dấu)

LỜI CAM ĐOAN

Nhóm chúng tôi gồm Phạm Trường Phát, Đinh Quốc Khánh, và Lê Đình Nhật Thắng xin cam đoan rằng:

Tất cả thông tin và kết quả nghiên cứu trong bài báo cáo này đều trung thực và khách quan, được thu thập từ các nguồn đáng tin cậy, chính thống. Chúng tôi đã phân tích kỹ lưỡng các tài liệu và đảm bảo rằng mọi dữ liệu hoặc ý kiến trích dẫn đều được ghi rõ ràng nguồn gốc, tuân thủ đúng quy định về trích dẫn học thuật.

Chúng tôi cam kết không có hành vi sao chép hoặc sử dụng trái phép bất kỳ thông tin nào từ các nguồn khác. Bài báo cáo này là sản phẩm nghiên cứu độc lập của nhóm, chưa từng được công bố trước đây và tuân thủ đầy đủ các quy định của môn học. Chúng tôi sử dụng công cụ nghiên cứu một cách hợp lý và chính xác, đảm bảo tính khoa học và đạo đức trong quá trình thực hiện.

Nhóm chúng tôi hy vọng rằng bài báo cáo sẽ cung cấp cái nhìn sâu sắc về chủ đề "Thu thập, lưu trữ và phân tích dữ liệu bất động sản từ website batdongsan.com.vn" và đóng góp tích cực vào phát triển của lĩnh vực này.

TPHCM, Ngày…… Tháng …… Năm 2025

Sinh viên

Phạm Trường Phát

Đinh Quốc Khánh

Lê Đình Nhật Thắng

DANH MỤC CÁC KÝ HIỆU, TỪ VIẾT TẮT VÀ TỪ KHOÁ

UC	Undetected Chromedriver để điều khiển trình duyệt Chrome tối ưu để tránh bị các website phát hiện là bot
HTML	HyperText Markup Language
XML	eXtensible Markup Language
BeautifulSoup	Thư viện Python dùng để phân tích HTML/XML
Selenium	Hệ thống tự động hoá trình duyệt
MongoDB	Hệ quản trị cơ sở dữ liệu NoSQL
CSS	Cascading Style Sheets, ngôn ngữ định dạng giao diện
lxml	Thư viện Python hỗ trợ phân tích cú pháp XML và HTML nhanh
MongoDB Atlas	Dịch vụ cơ sở dữ liệu đám mây
HTTP	HyperText Transfer Protocol, giao thức truyền tải siêu văn bản
DOM	Document Object Mode là một cấu trúc dữ liệu dạng cây phân cấp, biểu diễn tất cả các phần tử
JSON	JavaScript Object Notation, định dạng trao đổi dữ liệu nhẹ
JS	JavaScript, Ngôn ngữ lập trình chạy trên trình duyệt, xử lý các tương tác động
unidecode	Thư viện Python dùng để chuyển đổi các ký tự Unicode phức tạp
AWS	Amazon Web Services, nền tảng điện toán đám mây

MỤC LỤC

DANH SÁCH BẢNG **8**
DANH SÁCH HÌNH ẢNH, BIỂU ĐỒ **9**
CHƯƠNG 1: TỔNG QUAN **11**
1.1 Giới thiệu đề tài 11
1.2 Nhiệm vụ của đề tài 11
1.2.1 Tính cấp thiết của đề tài 11
1.2.2 Ý nghĩa khoa học và thực tiễn của đề tài 12
1.3 Mục tiêu 13
1.3.1 Mục tiêu tổng quát 13
1.3.2 Mục tiêu cụ thể 13
1.4 Đối tượng và phạm vi 13
1.4.1 Đối tượng 13
1.4.2 Phạm vi 13
1.5 Phương pháp nghiên cứu 13
1.5.1 Phương pháp nghiên cứu tài liệu 13
1.5.2 Phương pháp phân tích trang 14
1.5.3 Phương pháp thực nghiệm 14
1.5.4 Phương pháp kết hợp công cụ 14
1.5.5 Phương pháp đánh giá và phân tích dữ liệu 14
1.6 Những đóng góp nghiên cứu của đề tài 14
1.6.1 Trong lĩnh vực học thuật 14
1.6.2 Trong thực tiễn 14
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT **15**
2.1 Trích xuất dữ liệu web 15
2.1.1 Định nghĩa 15
2.1.2 Quy trình hoạt động 15
2.2 Selenium 16
2.2.1 Giới thiệu 16
2.2.2 Bộ công cụ selenium 16
2.3 Beautiful Soup 16
2.3.1 Giới thiệu 16

2.3.2 Kiến trúc và bộ phân tích cú pháp 16
2.3.3 Các đối tượng chính 17
2.3.4 Xử lý mã hóa 17
2.4 Cơ sở dữ liệu NoSQL 17
2.4.1 Giới thiệu 17
2.4.2 Ưu điểm của cơ sở dữ liệu NoSQL 18
2.4.3 Các trường hợp sử dụng cơ sở dữ liệu NoSQL 18
2.5 MongoDB 19
2.5.1 Giới thiệu 19
2.5.2 Các tính năng của MongoDB 19
2.6 MongoDB Atlas 20
2.6.1 Giới thiệu về MongoDB Atlas 20
2.6.2 Kiến trúc và mô hình triển khai 20
CHƯƠNG 3: PHƯƠNG PHÁP THỰC NGHIỆM 21
3.1 Phương pháp thu thập dữ liệu 21
3.1.1 Khởi tạo kết nối hệ thống 21
3.1.2 Điều phối thu thập đa mục tiêu 22
3.1.3 Trích xuất danh sách 24
3.1.4 Thu thập thông tin chi tiết 24
3.1.5 Lưu trữ dữ liệu MongoDB Atlas 25
3.2 Mô tả dữ liệu 26
CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM 30
4.1 Giới thiệu 30
4.2 Kết quả thu thập dữ liệu 30
4.3 Truy vấn dữ liệu 31
4.4 Phân tích dữ liệu bất động sản 35
CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ 47
5.1. Kết luận 47
5.2. Hạn chế của đề tài 47
5.3. Kiến nghị và Hướng phát triển 47
TÀI LIỆU THAM KHẢO 49

DANH SÁCH BẢNG

Bảng 3.1: Bảng mô tả các biến và kiểu dữ liệu.....27
Bảng 3.2: Thông tin mô tả bảng dự án (project_info).....27
Bảng 3.3: Thông tin mô tả bảng đặc diểm (spec) 28
Bảng 3.4: Thông tin mô tả bảng người môi giới (contact_info).....29

DANH SÁCH HÌNH ẢNH, BIỂU ĐỒ

Hình 3.1: Đoạn mã dùng để khởi tạo trình duyệt Chrome 21
Hình 3.2: Đoạn mã dùng để kết nối đến MongoDB Atlas 22
Hình 3.3: Cấu hình trong file config.yaml 23
Hình 3.4: Đoạn mã dùng để duyệt qua từng trang của từng loại giao dịch 23
Hình 3.5: Đoạn mã dùng để lấy danh sách các đường dẫn trong trang 24
Hình 3.6: Đoạn mã dùng để lấy các thông tin tin bất động sản 25
Hình 3.7: Cấu trúc dữ liệu được lưu trữ trên MongoDB 26
Hình 4.1: Tổng quan dữ liệu được lưu trữ trên MongDB Atlas 30
Hình 4.2: Tìm tất cả các bất động sản thuộc loại Villa/Townhouse 31
Hình 4.3: Tìm các bài đăng tin bán nhà ở Hải Phòng 31
Hình 4.4: Tìm nhà đất có hơn 3 phòng vệ sinh 32
Hình 4.5: Tìm bài đăng nhà có ban công hướng Bắc 32
Hình 4.6: Dự án có nhiều tin đăng nhất 33
Hình 4.7: Những bài đăng có nhiều hơn 24 hình ảnh 33
Hình 4.8: Tìm các tin đăng sau ngày 01/12/2025 33
Hình 4.9: Top 5 dự án có nhiều tin đăng 34
Hình 4.10: Đếm tổng số tin theo loại tin 34
Hình 4.11: Tìm những tin thuộc Apartment và có hướng Đông-Bắc 34
Hình 4.12: Thống kê mật độ tin theo khu vực 35
Hình 4.13: Đoạn mã dùng để kết nối và lấy dữ liệu về DataFrame 35
Hình 4.14: Mô tả tổng quan về dữ liệu trong DataFrame 36
Hình 4.15: Chia nhóm dữ liệu bất động sản thành hai phân khúc chính 37
Hình 4.16: Đoạn mã dùng để tách địa chỉ 37
Hình 4.17: Đoạn mã dùng để chuyển đổi cột price từ kiểu string sang float (sale) 38
Hình 4.18: Đoạn mã dùng để chuyển đổi cột price từ kiểu string sang float (rent) 38
Hình 4.19: Đoạn mã dùng để lấy top 5 loại hình bất động sản phổ biến nhất 39
Hình 4.20: Biểu đồ cơ cấu phân khúc bất động sản 40
Hình 4.21: Phân phối giá trên m² ở các quận/huyện tại Tp.Hồ Chí Minh 41
Hình 4.22: Phân phối giá ở các quận/huyện tại Hà Nội 41

Hình 4.23: Phân bố hướng nhà và mặt bằng giá trung vị 42
Hình 4.24: Phân phối giá thuê dưới 50 triệu VND 43
Hình 4.25: Phân phối giá thuê bán dưới 20 tỷ VND 43
Hình 4.26: Top 10 dự án nổi bật với số lượng tin và giá trung bình 44
Hình 4.27: Top 10 đặc trưng có tương quan cao nhất với tổng giá bất động sản 44
Hình 4.28: Bản đồ nhiệt phân bố tin đăng bất động sản tại Việt Nam 46

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu đề tài

Trong bối cảnh chuyển đổi số, dữ liệu đóng vai trò quan trọng trong nhiều lĩnh vực, đặc biệt bất động sản là nơi các tin đăng mua bán, cho thuê được cập nhật liên tục trên các nền tảng trực tuyến. Các website bất động sản không chỉ kết nối người mua và người bán mà còn phản ánh xu hướng thị trường, mức giá và đặc điểm bất động sản theo khu vực.

Trong đó, batdongsan.com.vn là một nền tảng lớn và uy tín tại Việt Nam, cung cấp khối lượng dữ liệu phong phú nhưng việc khai thác tự động gặp nhiều khó khăn do cấu trúc phức tạp và nội dung tải động. Xuất phát từ thực tế này, đề tài được thực hiện nhằm xây dựng một hệ thống thu thập và lưu trữ dữ liệu bất động sản tự động, phục vụ cho các hoạt động phân tích sau này.

1.2 Nhiệm vụ của đề tài

Nhiệm vụ của đề tài "Thu thập và lưu trữ dữ liệu bất động sản từ website batdongsan.com.vn sử dụng Selenium và MongoDB" là nghiên cứu cách thức hoạt động của các công cụ thu thập dữ liệu web, đặc biệt là Selenium, từ đó xây dựng chương trình thu thập dữ liệu bất động sản từ website batdongsan.com.vn một cách tự động và hiệu quả.

Bên cạnh đó, đề tài tiến hành phân tích cấu trúc HTML của website, trích xuất các thông tin quan trọng của tin đăng bất động sản như tiêu đề, giá bán, diện tích, địa chỉ, đặc điểm kỹ thuật, thông tin dự án và người đăng tin. Dữ liệu sau khi thu thập sẽ được làm sạch, chuẩn hóa và lưu trữ trong cơ sở dữ liệu MongoDB để phục vụ cho việc phân tích và khai thác dữ liệu.

1.2.1 Tính cấp thiết của đề tài

Hiện nay, thị trường bất động sản có sự biến động lớn về giá cả, loại hình và phân bố theo khu vực. Việc nắm bắt thông tin thị trường kịp thời và chính xác là nhu cầu thiết yếu đối với các nhà đầu tư, doanh nghiệp bất động sản cũng như các nhà

nghiên cứu. Tuy nhiên, dữ liệu bất động sản thường phân tán, cập nhật liên tục và khó tổng hợp nếu thu thập thủ công.

Website batdongsan.com.vn chứa một lượng lớn dữ liệu giá trị nhưng không cung cấp sẵn các công cụ trích xuất dữ liệu cho mục đích nghiên cứu. Việc thu thập dữ liệu tự động gặp nhiều khó khăn do website sử dụng nội dung tải động và các biện pháp hạn chế truy cập tự động. Vì vậy, cần có giải pháp kỹ thuật phù hợp để thu thập dữ liệu hiệu quả.

Selenium là công cụ mạnh mẽ cho phép mô phỏng hành vi người dùng trên trình duyệt, có khả năng xử lý các trang web động. Khi kết hợp Selenium với các thư viện phân tích HTML và cơ sở dữ liệu MongoDB, có thể xây dựng một hệ thống thu thập và lưu trữ dữ liệu bất động sản có tính linh hoạt, mở rộng và phù hợp với dữ liệu phi cấu trúc.

Do đó, việc nghiên cứu và ứng dụng Selenium trong thu thập dữ liệu bất động sản từ batdongsan.com.vn là cần thiết, mang tính thực tiễn cao và phù hợp với xu hướng ứng dụng khoa học dữ liệu hiện nay.

1.2.2 Ý nghĩa khoa học và thực tiễn của đề tài

Đề tài góp phần mở rộng kiến thức về lĩnh vực thu thập dữ liệu web, đặc biệt là đối với các website có nội dung tải động và cấu trúc phức tạp như các nền tảng bất động sản. Nghiên cứu giúp làm rõ cách thức kết hợp Selenium với các thư viện xử lý dữ liệu trong Python và cơ sở dữ liệu MongoDB để xây dựng một hệ thống thu thập dữ liệu hoàn chỉnh. Kết quả của đề tài có thể được sử dụng làm tài liệu tham khảo cho các nghiên cứu liên quan đến khoa học dữ liệu và khai phá dữ liệu web.

Ý nghĩa thực tiễn: Về mặt thực tiễn, đề tài cung cấp giải pháp tự động hóa việc thu thập dữ liệu bất động sản, giúp tiết kiệm thời gian và công sức so với phương pháp thu thập thủ công. Dữ liệu thu thập được có thể được sử dụng để phân tích giá bất động sản, đánh giá xu hướng thị trường, hỗ trợ ra quyết định đầu tư và nghiên cứu thị trường. Ngoài ra, hệ thống có thể mở rộng để thu thập dữ liệu theo thời gian thực hoặc tích hợp vào các bài toán phân tích và dự đoán trong tương lai.

1.3 Mục tiêu

1.3.1 Mục tiêu tổng quát

Đề tài "Thu thập và lưu trữ dữ liệu bất động sản từ website batdongsan.com.vn sử dụng Selenium và MongoDB" nhằm xây dựng một hệ thống thu thập dữ liệu bất động sản tự động, có khả năng xử lý nội dung động, chuẩn hóa dữ liệu và lưu trữ hiệu quả, từ đó tạo tiền đề cho các hoạt động phân tích và khai thác dữ liệu bất động sản.

1.3.2 Mục tiêu cụ thể

Phân tích cấu trúc website batdongsan.com.vn và xác định các thành phần dữ liệu cần thu thập. Xây dựng chương trình thu thập dữ liệu bất động sản sử dụng Selenium. Trích xuất các thông tin quan trọng của tin đăng bất động sản như giá, diện tích, địa chỉ, đặc điểm kỹ thuật và thông tin người đăng. Làm sạch và chuẩn hóa dữ liệu thu thập được. Lưu trữ dữ liệu vào cơ sở dữ liệu MongoDB để phục vụ cho việc truy vấn và phân tích.

1.4 Đối tượng và phạm vi

1.4.1 Đối tượng

Đối tượng nghiên cứu của đề tài là các công cụ và kỹ thuật thu thập dữ liệu web, cụ thể là Selenium trong Python, cùng với cơ sở dữ liệu MongoDB và dữ liệu bất động sản được công bố trên website batdongsan.com.vn.

1.4.2 Phạm vi

Đề tài tập trung vào việc thu thập dữ liệu các tin đăng mua bán và cho thuê bất động sản trên website batdongsan.com.vn. Phạm vi nghiên cứu chỉ giới hạn trong việc thu thập, làm sạch và lưu trữ dữ liệu, trực quan, không đi sâu vào các mô hình phân tích hay dự đoán giá bất động sản.

1.5 Phương pháp nghiên cứu

1.5.1 Phương pháp nghiên cứu tài liệu

Thu thập và nghiên cứu các tài liệu liên quan đến web scraping, Selenium, Python và MongoDB nhằm xây dựng cơ sở lý thuyết cho đề tài.

1.5.2 Phương pháp phân tích trang

Phân tích cấu trúc HTML và các thành phần tải động của website batdongsan.com.vn để xác định các dữ liệu cần trích xuất.

1.5.3 Phương pháp thực nghiệm

Xây dựng và thử nghiệm chương trình thu thập dữ liệu bất động sản bằng Selenium, đánh giá khả năng thu thập dữ liệu trong điều kiện thực tế.

1.5.4 Phương pháp kết hợp công cụ

Kết hợp Selenium với các thư viện BeautifulSoup xử lý dữ liệu và MongoDB nhằm đảm bảo việc thu thập, lưu trữ và quản lý dữ liệu hiệu quả.

1.5.5 Phương pháp đánh giá và phân tích dữ liệu

Đánh giá chất lượng, tính đầy đủ và độ chính xác của dữ liệu thu thập được sau quá trình làm sạch và lưu trữ.

1.6 Những đóng góp nghiên cứu của đề tài

1.6.1 Trong lĩnh vực học thuật

Đề tài đóng góp vào việc nghiên cứu và ứng dụng các kỹ thuật thu thập dữ liệu web trong lĩnh vực khoa học dữ liệu. Kết quả nghiên cứu giúp làm rõ cách xây dựng hệ thống thu thập dữ liệu cho các website có nội dung động và cấu trúc phức tạp, đồng thời cung cấp tài liệu tham khảo cho các nghiên cứu liên quan. tài liệu tham khảo hữu ích cho các nghiên cứu sau này trong lĩnh vực này.

1.6.2 Trong thực tiễn

Về mặt thực tiễn, đề tài cung cấp một giải pháp khả thi cho việc thu thập và lưu trữ dữ liệu bất động sản tại Việt Nam. Hệ thống có thể được sử dụng để hỗ trợ phân tích thị trường, nghiên cứu xu hướng giá và phục vụ cho các ứng dụng thực tế trong lĩnh vực bất động sản.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Trích xuất dữ liệu web

2.1.1 Định nghĩa

Web scraping là quá trình trích xuất dữ liệu từ các trang web một cách tự động thông qua các chương trình hoặc kịch bản phần mềm [1]. Python được sử dụng rộng rãi cho web scraping nhờ cú pháp dễ hiểu và các thư viện mạnh mẽ như BeautifulSoup và Selenium [2], [3], giúp trích xuất dữ liệu hiệu quả hơn so với sao chép thủ công. Quá trình web scraping thường bao gồm việc gửi yêu cầu đến máy chủ, tải về mã nguồn HTML và bóc tách dữ liệu cần thiết để lưu trữ, dựa trên nguyên lý kiến trúc REST của hệ thống web [4]. Khi thu thập dữ liệu từ web, cần xem xét các yếu tố pháp lý và đạo đức, đặc biệt là chính sách robots.txt và điều khoản sử dụng của trang web mục tiêu [5].

2.1.2 Quy trình hoạt động

Quy trình web scraping cơ bản bao gồm các bước sau [7]:

- Xác định mục tiêu như là URL và cấu trúc dữ liệu cần thu thập.
- Sử dụng giao thức HTTP để lấy mã nguồn HTML/CSS/JS từ máy chủ
- Phân tích cú pháp HTML để hiểu cấu trúc của mã nguồn, tìm kiếm các thẻ HTML, lớp, ID…
- Trích xuất dữ liệu bằng cách sử dụng các bộ chọn như XPath hoặc CSS selector để xác định và lấy nội dung mong muốn.
- Làm sạch dữ liệu thô và lưu trữ vào định dạng có cấu trúc như bảng tính, JSON, CSV hoặc cơ sở dữ liệu.

2.2 Selenium

2.2.1 Giới thiệu

Selenium là một framework mã nguồn mở dùng để tự động hóa trình duyệt web, cho phép mô phỏng hành vi người dùng như nhấp chuột, nhập liệu và điều hướng trang web [6]. Hỗ trợ nhiều trình duyệt như Chrome, Firefox, Edge…, đồng thời tích hợp liền mạch với các ngôn ngữ lập trình như Python, Java, C# và JavaScript. Đặc biệt, Selenium rất mạnh trong việc xử lý các trang web có nội dung tải động bằng JavaScript [9].

2.2.2 Bộ công cụ selenium

Bộ công cụ Selenium bao gồm:

- Selenium WebDriver: Công cụ chính để điều khiển trình duyệt thông qua mã lập trình, thực hiện các thao tác nhấp chuột, nhập văn bản, điều hướng trang và mô phỏng các hành động của người dùng.
- Selenium IDE: Trình duyệt để ghi lại các thao tác và tự động tạo kịch bản kiểm thử.
- Selenium Grid: Chạy thử nghiệm trên nhiều trình duyệt và hệ thống cùng lúc để kiểm thử tương thích đa trình duyệt.

2.3 Beautiful Soup

2.3.1 Giới thiệu

Beautiful Soup là một thư viện Python dùng để trích xuất dữ liệu từ các tệp HTML và XML. Chức năng cốt lõi của nó là cung cấp các phương thức điều hướng, tiềm kiếm và sửa đổi cây phân tích cú pháp một cách tự nhiên, giúp tiết kiệm thời gian viết mã cho lập trình viên [8].

2.3.2 Kiến trúc và bộ phân tích cú pháp

Beautiful Soup không tự thực hiện việc phân tích cú pháp mà cung cấp giao diện để làm việc với các thư viện phân tích cú pháp khác, hỗ trợ các paser sau:

- html.parser: Bộ phân tích tiêu chuẩn đi kèm với Python. Ưu điểm là tính ổn định và không yêu cầu cài đặt phụ thuộc bên ngoài.
- lxml: Bộ phân tích cú pháp XML/HTML viết bằng C. Một số tài liệu đánh giá lxml có tốc độ xử lý rất nhanh, đây là parser được khuyến nghị sử dụng trong môi trường sản phẩm.
- html5lib: Phân tích tài liệu HTML giống hệt cách trình duyệt web hoạt động. Parser này cực kỳ mạnh mẽ trong việc xử lý HTML bị lỗi nặng nhưng có tốc độ chậm hơn lxml.

2.3.3 Các đối tượng chính

Trong quá trình phân tích tài liệu HTML, Beautiful Soup chuyển đổi mã nguồn thành các đối tượng Python. Tài liệu phân loại 4 đối tượng chính quan trọng trong cấu trúc dữ liệu của thư viện:

- Tag: Tương ứng với một thẻ HTML hoặc XML trong tài liệu gốc. Đối tượng Tag sở hữu các thuộc tính quan trong như name và attrs từ điển chứa các thuộc tính như class, id,hred.
- NavigableString: Biểu diễn văn bản chứa bên trong một thẻ. Đây là chuỗi Unicode và là dữ liệu thực tế mà quá trình Web Scaping thường hướng tới.
- BeautifulSoup: Đối tương đại diện cho toàn bộ tài liệu được phân tích.
- Comment: Đối tượng đặc biết để xử lý các phần chú thích trong HTML.

2.3.4 Xử lý mã hóa

Xử lý mã hóa là tự động phát hiện và chuyển đổi bảng mã. Beautiful Soup tự động chuyển đổi tài liệu đầu vào sang Unicode và tài liệu đầu ra sang UTF-8. Điều này giải quyết vấn đề hiển thị sai ký tự thường gặp khi thu thập dữ liệu tiếng việt từ các trang web cũ hoặc có cấu hình server không chuẩn [10].

2.4 Cơ sở dữ liệu NoSQL

2.4.1 Giới thiệu

Cơ sở dữ liệu NoSQL được thiết kế để lưu trữ dữ liệu theo các mô hình linh hoạt, hỗ trợ khả năng mở rộng và hiệu năng cao cho các ứng dụng hiện đại [10]. Ngoài

ra, cơ sở dữ liệu NoSQL được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn. NoSQL đặc biệt phù hợp với dữ liệu phi cấu trúc và bán cấu trúc, đồng thời được công nhận rộng rãi trong các hệ thống quy mô lớn [11].

2.4.2 Ưu điểm của cơ sở dữ liệu NoSQL

Ưu điểm:

- Cung cấp các sơ đồ linh hoạt giúp công đoạn phát triển nhanh hơn và có khả năng lặp lại cao hơn lựa chọn lý tưởng cho dữ liệu phi cấu trúc hoặc dữ liệu bán cấu trúc [12].
- Khả năng mở rộng bằng cách phân tán dữ liệu trên nhiều máy chủ và thường được cung cấp dưới dạng dịch vụ đám mây được quản lý.
- Được tối ưu cho từng mô hình dữ liệu cụ thể nên cho hiệu suất và truy vấn cao hơn so với cơ sở dữ liệu quan hệ trong nhiều trường hợp [13].
- Cung cấp các API và kiểu dữ liệu cực kỳ thiết thực được xây dựng riêng cho từng mô hình dữ liệu tương ứng.

2.4.3 Các trường hợp sử dụng cơ sở dữ liệu NoSQL

Quản lý dữ liệu theo thời gian thực: NoSQL hỗ trợ xử lý dữ liệu nhanh và theo thời gian thực, cung cấp các tính năng như đề xuất cá nhân hóa, danh sách xem và tiếp tục xem. Nhờ khả năng mở rộng cao, NoSQL được sử dụng để phục vụ hàng trăm triệu người dùng cùng lúc, điển hình như các nền tảng xem phim trực tuyến.

Bảo mật trên nền tảng đám mây: NoSQL dạng đồ thị cho phép phân tích nhanh các mối quan hệ phức tạp trong dữ liệu, từ đó phát hiện và xử lý rủi ro bảo mật. Loại cơ sở dữ liệu này đặc biệt phù hợp cho các hệ thống giám sát an ninh và quản lý rủi ro trên môi trường đám mây.

Các ứng dụng có độ sẵn sàng cao: NoSQL phân tán rất phù hợp để xây dựng các ứng dụng yêu cầu độ trễ thấp và tính sẵn sàng cao như mạng xã hội, nhắn tin, chia sẻ tệp. Nhờ khả năng mở rộng và xử lý song song, giúp hệ thống hoạt động ổn định hơn.

NoSQL được ứng dụng rộng rãi trong các hệ thống thời gian thực, nền tảng đám mây và các ứng dụng yêu cầu độ sẵn sàng cao như mạng xã hội và thương mại điện tử [14].

2.5 MongoDB

2.5.1 Giới thiệu

Mongo là một hệ quản trị cơ sở dữ liệu thuộc loại NoSQL, được thiết kế để lưu trữ và xử lý dữ liệu dung lượng lớn với hiệu suất cao.

Khác với cơ sở dữ liệu quan hệ sử dụng bảng và hàng, MongoDB lưu trữ dữ liệu dưới dạng các tài liệu theo cặp khóa và giá trị, được tổ chức trong các tập hợp. Nhờ mô hình dữ liệu linh hoạt, MongoDB cho phép lưu trữ dữ liệu phi cấu trúc và bán cấu trúc một cách hiệu quả. Mô hình này giúp tổ chức dữ liệu linh hoạt và dễ mở rộng theo chiều ngang [15].

Đồng thời hỗ trợ các chức năng như truy vấn nhanh, lập chỉ mục, tổng hợp dữ liệu và mở rộng hệ thống dễ dàng trên nhiều máy chủ. MongoDB cũng hỗ trợ nhiều ngôn ngữ lập trình như Python, java, C#, Go và Ruby.

2.5.2 Các tính năng của MongoDB

MongoDB cung cấp các tính năng nổi bật như:

- Sao chép dữ liệu: thông qua bộ sao chép nhằm đảm bảo tính sẵn sàng cao cho hệ thống. Trong đó, một máy chủ chính xử lý các thao tác đọc và ghi, các máy chủ lưu trữ bản sao dữ liệu và sẵn sàng thay thế khi máy chủ chính gặp sự cố. Đảm bảo tính sẵn sàng cao thông qua Replica Sets [16].
- Khả năng mở rộng theo 2 chiều: Mở rộng theo chiều dọc, tăng cấu hình cho một máy chủ. Mở rộng theo chiều ngang, bổ sung thêm nhiều máy chủ vào hệ thống để chia tải.
- Cân bằng tải: tự động cân bằng tải giữa các máy chủ thông qua cơ chế mở rộng và phân tán dữ liệu, phân mảnh dữ liệu để cân bằng tải và mở rộng hệ thống [17].
- Không cần lược đồ cố định: MongoDB là cơ sở dữ liệu không cần lược đồ cố định, cho phép thay đổi cấu trúc dữ liệu linh hoạt trong quá trình lưu trữ và xử lý.

- Mô hình lưu trữ dữ liệu: Được lưu trữ dưới dạng các tài liệu gồm các cặp khóa-giá trị, thay vì hàng và cột như trong SQL, giúp dữ liệu linh hoạt và dễ mở rộng.

2.6 MongoDB Atlas

2.6.1 Giới thiệu về MongoDB Atlas

MongoDB Atlas là dịch vụ cơ sở dữ liệu NoSQL dạng đám mây do MongoDB Inc. phát triển, cho phép triển khai, quản lý và vận hành cơ sở dữ liệu MongoDB mà không cần cài đặt và duy trì hạ tầng máy chủ cục bộ [18]. Dịch vụ này hỗ trợ triển khai trên các nền tảng điện toán đám mây phổ biến như Amazon Web Services (AWS), Google Cloud Platform (GCP) và Microsoft Azure.

MongoDB Atlas kế thừa đầy đủ các đặc điểm cốt lõi của MongoDB, bao gồm mô hình dữ liệu hướng tài liệu, khả năng lưu trữ dữ liệu phi cấu trúc và hỗ trợ truy vấn linh hoạt. Nhờ các đặc điểm này, MongoDB Atlas phù hợp với các hệ thống xử lý dữ liệu lớn và các ứng dụng yêu cầu khả năng mở rộng cao.

2.6.2 Kiến trúc và mô hình triển khai

MongoDB Atlas được triển khai dưới dạng các cụm cơ sở dữ liệu trên nền tảng đám mây. Mỗi cluster bao gồm nhiều node đảm nhiệm các vai trò khác nhau nhằm đảm bảo tính sẵn sàng, khả năng sao lưu và độ tin cậy của hệ thống. MongoDB Atlas hỗ trợ cơ chế sao lưu tự động, giám sát hiệu năng và bảo mật kết nối thông qua các giao thức mã hóa giúp tối ưu hóa khả năng truy cập và giảm thiểu chi phí quản lý vận hành.

Người dùng có thể kết nối đến MongoDB Atlas thông qua chuỗi kết nối được cung cấp sẵn, cho phép các ứng dụng bên ngoài truy cập cơ sở dữ liệu một cách thuận tiện. Mô hình triển khai này giúp tách biệt rõ ràng giữa tầng ứng dụng và tầng lưu trữ dữ liệu, từ đó nâng cao tính linh hoạt và khả năng mở rộng của hệ thống.

Dữ liệu bất động sản có đặc trưng là số lượng lớn, cấu trúc không đồng nhất và thường xuyên thay đổi, do đó việc sử dụng cơ sở dữ liệu NoSQL như MongoDB Atlas là phù hợp. Việc triển khai MongoDB Atlas trên nền tảng đám mây giúp hệ thống giảm thiểu công sức quản lý hạ tầng, đồng thời cho phép truy cập dữ liệu linh hoạt từ nhiều môi trường khác nhau.

CHƯƠNG 3: PHƯƠNG PHÁP THỰC NGHIỆM

3.1 Phương pháp thu thập dữ liệu

Công việc thu thập dữ liệu về tin thuê bán nhà đất từ web batdongsan.vn được tiến hành thực hiện bằng việc sử dụng công cụ Selenium để truy cập đến từng trang tin bất động sản và dùng beautifulsoup đề trích xuất các thông tin. Sau đó tiến hành lưu trữ thông tin đã được thu thập đến cơ sở dữ liệu MongoDB Atlas. Quá trình này được thực hiện theo các bước sau:

3.1.1 Khởi tạo kết nối hệ thống

Để hạn chế việc khả năng bị phát hiện bởi các cơ chế chống bot của website, thư viện undetected_chromedriver (UC) dùng để khởi tạo và điều khiển trình duyệt Chrome được sử dụng trong bài toán này.

```
def init_browser(headless=True):
    logger.info("Initializing driver...")
    options = uc.ChromeOptions()
    if headless:
        options.add_argument("--headless=new")
    options.add_argument("--start-maximized")
    options.add_argument("--disable-gpu")
    options.add_argument("--disable-dev-shm-usage")
    options.add_argument("--no-sandbox")
    options.add_argument("--disable-extensions")
    user_agent = (
        "Mozilla/5.0 (Windows NT 10.0; Win64; x64) "
        "AppleWebKit/537.36 (KHTML, like Gecko) "
        "Chrome/120.0.0.0 Safari/537.36"
    )
    options.add_argument(f'--user-agent={user_agent}')

    try:
        driver = uc.Chrome(options=options)
        driver.set_page_load_timeout(60)
        return driver

    except Exception as e:
        logger.error(f"Failed to initialize driver: {e}")
        raise
```

Hình 3.1: Đoạn mã dùng để khởi tạo trình duyệt Chrome

Sau đó tạo client kết nối đến tới MongoDB Atlas thông qua chuỗi kết nối (URI) được bảo mật trong tệp biến môi trường .env.

```python
class MongoDBClient:
    def __init__(self):
        user = os.getenv("MONGO_USER")
        pw = os.getenv("MONGO_PASS")
        cluster = os.getenv("MONGO_CLUSTER")
        db_name = os.getenv("MONGO_DB")
        col_name = os.getenv("MONGO_COLLECTION")
        uri = f"mongodb+srv://{user}:{pw}@{cluster}/?retryWrites=true&w=majority"

        try:
            self.client = MongoClient(uri, serverSelectionTimeoutMS=5000)
            self.client.server_info()
            self.db = self.client[db_name]
            self.col = self.db[col_name]
            self.col.create_index(
                keys: [("post_id", 1), ("transaction_type", 1)],
                unique=True
            )
            logger.info(f"Connected to Atlas: DB [{db_name}] | Collection [{col_name}]")

        except Exception as e:
            logger.error(f"MongoDB connection failed: {e}")
            raise
```

Hình 3.2: Đoạn mã dùng để kết nối đến MongoDB Atlas

3.1.2 Điều phối thu thập đa mục tiêu

Trên trang batdongsan.com.vn, có 2 mục tiêu chính được chọn ra để thu thập dữ liệu:

– Nhà đất bán: https://batdongsan.com.vn/nha-dat-ban

– Nhà đất cho thuê: https://batdongsan.com.vn/nha-dat-cho-thue

```yaml
scraper:
  headless: true

targets:
  - name: "nha_dat_ban"
    url: "https://batdongsan.com.vn/nha-dat-ban"
    start_page: 1
    end_page: 2
    enabled: true
  - name: "nha_dat_cho_thue"
    url: "https://batdongsan.com.vn/nha-dat-cho-thue"
    start_page: 1
    end_page: 2
    enabled: true
```

Hình 3.3: Cấu hình trong file config.yaml

Việc cấu hình targets có trong file config.yaml, giúp điều hướng luân phiên dễ dàng tới các URL loại giao dịch bất động sản. Mỗi một mục tiêu trong danh sách, chương trình tự động xác định phạm vi thu thập dựa trên tham số start_page và end_page. Sử dụng Selenium WebDriver chạy qua các trang tiếp theo của mỗi loại bằng cách nối hậu tố /p{số_trang}.

```python
def process_multiple_pages(driver, target, mongo_client):
    name = target.get("name")
    base_url = target.get("url")
    start_page = target.get("start_page", 1)
    end_page = target.get("end_page", 2)

    total_new = 0

    for p in range(start_page, end_page + 1):
        page_url = base_url if p == 1 else f"{base_url}/p{p}"

        links = fetch_list_links(driver, page_url,
                                 mongo_client)

        if not links:
            continue

        inserted_count = process_single_page(driver, links,
mongo_client)

        time.sleep(random.uniform(3, 6))

    return total_new
```

Hình 3.4: Đoạn mã dùng để duyệt qua từng trang của từng loại giao dịch

3.1.3 Trích xuất danh sách

Tại mỗi trang danh sách đã được xác định, chương trình thực hiện nhận diện các đường dẫn của tin bất động sản có trong trang đó. Sau đó thu thập các URL vào một danh sách, chương trình trích xuất id của từng tin thông qua URL, kiểm tra xem id này có nằm trong cơ sở dữ liệu MongoDB Alas nhằm đảm bảo khi chạy nhiều lần sẽ không bị trùng lặp tin.

```
def fetch_list_links(driver, page_url, mongo_client):
    links = []

    driver.get(page_url)
    wait = WebDriverWait(driver, 15)
    wait.until(EC.presence_of_element_located((By.CLASS_NAME, "js__card")))
    items = driver.find_elements(By.CLASS_NAME, "js__card")
    type_post = classify_transaction_type(page_url)

    for item in items:
        try:
           link_tag = item.find_element(By.TAG_NAME, "a")
           url = link_tag.get_attribute("href")
           pid = get_post_id(url)

           if pid and not mongo_client.check_duplicated(pid, type_post):
               links.append((url, pid))

        except (NoSuchElementException, StaleElementReferenceException):
             continue
    return links
```

Hình 3.5: Đoạn mã dùng để lấy danh sách các đường dẫn trong trang

3.1.4 Thu thập thông tin chi tiết

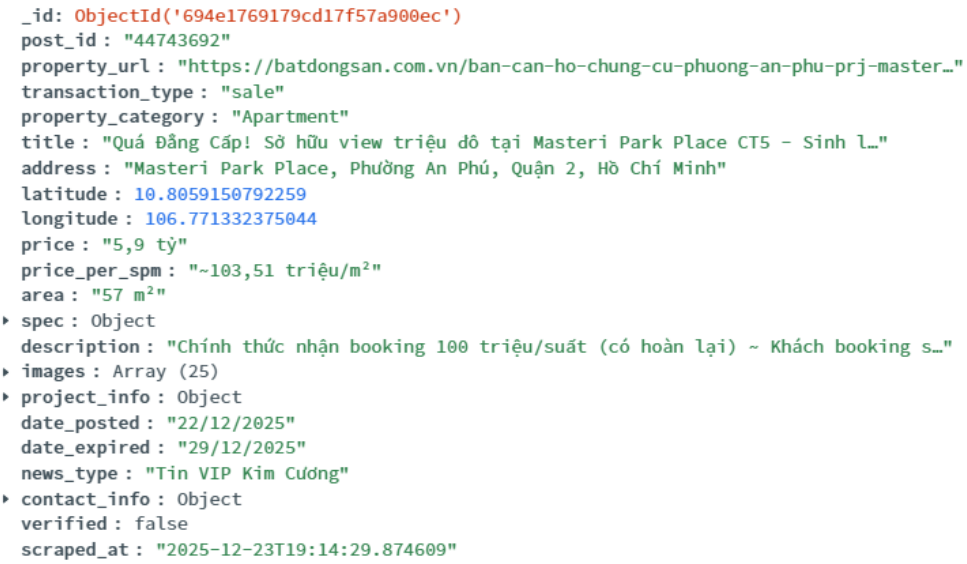
Sử dụng Selenium WebDriver để truy cập lần lượt các URL bài đăng và thu thập mã nguồn HTML của từng trang. Áp dụng các cơ chế chờ để đảm bảo các thành phần quan trọng của trang được hiển thị đầy đủ trước khi tiến hành trích xuất dữ liệu. Sau khi trang được tải hoàn tất, mã HTML được trích xuất và phân tích bằng thư viện BeautifulSoup nhằm thu thập các thông tin chi tiết của bất động , bao gồm: tiêu đề bài đăng, giá bán, diện tích, địa chỉ, mô tả chi tiết, đặc điểm bất động sản và các thông tin liên quan khác.

```
def parse_detail_page(html_content, url):
    post_id = get_post_id(url)
    soup = BeautifulSoup(html_content, "lxml")
    title = get_text(soup, "h1")
    address = get_text(soup, "span.re__pr-short-description")
    price_per_spm = get_text(soup, "span.ext")
    description = get_description(soup)
    images = get_images(soup)
    coords = get_coordinate(html_content)
    project_info = get_project_info(soup)
    agent_info = get_agent_info(soup)
    sub_info = get_sub_info(soup)
    specs = get_specs(soup)

    spec_data = {}
    for key, value in specs.items():
        mapped_key = SPEC_KEY_MAPPING.get(key, key)
        if mapped_key not in ['price', 'area']:
            spec_data[mapped_key] = value
    verified_tag = soup.find("div", class_="re__pr-stick-
listing-verified")
    verified = "verified" if verified_tag else "unverified"

    data = {
        "post_id": post_id,
        "property_url": url,
        "transaction_type": classify_transaction_type(url),
        "property_category": classify_property_type(url),
        "title": title,
        "address": address,
        "latitude": coords.get('latitude'),
        "longitude": coords.get('longitude'),
        "price": specs.get("khoang_gia"),
        "price_per_spm": price_per_spm,
        "area": specs.get("dien_tich"),
        "spec": spec_data,
        "description": description,
        "images": images,
        "project_info": project_info,
        "date_posted": sub_info.get("ngay_dang"),
        "date_expired": sub_info.get("ngay_het_han"),
        "news_type": sub_info.get("loai_tin"),
        "contact_info": agent_info,
        "verified_status": verified,
        "scraped_at": datetime.now().isoformat()
    }
    return data
```

Hình 3.6: Đoạn mã dùng để lấy các thông tin tin bất động sản

3.1.5 Lưu trữ dữ liệu MongoDB Atlas

Sau khi dữ liệu đã được bóc tách, toàn bộ thông tin được đóng gói dưới dạng một đối tượng để đẩy lên MongoDB Atlas. Đây là cở sở dữ liệu NoSQL dựa trên đám mây, giúp hệ thống có khả năng mở rộng tốt và lưu trữ linh hoạt các dữ liệu không cấu trúc từ website.



Hình 3.7: Cấu trúc dữ liệu được lưu trữ trên MongoDB

3.2 Mô tả dữ liệu

Tên biến	**Mô tả**	**Kiểu dữ liệu**
post_id	ID của bài đăng	string
property_url	Đường dẫn bài đăng	string
property_category	Loại bất động sản	string
transaction_type	Loại giao dịch	string
title	Tiêu đề	string
address	Địa chỉ	string
latitude	Vĩ độ	float
longitude	Kinh độ	float
price	Giá cả	string
price_per_spm	Đơn giá trên mỗi m^2	string
area	Diện tích	string

description	Nội dung mô tả	string
images	Danh sách hình ảnh	array
date_posted	Ngày đăng	date
date_expired	Ngày hết hạn	date
news_type	Loại tin tức	string
verified_status	Tình trạng tin xác thực	string
spec	Chứa thông tin đặc điểm bất động sản	object
contact_info	Chứa thông tin môi giới	object
project_info	Chứa thông tin dự án	object

Bảng 3.1: Bảng mô tả các biến và kiểu dữ liệu

Tên biến	**Mô tả**	**Kiểu dữ liệu**
name	Tên dự án	string
status	Tình trạng dự án	string
price	Khoảng giá	string
investor	Nhà đầu tư	string
image	URL hình ảnh dự án	string
project_url	URL dự án	string
listing_count	Số tin bất động sản có trong dự án	int

Bảng 3.2: Thông tin mô tả bảng dự án (project_info)

Tên biến	Mô tả	Kiểu dữ liệu
bedroom	Số phòng ngủ	string
bathroom	Số phòng vệ sinh	string
num_floor	Số tầng	string
orientation	Hướng nhà	string
balcony_direction	Hướng ban công	string
front_width	Chiều rộng mặt tiền	string
road_width	Chiều dài đường vào	string
legal	Pháp lý	string
furniture	Nội thất	string
exdate	Thời gian dự kiến vào ở	string
electricity	Thỏa thuận chi phí điện	string
water	Thỏa thuận chi phí nước	string
internet	Thỏa thuận chi phí internet	string
utilities	Tiện ích	string

Bảng 3.3: Thông tin mô tả bảng đặc diểm (spec)

Tên biến	Mô tả	Kiểu dữ liệu
name	Tên môi giới	string
profile_url	URL môi giới	string
avartar_url	URL hình ảnh môi giới	string
phone_invisible	Số điện thoại	string
zalo_url	URL zalo môi giới	string
join_duration	Thời gian tham gia	string
listings	Số tin đang có	int

Bảng 3.4: Thông tin mô tả bảng người môi giới (contact_info)

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

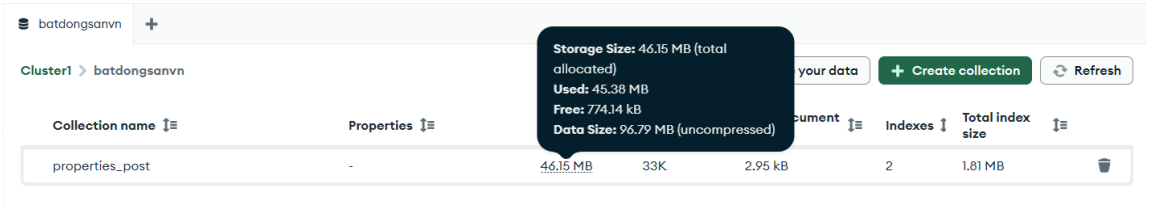
4.1 Giới thiệu

Trong phần này, những kết quả đã đạt được từ việc áp dụng công cụ Selenium để thu thập dữ liệu từ website batdongsan.com.vn sẽ được mô tả chi tiết trong phần này.

4.2 Kết quả thu thập dữ liệu

Sau khi hoàn thành quá trình thu thập, tổng cộn dữ liệu đã thu được khoảng 32.794 bản ghi thông tin bài đăng bất động sản. Dữ liệu này bao gồm đa dạng các loại hình từ nhà đất bán đến nhà đất cho thuê.

- Tổng số bản ghi: 32.794 tin đăng
- Tổng số giờ cho chạy thực tế: 46 tiếng
- Thời gian thu thập trung bình mỗi bài là 5 giây/1 tin
- Tỷ lệ trích xuất thành công: ~98% (Một số ít tin đăng bị lỗi do cấu trúc HTML)
- Dung lượng lưu trữ xấp xỉ 47MB dữ liệu văn bản thuần khi đã được nén



Hình 4.1: Tổng quan dữ liệu được lưu trữ trên MongDB Atlas

4.3 Truy vấn dữ liệu

```
>_MONGOSH
> db.properties_post.find(
    { property_category: "Villa/Townhouse" },
    { _id: 0, title: 1, area: 1, price: 1 }
  )
< {
    title: 'Quỹ hàng đợt cuối chiết khấu 18%, tặng xe điện VF8/VF9 Vinhomes Đan Phượng. LH:0937 996 ***',
    price: '15,6 tỷ',
    area: '104 m²'
  }
  {
    title: 'Bán nhà biệt thự tại Swan Bay, 23.2 tỷ VND, 320m2 - Zone 4.3 : LH:0909 165 ***',
    price: '23,2 tỷ',
    area: '320 m²'
  }
  {
    title: 'TỔNG HỢP CÁC CĂN BIỆT THỰ SONG LẬP, ĐƠN LẬP, CĂN GÓC VỊ TRÍ ĐẸP, GIÁ TỐT TẠI VINHOMES OCEAN PARK 2',
    price: '17,6 tỷ',
    area: '120 m²'
  }
  {
    title: 'CHIẾT KHẤU 20-25%, QUỸ CĂN ĐẸP VÀ RẺ NHẤT DỰ ÁN, NỘP 30% NHẬN NHÀ. LIÊN HỆ0942 906 ***XEM DỰ ÁN Ạ',
    price: '17,4 tỷ',
    area: '120 m²'
  }
  {
    title: 'Duy nhất căn Liền kề TV2BSP-08 Shop Vinhomes Cổ Loa 75m2 2 mặt tiền rẻ nhất DA chỉ 15 tỷ CK 19%',
    price: '18 tỷ',
    area: '75 m²'
  }
  {
    title: 'Suất ngoại giao chiết khấu 18% villas rừng song lập, đơn lập, shophouse. CDT Ecopark',
    price: '16 tỷ',
    area: '220 m²'
  }
```

Hình 4.2: Tìm tất cả các bất động sản thuộc loại Villa/Townhouse

```
>_MONGOSH
> db.properties_post.find(
    { address: /Hải Phòng/i },
    { _id: 0, title: 1, address: 1, price: 1 }
  )
< {
    title: 'Vinhomes Dương Kinh (240ha) giá F1 - 5 tỷ 4 tầng, vị trí vàng, chiết khấu sâu, quà tặng khủng',
    address: 'Dự án Vinhomes Golden City, Phường Hòa Nghĩa, Dương Kinh, Hải Phòng',
    price: '5 tỷ'
  }
  {
    title: 'Giỏ hàng mới Vin Dương Kinh giá tốt nhất từ 4.9 tỷ tặng VF3, chiết khấu trực tiếp từ PKD Chủ đầu tư',
    address: 'Vinhomes Golden City, Phường Hòa Nghĩa, Dương Kinh, Hải Phòng',
    price: '4,9 tỷ'
  }
  {
    title: 'Sở Hữu Nhà Vườn Độc Bản - Sống Xanh Sóng Đẳng Cấp Tại Biệt Thự Liền Kề Emerald Symphony',
    address: 'Emerald Symphony, Hoa Động, Thủy Nguyên, Hải Phòng',
    price: 'Thỏa thuận'
  }
  {
    title: 'Quỹ căn giá tốt nhất - Dự án của Doji Land: Emerald Symphony',
    address: 'Emerald Symphony, Hoa Động, Thủy Nguyên, Hải Phòng',
    price: '9,8 tỷ'
  }
  {
    title: 'CHUYỂN NHƯỢNG CĂN HỘ MINATO RẺ NHẤT DỰ ÁN - CÓ SẴN HỢP ĐỒNG THUÊ LÂU DÀI',
    address: 'The Minato Residence, Vĩnh Niệm, Lê Chân, Hải Phòng',
    price: '2,6 tỷ'
  }
  {
    title: 'Chính chủ bán lô đất 2 mặt ngõ dt 826m2 Hòa Bình, Thủy Nguyên. Giá 15tr/m2',
    address: 'Phường Hòa Bình, Thủy Nguyên, Hải Phòng',
    price: '15 triệu/m²'
  }
  {
    title: 'Bán Shophouse Hoàng Huy New City, 17 tỷ VND, 96m2, hàng hiếm tại Thủy Nguyên, Hải Phòng',
    address: 'Hoàng Huy New City, Phường Dương Quan, Thủy Nguyên, Hải Phòng'
```

Hình 4.3: Tìm các bài đăng tin bán nhà ở Hải Phòng

```
>_MONGOSH
> db.properties_post.find(
    { "spec.bathroom": /3|4|5/ },
    { _id: 0, title: 1, "spec.bathroom": 1 }
  )
< {
    title: 'Shophouse Vin Cổ Loa CĐT CK 22%-33% từ 15.8 tỷ HTLS 2 năm, 2MT, đường lớn, gần công viên, chung cư',
    spec: {
      bathroom: '5 phòng'
    }
  }
  {
    title: 'Duy nhất căn Liền kề TV2BSP-08 Shop Vinhomes Cổ Loa 75m2 2 mặt tiền rẻ nhất DA chỉ 15 tỷ CK 19%',
    spec: {
      bathroom: '5 phòng'
    }
  }
  {
    title: 'Suất ngoại giao chiết khấu 18% villas rừng song lập, đơn lập, shophouse. CĐT Ecopark',
    spec: {
      bathroom: '5 phòng'
    }
  }
  {
    title: 'Nhà riêng tại Hưng Long, Long Thượng, Cần Giuộc, Long An, giá siêu hời chỉ với 1,86 tỷ, 50m2',
    spec: {
      bathroom: '3 phòng'
    }
  }
  {
    title: 'Siêu phẩm liền kề xẻ khe San Hô Vinhomes Ocean Park 2, 98m2, Đông Nam, giá 13,4 tỷ, kinh doanh đỉnh',
    spec: {
      bathroom: '5 phòng'
    }
  }
```

Hình 4.4: Tìm nhà đất có hơn 3 phòng vệ sinh

```
>_MONGOSH
> db.properties_post.find(
    { "spec.balcony_direction": "Bắc" },
    { _id: 0, title: 1, "spec.balcony_direction": 1 }
  )
< {
    title: 'Nhà riêng tại Hưng Long, Long Thượng, Cần Giuộc, Long An, giá siêu hời chỉ với 1,86 tỷ, 50m2',
    spec: {
      balcony_direction: 'Bắc'
    }
  }
  {
    title: 'Bán căn skylinked villa- 3PN+ 1, 2WC (198,2m2), giá bán 13.085tỷ. Căn hộ xe hơi chạy lên tận nhà',
    spec: {
      balcony_direction: 'Bắc'
    }
  }
  {
    title: 'Capital Square, căn hộ view sông ngay cầu sông Hàn, cạnh Vincom. Mở bán đợt đầu chiết khấu đến 17%',
    spec: {
      balcony_direction: 'Bắc'
    }
  }
  {
    title: 'Nam Long mở bán căn hộ ven sông Mizuki Park, liền kề Phú Mỹ Hưng',
    spec: {
      balcony_direction: 'Bắc'
    }
  }
  {
    title: 'Cập nhật giỏ hàng T12/2025 giá tố nhà phố & biệt thự Vinhomes Grand Park, siêu phẩm view sông',
    spec: {
      balcony_direction: 'Bắc'
    }
  }
  {
    title: 'Quỹ 200 căn chuyển nhượng 1PN, 2PN, 3PN, 4PN giá tốt nhất thị trường - Hỗ trợ lãi suất 0% 30 tháng',
```

Hình 4.5: Tìm bài đăng nhà có ban công hướng Bắc

```
>_MONGOSH
    }
  }
  Type "it" for more
> db.properties_post.find(
    { "project_info.listing_count": { $gt: 50 } },
    { _id: 0, title: 1, "project_info.listing_count": 1 }
  ).sort({ "project_info.listing_count": -1 })
< {
    title: 'Lần đầu tiên CĐT MIK Group mở bán chung cư cao cấp tại Vin Ocean Park 2 với mức giá cực hấp dẫn',
    project_info: {
      listing_count: 944
    }
  }
  {
    title: 'Chính thức nhận booking chung cư cao cấp của CĐT MIK Group tại Vin Ocean Park 2',
    project_info: {
      listing_count: 944
    }
  }
```

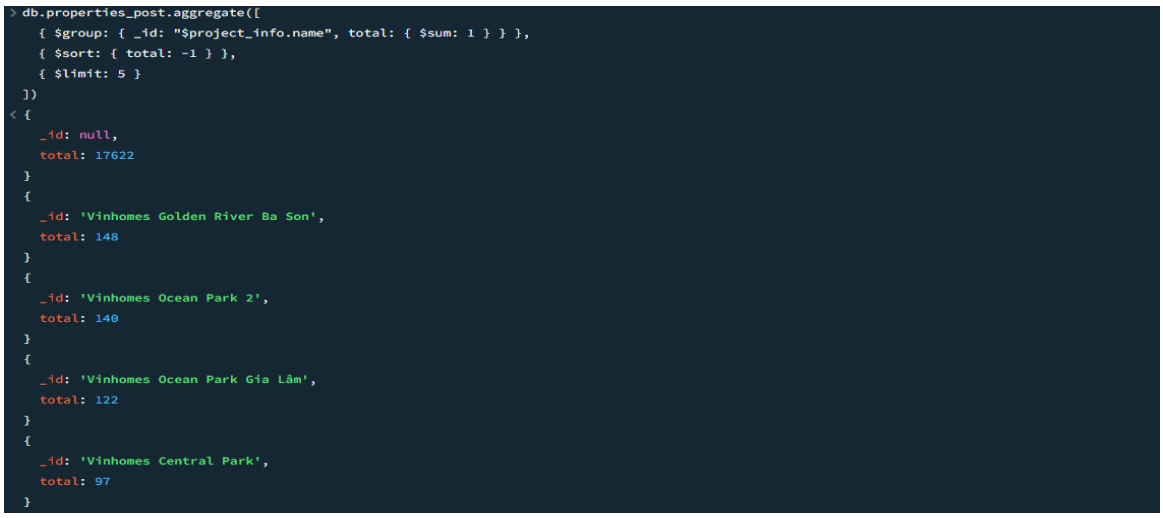
Hình 4.6: Dự án có nhiều tin đăng nhất

```
>_MONGOSH
> db.properties_post.find(
    { $expr: { $gte: [{ $size: "$images" }, 24] } },
    { _id: 0, title: 1 }
  )
< {
    title: 'Quá Đẳng Cấp! Sở hữu view triệu đô tại Masteri Park Place CT5 - Sinh lời bền vững. Gọi0902 999 ***'
  }
  {
    title: 'CHIẾT KHẤU 20-25%, QUỸ CĂN ĐẸP VÀ RẺ NHẤT DỰ ÁN, NỘP 30% NHẬN NHÀ. LIÊN HỆ0942 906 ***XEM DỰ ÁN Ạ'
  }
  {
    title: 'Căn hộ 3PN Phú Đông Sky Garden "nhận nhà ở ngay - nhận sổ liền tay - giá tốt nhất thị trường'
  }
  {
    title: 'Căn góc Sonata hướng sông sát 2 tòa chung cư, thích hợp cho thuê, kinh doanh, buôn bán. CK 10%'
  }
```

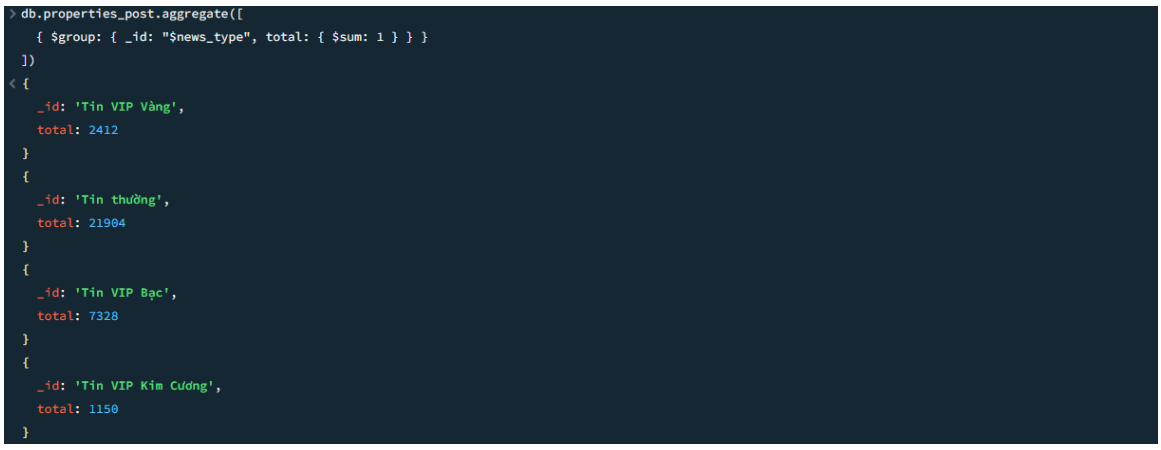
Hình 4.7: Những bài đăng có nhiều hơn 24 hình ảnh

```
>_MONGOSH
> db.properties_post.find(
    { date_posted: { $gte: "2025-12-01" } },
    { _id: 0, title: 1, date_posted: 1 }
  )
< {
    title: 'Quá Đẳng Cấp! Sở hữu view triệu đô tại Masteri Park Place CT5 - Sinh lời bền vững. Gọi0902 999 ***',
    date_posted: '22/12/2025'
  }
  {
    title: 'Bán đất tại phố Hòa Bình, Thanh Miếu, Phú Thọ diện tích 264 m2',
    date_posted: '22/12/2025'
  }
  {
    title: 'Bán nhà biệt thự tại Swan Bay, 23.2 tỷ VND, 320m2 - Zone 4.3 : LH:0909 165 ***',
    date_posted: '23/12/2025'
  }
  {
    title: 'GREEN SKYLINE  CĂN HỘ XÂY XONG MỚI BÁN HIẾM HOI GIÁ TỪ 3 TỶ LH0939 720 ***',
    date_posted: '23/12/2025'
  }
  {
    title: 'Chủ bán kín! Nhà đẹp 5 tầng khu Vip Cống Lở, 4mX20m, hẻm 8m có lề, chỉ 9.5 tỷ TL',
    date_posted: '23/12/2025'
  }
  {
    title: 'TỔNG HỢP CÁC CĂN BIỆT THỰ SONG LẬP, ĐƠN LẬP, CĂN GÓC VỊ TRÍ ĐẸP, GIÁ TỐT TẠI VINHOMES OCEAN PARK 2',
    date_posted: '22/12/2025'
  }
  {
    title: 'Shophouse Vin Cổ Loa CĐT CK 22%-33% từ 15.8 tỷ HTLS 2 năm, 2MT, đường lớn, gần công viên, chung cư',
    date_posted: '23/12/2025'
  }
  {
    title: 'CHIẾT KHẤU 20-25%, QUỸ CĂN ĐẸP VÀ RẺ NHẤT DỰ ÁN, NỘP 30% NHẬN NHÀ. LIÊN HỆ0942 906 ***XEM DỰ ÁN Ạ',
    date_posted: '22/12/2025'
  }
```

Hình 4.8: Tìm các tin đăng sau ngày 01/12/2025



Hình 4.9: Top 5 dự án có nhiều tin đăng



Hình 4.10: Đếm tổng số tin theo loại tin

Hình 4.11: Tìm những tin thuộc Apartment và có hướng Đông-Bắc

Hình 4.12: Thống kê mật độ tin theo khu vực

4.4 Phân tích dữ liệu bất động sản

Tiến hành kết nối đến MongoDB Atlas để lấy dữ liệu về đưa vào DataFrame

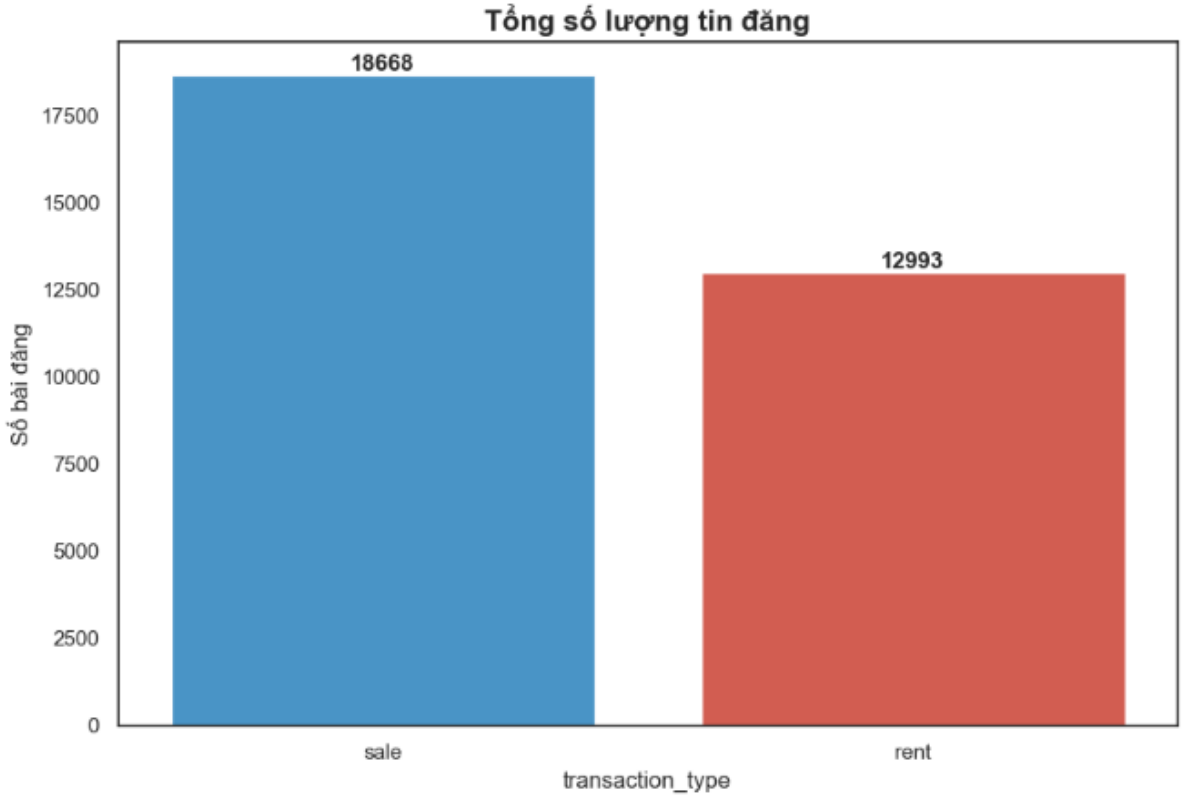
```
import sys
from pathlib import Path
import pandas as pd
import os
from dotenv import load_dotenv
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from pymongo import MongoClient

PROJECT_ROOT = Path.cwd().parent
sys.path.append(str(PROJECT_ROOT))
load_dotenv(dotenv_path=os.path.abspath("../.env"))

user = os.getenv("MONGO_USER")
pw = os.getenv("MONGO_PASS")
cluster = os.getenv("MONGO_CLUSTER")
db_name = os.getenv("MONGO_DB")
col_name = os.getenv("MONGO_COLLECTION")

uri =
f"mongodb+srv://{user}:{pw}@{cluster}/?retryWrites=true&w=
majority"

client = MongoClient(uri)
db = client[db_name]
col = db[col_name]
data = list(col.find({}, {"_id": 0}))

df = pd.DataFrame(data)
df = pd.json_normalize(data, sep=".")
```

Hình 4.13: Đoạn mã dùng để kết nối và lấy dữ liệu về DataFrame

```
Index: 31661 entries, 0 to 32792
Data columns (total 47 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   post_id                         31661 non-null  object
 1   property_url                    31661 non-null  object
 2   transaction_type                31661 non-null  object
 3   property_category               31661 non-null  object
 4   title                           31661 non-null  object
 5   address                         31661 non-null  object
 6   latitude                        31661 non-null  float64
 7   longitude                       31661 non-null  float64
 8   price                           31660 non-null  object
 9   price_per_spm                   20442 non-null  object
 10  area                            31661 non-null  object
 11  description                     31661 non-null  object
 12  images                          31660 non-null  object
 13  date_posted                     31661 non-null  datetime
 14  date_expired                    31661 non-null  datetime
 15  news_type                       31661 non-null  object
 16  verified                        26117 non-null  object
 17  scraped_at                      31661 non-null  object
 18  spec.bedroom                    18655 non-null  object
 19  spec.bathroom                   18357 non-null  object
 20  spec.legal                      16185 non-null  object
 21  spec.furniture                  17101 non-null  object
 22  project_info.name               14354 non-null  object
 23  project_info.status             14352 non-null  object
 24  project_info.investor           14354 non-null  object
 25  project_info.image              14354 non-null  object
 26  project_info.project_url        14354 non-null  object
 27  project_info.listing_count      14354 non-null  float64
 28  contact_info.name               31657 non-null  object
 29  contact_info.profile_url        31657 non-null  object
 30  contact_info.avatar_url         29530 non-null  object
 31  contact_info.phone_invisible    31661 non-null  object
 32  contact_info.zalo_url           31661 non-null  object
 33  contact_info.join_duration      13203 non-null  object
 34  contact_info.listings           13203 non-null  object
 35  spec.front_width                11081 non-null  object
 36  spec.road_width                 9327 non-null   object
 37  spec.num_floor                  9674 non-null   object
 38  project_info.price              7972 non-null   object
 39  spec.orientation                8685 non-null   object
 40  spec.balcony_direction          5995 non-null   object
 41  spec.exdate                     3246 non-null   object
 42  spec.electricity                2696 non-null   object
 43  spec.water                      2606 non-null   object
 44  spec.internet                   1882 non-null   object
 45  spec.utilities                  3500 non-null   object
 46  verified_status                 5544 non-null   object
dtypes: datetime64[ns](2), float64(3), object(42)
```

Hình 4.14: Mô tả tổng quan về dữ liệu trong DataFrame

Để làm rõ bức tranh toàn cảnh và nắm bắt các xu hướng đặc thù của thị trường, chúng tôi tiến hành tách nhóm dữ liệu bất động sản thành hai phân khúc chính: cho thuê (rent) và mua bán (sale). Việc tách biệt này không chỉ giúp nhận diện các loại hình sản phẩm phổ biến mà còn cho phép so sánh sự khác biệt rõ rệt về mục đích sử dụng, hành vi khách hàng và giá trị đầu tư giữa hai thị trường.



Hình 4.15: Chia nhóm dữ liệu bất động sản thành hai phân khúc chính

Nhằm nâng cao chất lượng phân tích dữ liệu, các cột Price và Address được tiến hành chuẩn hóa như sau:

```
def split_address(addr):
    if pd.isna(addr): return None, None, None
    parts = [p.strip() for p in str(addr).split(',')]
    province = parts[-1] if len(parts) >= 1 else None
    district = parts[-2] if len(parts) >= 2 else None
    ward = parts[-3] if len(parts) >= 3 else None
    return ward, district, province

df[['ward', 'district', 'province']] =
df['address'].apply(lambda x: pd.Series(split_address(x)))
```

Hình 4.16: Đoạn mã dùng để tách địa chỉ

```
def convert_price_sale(x):
    if pd.isna(x): return np.nan, "valid"
    text = str(x).lower().strip().replace(",", ".")
    if "thỏa thuận" in text: return np.nan, "negotiable"

    try:
        match = re.search(r"(\d+\.?\d*)", text)
        if not match or "/m" in text: return np.nan,
"error"

        num = float(match.group(1))
        if "tỷ" in text: return num * 1000, "valid"
        if "triệu" in text: return num, "valid"
        return num, "valid"
    except:
        return np.nan, "error"

res_sale = df_sale["price"].apply(convert_price_sale)
df_sale["price"] = [x[0] for x in res_sale]
df_sale["price_status"] = [x[1] for x in res_sale]

df_sale = df_sale[df_sale["price_status"] !=
"error"].drop(columns=["price_status"])
```

Hình 4.17: Đoạn mã dùng để chuyển đổi cột price từ kiểu string sang float (sale)

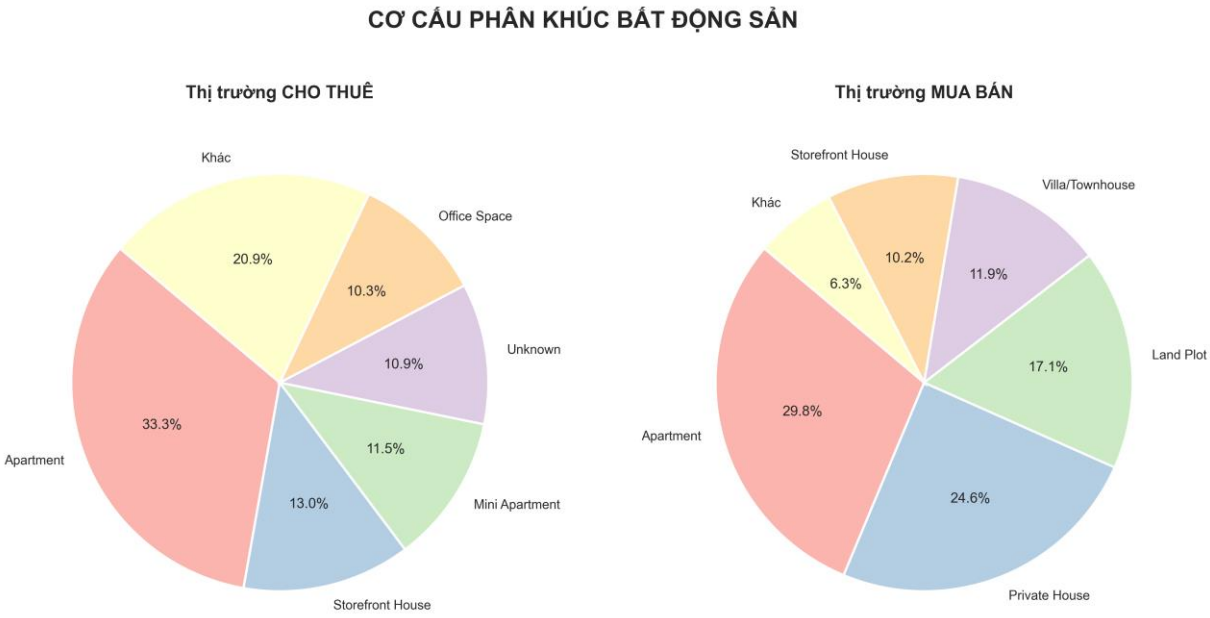
```
def convert_price_rent(x):
    if pd.isna(x): return np.nan, "valid"
    text = str(x).lower().strip().replace(",", ".")
    if "thỏa thuận" in text: return np.nan, "negotiable"

    try:
        match = re.findall(r"(\d+\.?\d*)", text)
        if not match: return np.nan, "error"

        return float(match[0]), "valid"
    except:
        return np.nan, "error"

res_rent = df_rent["price"].apply(convert_price_rent)
df_rent["price"] = [x[0] for x in res_rent]
df_rent["price_status"] = [x[1] for x in res_rent]

df_rent = df_rent[df_rent["price_status"] !=
"error"].drop(columns=["price_status"])
```

Hình 4.18: Đoạn mã dùng để chuyển đổi cột price từ kiểu string sang float (rent)

```
def get_category_counts(df, top_n=5):
    counts = df['property_category'].value_counts()
    if len(counts) > top_n:
        top_counts = counts[:top_n]
        others_count = counts[top_n:].sum()
        top_counts['Khác'] = others_count
        return top_counts
    return counts

rent_categories = get_category_counts(df_rent)
sale_categories = get_category_counts(df_sale)

colors = plt.get_cmap('Pastel1').colors

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 8))
fig.suptitle('CƠ CẤU PHÂN KHÚC BẤT ĐỘNG SẢN', fontsize=20,
fontweight='bold', y=1.05)

ax1.pie(rent_categories, labels=rent_categories.index,
autopct='%1.1f%%',
        startangle=140, colors=colors,
wedgeprops={'edgecolor': 'white', 'linewidth': 2})
ax1.set_title('Thị trường CHO THUÊ', fontsize=15,
fontweight='bold', pad=20)

ax2.pie(sale_categories, labels=sale_categories.index,
autopct='%1.1f%%',
        startangle=140, colors=colors,
wedgeprops={'edgecolor': 'white', 'linewidth': 2})
ax2.set_title('Thị trường MUA BÁN', fontsize=15,
fontweight='bold', pad=20)

plt.tight_layout()
plt.show()
```

Hình 4.19: Đoạn mã dùng để lấy top 5 loại hình bất động sản phổ biến nhất

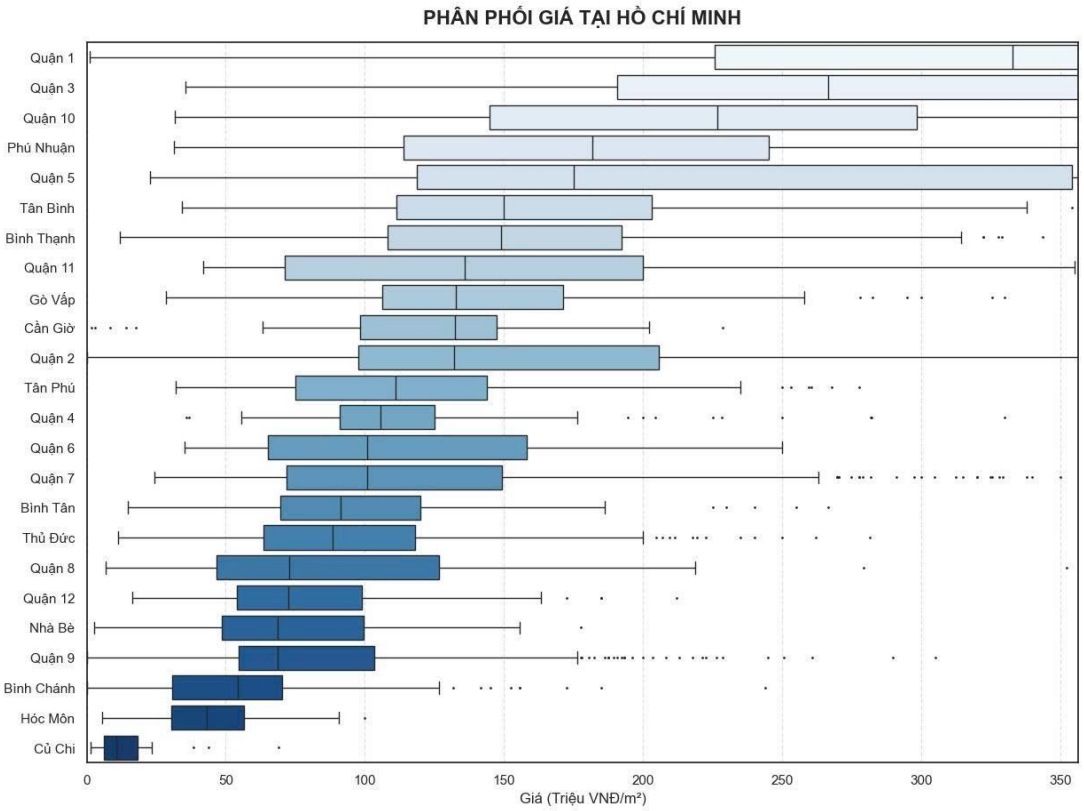
Bắt đầu quá trình phân tích:



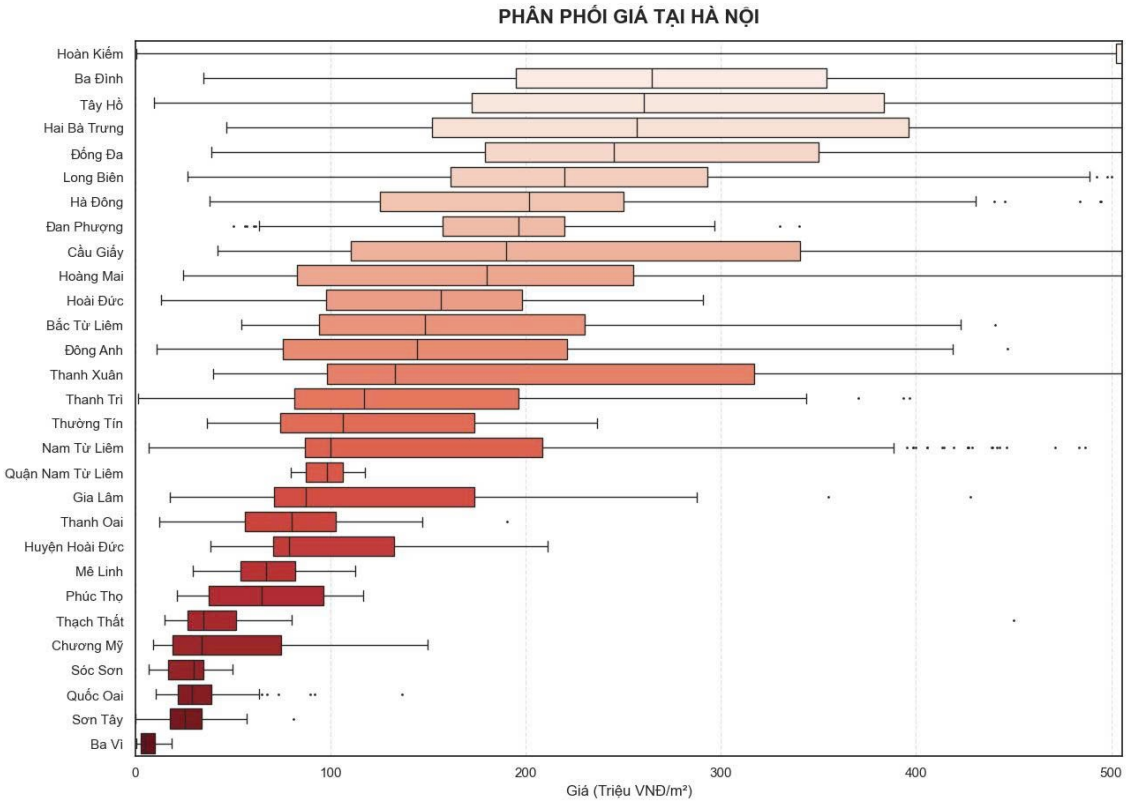
Hình 4.20: Biểu đồ cơ cấu phân khúc bất động sản

Nhận xét biểu đồ cơ cấu phân khúc bất động sản:

- Phân khúc căn hộ (Apartment) đang xu thế dẫn đầu ở cả hai thị trường bán và thuê lần lượt là 33.3% và 29.8%.
- Nhà riêng (Private House) chiếm tỉ trọng 24.6% lớn thứ hai trong loại giao dịch mua, điều này phản ánh tâm lý người dân rất thích sỡ hữu nhà riêng ngược lại bên cho thuê thì chủ yếu tập trung phân khúc nhà mặt phố (Storefront House) phục vụ chủ yếu cho mục đính kinh doanh, thương mại.
- Ta thấy rõ rệt phân khúc biệt thự (Villa/Townhouse) bên mua bán là 11.9% nhưng lại không nằm trong nhóm chính bên cho thuê. Điều này cho thấy biệt thự chủ yếu là tài sản giá trị cao dùng để đầu tư dài hạn hoặc định cư, ít phổ biến tại trong thị trường cho thuê.



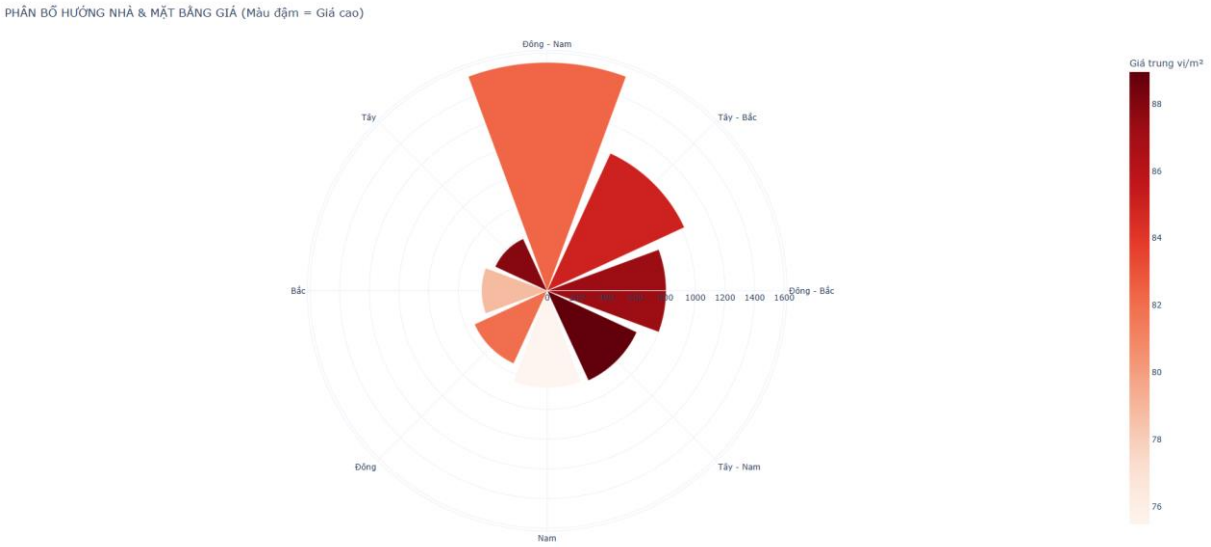
Hình 4.21: Phân phối giá trên m² ở các quận/huyện tại Tp.Hồ Chí Minh



Hình 4.22: Phân phối giá ở các quận/huyện tại Hà Nội

So sánh hai biểu đồ phân phối giá trên m² ở các quận/huyện tại TP.HCM và Hà Nội:

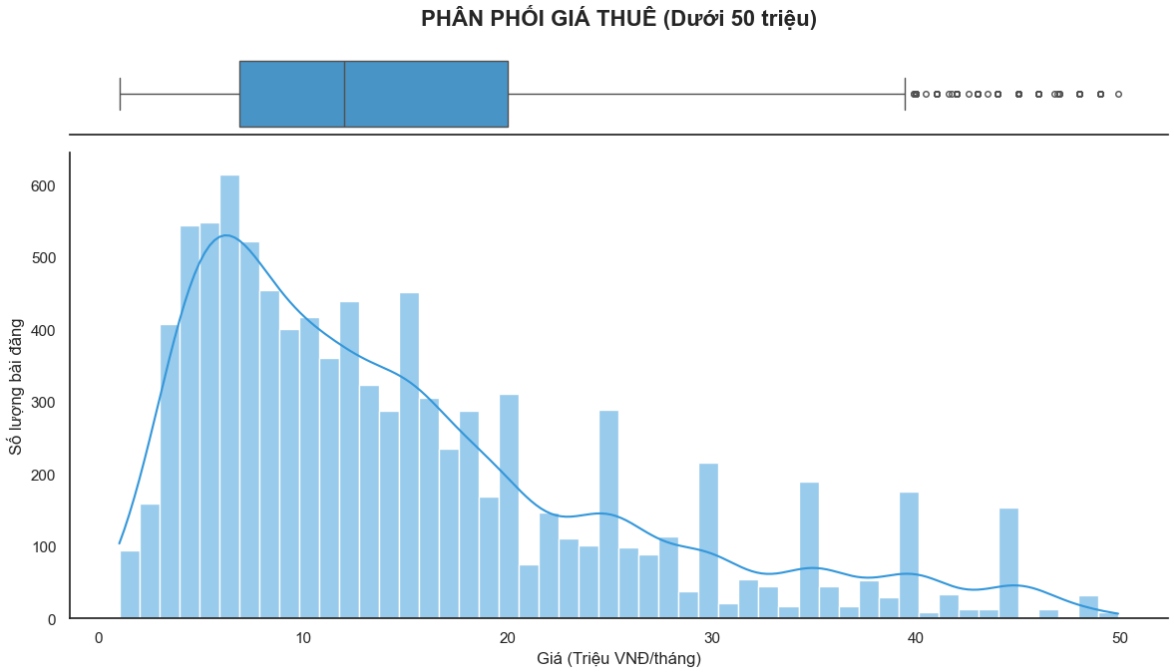
- Hà Nội có mức giá trần cao hơn TP.HCM, trong khi TP.HCM có phân bố giá trải rộng đều hơn ở nhiều quận.
- Cả hai thành phố đều thể hiện xu hướng: giá giảm dần từ khu vực trung tâm ra ngoại thành.
- Sự hiện diện của nhiều giá trị ngoại lai cho thấy thị trường bất động sản ở cả hai đô thị đều có mức độ phân hóa cao.



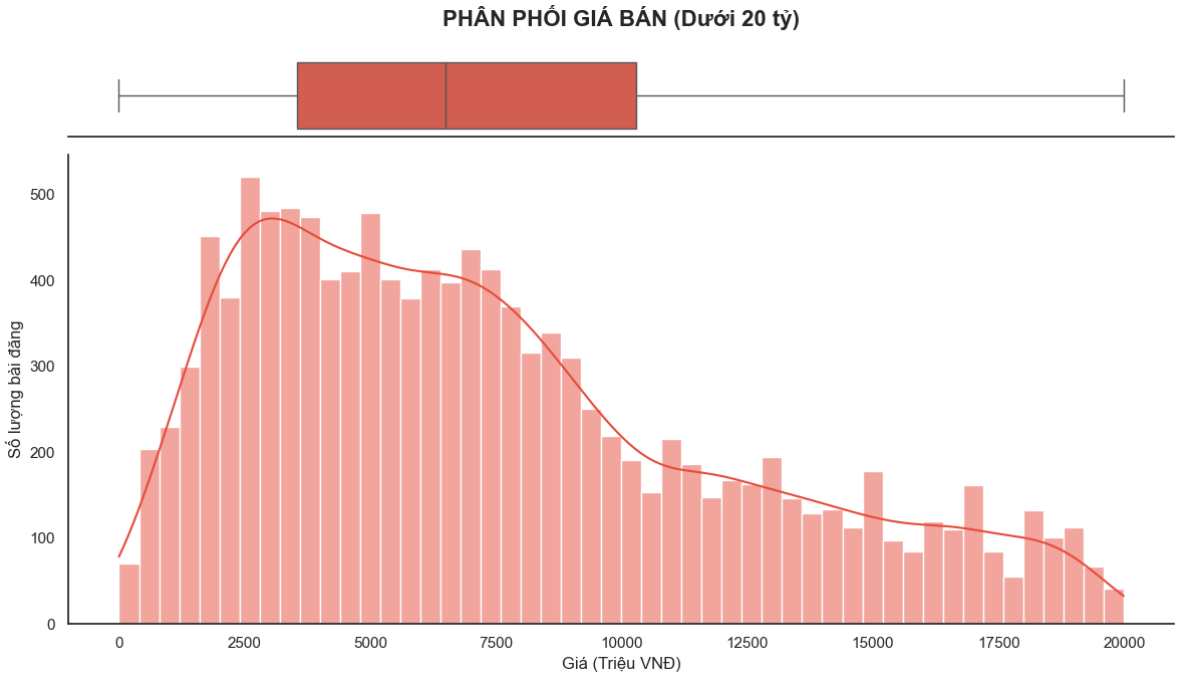
Hình 4.23: Phân bố hướng nhà và mặt bằng giá trung vị

Nhận xét phân bố hướng nhà:

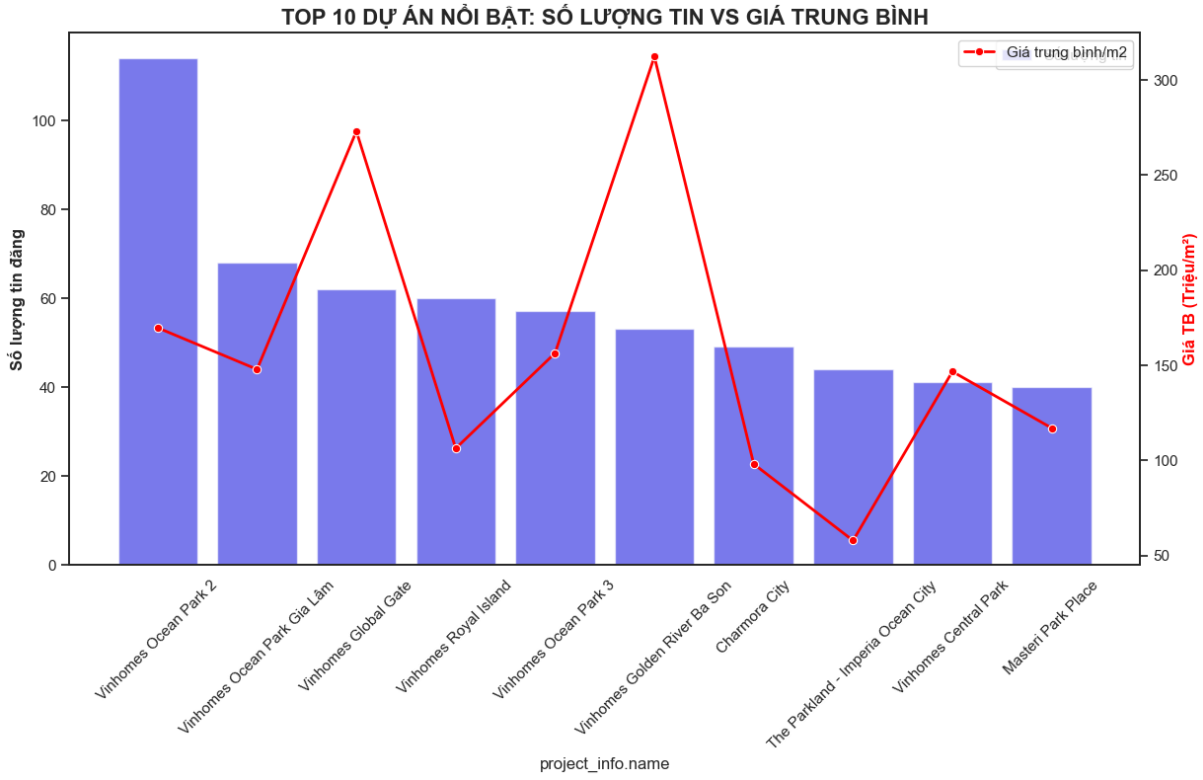
- Hướng Đông Nam có số lượng bất động sản nhiều nhất đồng thời giá trung bình cao, cho thấy đây là hướng được ưa chuộng.
- Các hướng Tây Bắc và Đông Bắc cũng có mức giá tương đối cao.
- Nhìn chung, biểu đồ cho thấy hướng nhà có ảnh hưởng nhất định đến giá bất động sản.



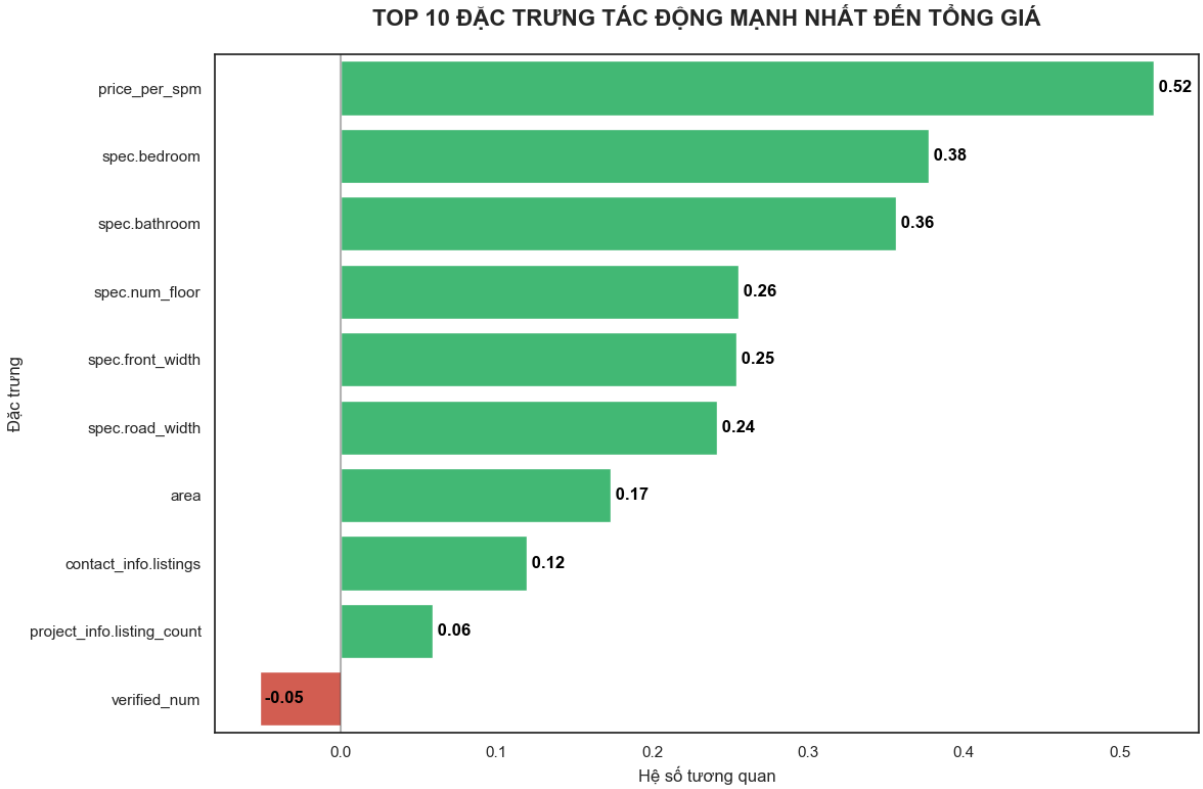
Hình 4.24: Phân phối giá thuê dưới 50 triệu VND



Hình 4.25: Phân phối giá thuê bán dưới 20 tỷ VND



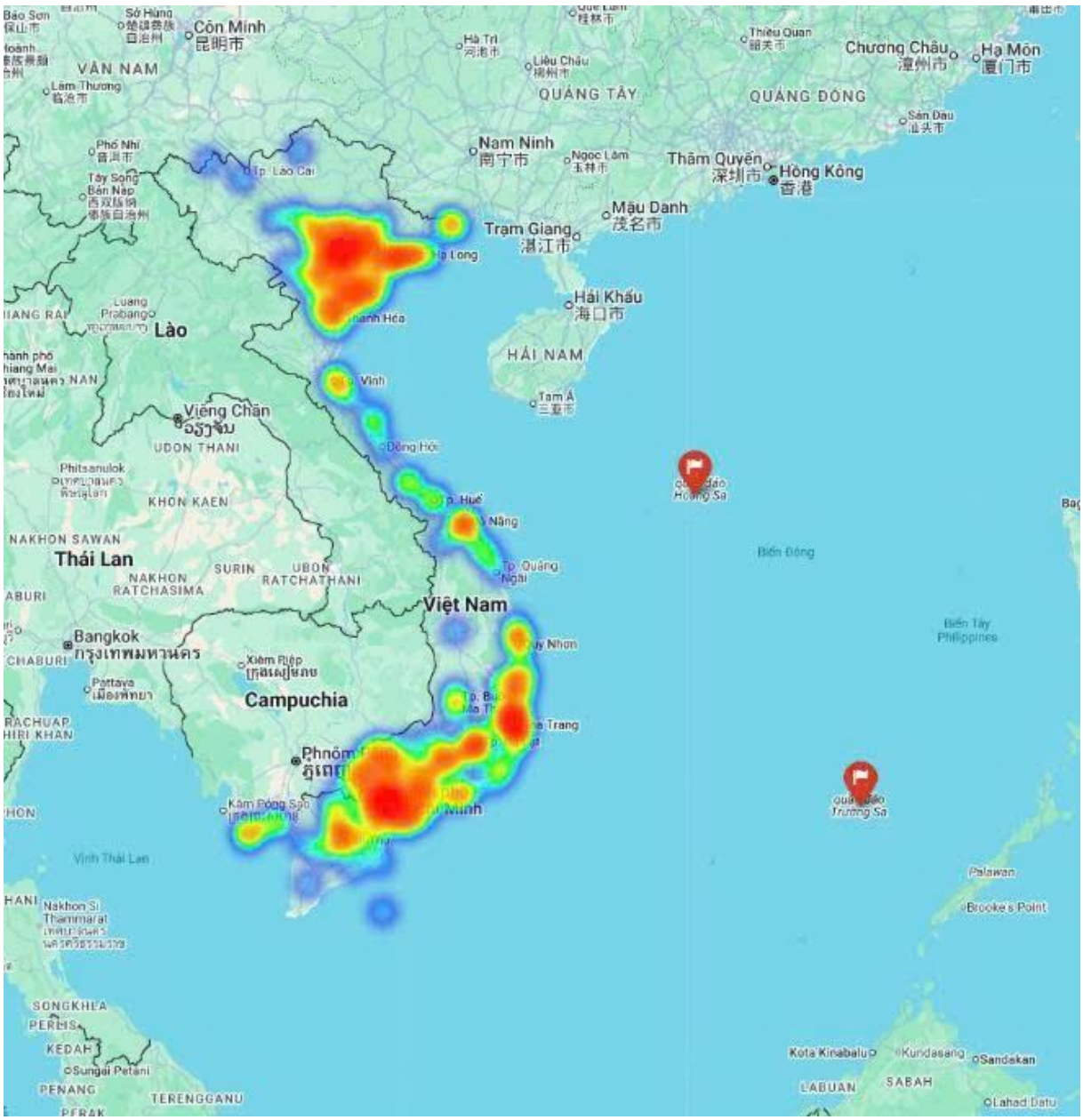
Hình 4.26: Top 10 dự án nổi bật với số lượng tin và giá trung bình



Hình 4.27: Top 10 đặc trưng có tương quan cao nhất với tổng giá bất động sản

Nhận xét:

- Về mối quan hệ giữa số lượng tin và giá bất động sản: Các dự án lớn như Vinhomes Ocean Park và Vinhomes Smart City có số lượng tin đăng cao, phản ánh mức độ quan tâm lớn của thị trường; tuy nhiên, các dự án có giá trung bình cao không nhất thiết đi kèm với số lượng tin đăng lớn, cho thấy sự phân hóa rõ rệt giữa tính thanh khoản và mức giá của từng dự án.
- Về mức độ ảnh hưởng của các đặc trưng đến giá: Giá trên mỗi mét vuông (price_per_spm) là yếu tố tác động mạnh nhất đến tổng giá bất động sản, tiếp theo là số phòng ngủ và số phòng tắm. Các đặc trưng về kích thước mặt tiền và độ rộng đường cũng có ảnh hưởng đáng kể, trong khi một số đặc trưng khác như mức độ xác thực tin đăng thể hiện tác động không rõ rệt.



Hình 4.28: Bản đồ nhiệt phân bố tin đăng bất động sản tại Việt Nam

Nhận xét:

Tin đăng bất động sản tập trung chủ yếu tại các đô thị lớn như Hà Nội, TP. Hồ Chí Minh và các tỉnh ven biển, phản ánh xu hướng đô thị hóa mạnh mẽ và sự chênh lệch đáng kể về mức độ sôi động của thị trường giữa các khu vực.

CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ

5.1. Kết luận

Sau một thời gian tập trung nghiên cứu và trực tiếp sử dụng các công cụ mã nguồn mở để thu thập dữ liệu và lưu trữ, đề tài đã hoàn thành các mục tiêu đề ra ban đầu, đạt được những kết quả như sau:

Về kỹ thuật thu thập dữ liệu, bộ công cụ crawler đã được xây dựng thành công dựa trên Selenium kết hợp với Undetected Chromedriver. Nhờ vào áp dụng những thư viện vào chương trình python đã vận hành ổn định nhờ cơ chế giả lập hành vi người dùng thông qua việc ngẫu nhiên hóa thời gian chờ giữa các thao tác truy cập.

Việc tích hợp BeautifulSoup vào module parser đã thực hiện hiệu quả việc bóc tách và trích xuất các trường thông tin phức tạp từ mã HTML thô.

Mô hình lưu trữ đã triển khai thành công lên nền tảng điện toán đám mây MongoDB Atlas. Việc thiết lập Compound Index giúp giải quyết hiệu quả vấn đề trùng lặp dữ liệu, đảm bảo tính toàn vẹn, nhất quán và duy nhất của kho dữ liệu thu thập được.

5.2. Hạn chế của đề tài

Do cơ chế bóc tách dựa trên cây DOM, hệ thống sẽ gặp khó khăn nếu website mục tiêu thay đổi cấu trúc thẻ HTML hoặc thiết kế lại giao diện. Khi đó, module parser cần phải được cập nhật thủ công để thích ứng.

Để ưu tiên tính an toàn và không bị chặn địa chỉ IP, thiết lập cài đặt cần phải tạo các khoảng nghỉ giữa mỗi lần truy cập. Chính sự thận trọng này vô tình làm giới hạn tốc độ thu thập khi cần xử lý một lượng dữ liệu khổng lồ trong thời gian ngắn.

5.3. Kiến nghị và Hướng phát triển

Để dự án thực sự hoàn thiện và cải thiện hơn trong tương lai, chúng tôi đề xuất một số hướng phát triển sau:

Sử dụng danh sách proxy xoay vòng kết hợp với các giải thuật đa luồng hoặc đa tiến trình sẽ giúp tăng đáng kể tốc độ thu thập dữ liệu, đồng thời vẫn đảm bảo an toàn và hạn chế nguy cơ bị chặn từ phía website mục tiêu.

Xây dựng dashboard trực quan hóa dữ liệu. Phát triển một giao diện web cho phép hiển thị các biểu đồ phân tích như biến động giá theo khu vực, mật độ tin đăng bất động sản hoặc bản đồ nhiệt dựa trên dữ liệu tọa độ GPS đã thu thập, từ đó hỗ trợ người dùng trong việc phân tích và ra quyết định.

TÀI LIỆU THAM KHẢO

[1] R. Mitchell, *Web Scraping with Python: Collecting More Data from the Modern Web*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.

[2] S. Munzert, C. Rubba, P. Meißner and D. Nyhuis, *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*, Chichester: Wiley, 2015.

[3] M. Grinberg, Flask Web Development: *Developing Web Applications with Python*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.

[4] R. T. Fielding, "Architectural styles and the design of network-based software architectures" Ph.D. dissertation, Dept. Inf. and Comput. Sci., Univ. of California, Irvine, CA, 2000.

[5] V. Krotov and J. Silva, "Legality and Ethics of Web Scraping" *Communications of the Association for Information Systems,* vol. 47, no. 1, p. 1–28, 2020.

[6] U. Gundecha, *Selenium Testing Tools Cookbook*, 2nd ed. Packt Publishing, 2016.

[7] A. Mesbah and A. van Deursen, "Invariant-based Automatic Testing of AJAX User Interfaces" *IEEE Transactions on Software Engineering,* vol. 38, no. 1, p. 210–229, 2012.

[8] S. Bradshaw, E. Brazil and K. Chodorow, *MongoDB: The Definitive Guide*, 3rd ed. Sebastopol, CA: O'Reilly Media, 2019.

[9] P. J. Sadalage and M. Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Boston, MA: Addison-Wesley, 2012.

[10] R. Cattell, "Scalable SQL and NoSQL Data Stores" *ACM SIGMOD Record,* vol. 39, no. 4, p. 12–27, 2011.

[11] D. J. Abadi, "Consistency Tradeoffs in Modern Distributed Database System Design," *IEEE Computer,* vol. 45, no. 2, p. 37–42, 2012.

[12] MongoDB Inc., *MongoDB Manual*, 2025. [Online]. Available: https://www.mongodb.com/docs/manual/. [Accessed 2025].

[13] L. Richardson, *Beautiful Soup Documentation*, 2025. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/. [Accessed 2025].

[14] SeleniumHQ, *Selenium WebDriver Documentation*, 2025. [Online]. Available: https://www.selenium.dev. [Accessed 2025].

[15] K. Banker, *MongoDB in Action*, 2nd ed. Shelter Island, NY: Manning Publications, 2016.

[16] MongoDB Inc., Replication, 2025. [Online]. Available: https://www.mongodb.com/docs/manual/replication/. [Accessed 2025].

[17] MongoDB Inc., Sharding, 2025. [Online]. Available: https://www.mongodb.com/docs/manual/sharding/. [Accessed 2025].

[18] MongoDB Inc., MongoDB Atlas Architecture, 2025. [Online]. Available: https://www.mongodb.com/docs/atlas/architecture/. [Accessed 2025].

PHỤ LỤC 1

```
from bs4 import BeautifulSoup
import re
import unidecode
from urllib.parse import urlparse
from datetime import datetime

SALE_PROPERTY_TYPES = {
    "ban-can-ho-chung-cu": "Apartment",
    "ban-can-ho-chung-cu-mini": "Mini Apartment",
    "ban-nha-rieng": "Private House",
    "ban-nha-biet-thu-lien-ke": "Villa/Townhouse",
    "ban-nha-mat-pho": "Storefront House",
    "ban-shophouse-nha-pho-thuong-mai": "Shophouse",
    "ban-dat-nen-du-an": "Project Land",
    "ban-dat": "Land Plot",
    "ban-trang-trai-khu-nghi-duong": "Farm/Resort",
    "ban-condotel": "Condotel",
    "ban-kho-nha-xuong": "Warehouse/Factory"
}

RENT_PROPERTY_TYPES = {
    "cho-thue-can-ho-chung-cu": "Apartment",
    "cho-thue-can-ho-chung-cu-mini": "Mini Apartment",
    "cho-thue-nha-rieng": "Private House",
    "cho-thue-nha-biet-thu-lien-ke": "Villa/Townhouse",
    "cho-thue-nha-mat-pho": "Storefront House",
    "cho-thue-shophouse-nha-pho-thuong-mai": "Shophouse",
    "cho-thue-dat": "Land Plot",
    "cho-thue-trang-trai-khu-nghi-duong": "Farm/Resort",
    "cho-thue-condotel": "Condotel",
    "cho-thue-kho-nha-xuong": "Warehouse/Factory",
    "cho-thue-van-phong": "Office Space",
    "cho-thue-cua-hang-ki-ot": "Shop/Kiosk",
    "cho-thue-phong-tro": "Room"
}

ALL_PROPERTY_TYPES = {**SALE_PROPERTY_TYPES,
**RENT_PROPERTY_TYPES}

TRANSACTION_TYPE_PATTERNS = {
    "sale": r'^ban-',
    "rent": r'^cho-thue-'
}
```

```
SPEC_KEY_MAPPING = {
    "khoang_gia": "price",
    "dien_tich": "area",
    "so_phong_ngu": "bedroom",
    "so_phong_tam_ve_sinh": "bathroom",
    "so_tang": "num_floor",
    "huong_nha": "orientation",
    "huong_ban_cong": "balcony_direction",
    "mat_tien": "front_width",
    "duong_vao": "road_width",
    "phap_ly": "legal",
    "noi_that": "furniture",
    "thoi_gian_du_kien_vao_o": "exdate",
    "muc_gia_dien": "electricity",
    "muc_gia_nuoc": "water",
    "muc_gia_internet": "internet",
    "tien_ich": "utilities"
}

def normalize_key(key):
    key = unidecode.unidecode(key).lower()
    key = re.sub(r'[\s,]+', '_', key)
    key = re.sub(r'[^\w_]', '', key)
    key = re.sub(r'_+', '_', key)
    return key.strip('_')


def remove_empty_fields(data):
    cleaned = {}
    for key, value in data.items():
        if value is None:
            continue
        if isinstance(value, str) and not value.strip():
            continue
        if isinstance(value, dict):
            cleaned_nested = remove_empty_fields(value)
            if cleaned_nested:
                cleaned[key] = cleaned_nested
            continue
        if isinstance(value, list) and not value:
            continue
        cleaned[key] = value
    return cleaned


def get_text(soup, selector, default=None):
    if soup is None:
        return default
    tag = soup.select_one(selector)
    return tag.get_text(strip=True) if tag else default

POST_ID_PATTERN = re.compile(r"pr(\d+)$")
def get_post_id(url):
    match = POST_ID_PATTERN.search(url)
    return match.group(1) if match else None
```

```
def classify_property_type(url):
    path = urlparse(url).path.strip('/').lower()
    sorted_types = sorted(
        ALL_PROPERTY_TYPES.items(),
        key=lambda x: len(x[0]),
        reverse=True
    )
    for path_segment, category in sorted_types:
        if path.startswith(path_segment):
            is_full_match = len(path) == len(path_segment)
            if is_full_match or path[len(path_segment)] in
('-'):
                return category
    return "Unknown"


def classify_transaction_type(url):
    path = urlparse(url).path.strip('/').lower()
    for transaction_type, pattern in
TRANSACTION_TYPE_PATTERNS.items():
        if re.match(pattern, path):
            return transaction_type
    return "Unknown"



def get_coordinate(html_content):
    geo = {}
    keys = ['latitude', 'longitude']
    for key in keys:
        pattern = rf'["\']?{re.escape(key)}["\']?\s*:\s*(-
?[\d\.]+)'
        match = re.search(pattern, html_content)
        if match:
            geo[key] = float(match.group(1).replace(' ', ''))
    return geo


def get_specs(soup):
    specs = {}
    for item in soup.select(".re__pr-specs-content-item"):
        label_tag = item.select_one(".re__pr-specs-content-
item-title")
        value_tag = item.select_one(".re__pr-specs-content-
item-value")
        if label_tag and value_tag:
            key =
normalize_key(label_tag.get_text(strip=True))
            value = value_tag.get_text(strip=True)
            specs[key] = value
    return specs


def get_sub_info(soup):
    sub_info = {}
    for item in soup.select("div.re__pr-short-info-item"):
        title_tag = item.select_one("span.title")
        value_tag = item.select_one("span.value")
        if title_tag and value_tag:
            key =
normalize_key(title_tag.get_text(strip=True))
            value = value_tag.get_text(strip=True)
            sub_info[key] = value
    return sub_info
```

```
def get_agent_info(soup):
    agent_info = {}
    contact_box = soup.find("div", class_="re__ldp-contact-
box")
    agent_infor = contact_box.find("div", class_="re__agent-
infor re__agent-name")
    if agent_infor:
        name_tag = (
                agent_infor.find("a", class_="re__contact-
name") or
                agent_infor.find("a", class_="js__agent-
contact-name")
        )
        if name_tag:
            agent_info['name'] = name_tag.get_text(strip=True)
            href = name_tag.get('href')
            if href:
                agent_info['profile_url'] = href
    avatar_tag = soup.select_one("img.re__contact-avatar")
    if avatar_tag:
        src = avatar_tag.get('src')
        if src:
            agent_info['avatar_url'] = src
    phone_tag = soup.select_one("div.js__phone")
    if phone_tag:
        phone_span = phone_tag.find('span')
        if phone_span:
            agent_info['phone_invisible'] =
phone_span.get_text(strip=True)
    zalo_tag = soup.select_one("a.js__zalo-chat")
    if zalo_tag:
        data_href = zalo_tag.get('data-href')
        if data_href:
            agent_info['zalo_url'] = data_href
    other_agent_info = soup.select("div.re__agent-experiment
div.agent-deail-infor")
    for item in other_agent_info:
        label_tag = item.select_one("span")
        value_tag = item.select_one("i")
        if label_tag and value_tag:
            label = label_tag.get_text(strip=True).lower()
            value = value_tag.get_text(strip=True)
            if "tham gia" in label:
                agent_info["join_duration"] = value
            elif "tin đăng" in label:
                agent_info["listings"] = value
    return agent_info
```

```
def get_project_info(soup):
    project_info = {}
    card = soup.select_one("div.re__ldp-project-info")
    if not card:
        return project_info
    title = get_text(card, "div.re__project-title")
    if title:
        project_info["name"] = title
    for item in card.select("span.re__prj-card-config-value"):
        text = item.get_text(strip=True)
        aria_label = unidecode.unidecode(item.get("aria-
label", "")).lower()
        if "trang thai" in aria_label:
            project_info["status"] = text
        elif "gia" in aria_label:
            project_info["price"] = text
    investor = get_text(card, "span.re__prj-card-config-value
i.re__icon-office--sm + span.re__long-text")
    if investor:
        project_info["investor"] = investor
    img = card.select_one("div.re__section-avatar img")
    if img:
        src = img.get("src")
        if src:
            project_info["image"] = src
    link = card.select_one("div.re__section-avatar a")
    if link:
        href = link.get("href")
        if href:
            project_info["project_url"] = href
    a_tag = card.select_one("a.re__link-pr span")
    if a_tag:
        text = a_tag.get_text(strip=True)
        match = re.search(r'\d+', text.replace(',', ''))
        if match:
            project_info["listing_count"] = int(match.group())
    return project_info


def get_description(soup):
    prefix = "Thông tin mô tả"
    description_tag = soup.select_one(".re__pr-description")
    description = description_tag.get_text(strip=True)
    if description.startswith(prefix):
        description = description[len(prefix):].strip()
    return description if description else None


def get_images(soup):
    images = []
    for item in soup.select("div.re__media-thumb-
item.js__media-thumbs-item"):
        img_tag = item.find("img")
        if img_tag:
            img_url = img_tag.get("data-src") or
img_tag.get("src")
            if img_url:
                images.append(img_url)
    return images
```

Phân phối giá của hai thành phố Hà Nội và Tp. Hồ Chí Minh

```
df_sale['province_clean'] = df_sale['province'].str.replace('Thành phố ', '').str.replace('TP. ', '').str.strip()

def plot_district_prices_fixed(city_name, ax, color_palette):
    city_df = df_sale[df_sale['province_clean'].str.contains(city_name, case=False, na=False)].copy()

    if city_df.empty:
        ax.set_title(f"Không có dữ liệu cho {city_name}")
        return
    district_counts = city_df['district'].value_counts()
    top_districts = district_counts[district_counts > 10].index
    city_df = city_df[city_df['district'].isin(top_districts)]

    order = city_df.groupby('district')['price_per_spm'].median().sort_values(ascending=False).index

    sns.boxplot(
        data=city_df,
        y='district',
        x='price_per_spm',
        order=order,
        ax=ax,
        palette=color_palette,
        fliersize=1
    )

    ax.set_xlim(0, city_df['price_per_spm'].quantile(0.95))
    ax.set_title(f'PHÂN PHỐI GIÁ TẠI {city_name.upper()}', fontsize=16, fontweight='bold', pad=15)
    ax.set_xlabel('Giá (Triệu VNĐ/m²)', fontsize=12)
    ax.set_ylabel('', fontsize=12)
    ax.grid(axis='x', linestyle='--', alpha=0.6)
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(14, 20))

plot_district_prices_fixed('Hà Nội', ax1, 'Reds')
plot_district_prices_fixed('Hồ Chí Minh', ax2, 'Blues')

plt.tight_layout(pad=4.0)
plt.show()
```

Phân phối giá theo hướng nhà

```
import plotly.io as pio
pio.renderers.default = "browser"
import plotly.express as px

orientation_counts = df_sale['spec.orientation'].value_counts().reset_index()
orientation_counts.columns = ['Hướng', 'Số lượng']

orientation_price = df_sale.groupby('spec.orientation')['price_per_spm'].median().reset_index()
orientation_price.columns = ['Hướng', 'Giá trung vị/m²']

orientation_df = orientation_counts.merge(orientation_price, on='Hướng')

fig = px.bar_polar(orientation_df, r='Số lượng', theta='Hướng',
                   color='Giá trung vị/m²', template="plotly_white",
                   color_continuous_scale=px.colors.sequential.Reds,
                   title='PHÂN BỔ HƯỚNG NHÀ & MẶT BẰNG GIÁ (Màu đậm = Giá cao)')
fig.show()
```

Phân phối giá thuê và giá bán

```
def plot_dist_with_boxplot(data, title, color, unit):
    f, (ax_box, ax_hist) = plt.subplots(2, sharex=True,
                                        gridspec_kw={"height_ratios": (.15, .85)},
                                        figsize=(12, 7))

    sns.boxplot(x=data, ax=ax_box, color=color, fliersize=4)
    ax_box.set(xlabel='')
    ax_box.set_title(title, fontsize=16, fontweight='bold', pad=20)

    sns.histplot(x=data, ax=ax_hist, kde=True, color=color, bins=50)
    ax_hist.set_xlabel(f'Giá ({unit})', fontsize=12)
    ax_hist.set_ylabel('Số lượng bài đăng', fontsize=12)
    sns.despine(ax=ax_hist)
    sns.despine(ax=ax_box, left=True)
    plt.tight_layout()
    plt.show()

rent_prices = df_rent[df_rent['price'] < 50]['price'].dropna()
plot_dist_with_boxplot(rent_prices, 'PHÂN PHỐI GIÁ THUÊ (Dưới 50 triệu)', '#3498db', 'Triệu
VNĐ/tháng')

sale_prices = df_sale[df_sale['price'] < 20000]['price'].dropna()
plot_dist_with_boxplot(sale_prices, 'PHÂN PHỐI GIÁ BÁN (Dưới 20 tỷ)', '#e74c3c', 'Triệu
VNĐ')
```

Lấy top 10 dự án có nhiều tin đăng nhất

```
top_projects = df_sale['project_info.name'].value_counts().head(11)
if 'None' in top_projects or None in top_projects:
    top_projects = top_projects.drop(labels=[None], errors='ignore')
top_projects = top_projects.head(10)

project_prices =
df_sale[df_sale['project_info.name'].isin(top_projects.index)].groupby('project_info.name')['price_
per_spm'].mean().sort_values(ascending=False)

fig, ax1 = plt.subplots(figsize=(14, 7))
sns.barplot(x=top_projects.index, y=top_projects.values, ax=ax1, alpha=0.6, color='blue', label='Số
lượng tin')
ax1.set_ylabel('Số lượng tin đăng', fontsize=12, fontweight='bold')
ax1.tick_params(axis='x', rotation=45)
ax2 = ax1.twinx()
sns.lineplot(x=project_prices.index, y=project_prices.values, ax=ax2, marker='o', color='red',
linewidth=2, label='Giá trung bình/m2')
ax2.set_ylabel('Giá TB (Triệu/m²)', fontsize=12, fontweight='bold', color='red')
plt.title('TOP 10 DỰ ÁN NỔI BẬT: SỐ LƯỢNG TIN VS GIÁ TRUNG BÌNH', fontsize=16, fontweight='bold')
plt.show()
```

Lấy Top 10 đặc trưng ảnh hưởng đến 'price'

```
features_to_check = [
    'price', 'area', 'price_per_spm', 'spec.front_width',
    'spec.road_width', 'spec.num_floor', 'spec.bedroom',
    'spec.bathroom', 'contact_info.listings', 'project_info.listing_count', 'verified_num'
]

corr_matrix = df_sale[features_to_check].dropna(subset=['price']).corr()
price_corr = corr_matrix['price'].drop('price', errors='ignore')
top_10_features = price_corr.abs().sort_values(ascending=False).head(10)
top_10_data = price_corr[top_10_features.index] # Lấy giá trị gốc để biết âm hay dương

plt.figure(figsize=(12, 8))
colors = ['#2ecc71' if x > 0 else '#e74c3c' for x in top_10_data.values]
sns.barplot(x=top_10_data.values, y=top_10_data.index, palette=colors)
plt.axvline(x=0, color='black', linestyle='-', alpha=0.3)
plt.title('TOP 10 ĐẶC TRƯNG TÁC ĐỘNG MẠNH NHẤT ĐẾN TỔNG GIÁ', fontsize=16,
fontweight='bold', pad=20)
plt.xlabel('Hệ số tương quan', fontsize=12)
plt.ylabel('Đặc trưng', fontsize=12)

for i, v in enumerate(top_10_data.values):
    plt.text(v, i, f' {v:.2f}', va='center', fontweight='bold', color='black')

plt.tight_layout()
plt.show()
```

Bản đồ nhiệt phân bố tin đăng bất động sản tại Việt Nam

```
import folium
from folium.plugins import HeatMap

def clean_geo_data(df_input):
    temp_df = df_input.copy()
    temp_df['latitude'] = pd.to_numeric(temp_df['latitude'],
errors='coerce')
    temp_df['longitude'] = pd.to_numeric(temp_df['longitude'],
errors='coerce')
    return temp_df.dropna(subset=['latitude', 'longitude'])

df_geo = clean_geo_data(df_sale)
google_maps_vi_url =
'https://mt1.google.com/vt/lyrs=m&hl=vi&x={x}&y={y}&z={z}'
center_lat = df_geo['latitude'].mean()
center_lon = df_geo['longitude'].mean()
m_final = folium.Map(
    location=[center_lat, center_lon], zoom_start=11,
    tiles=google_maps_vi_url,
    attr='Dữ liệu © Google Maps Việt Nam'
)
heat_data = [[row['latitude'], row['longitude']] for index,
row in df_geo.iterrows()]

HeatMap(
    heat_data, radius=15, blur=10, min_opacity=0.5,
     name='Mật độ tin đăng'
).add_to(m_final)

folium.Marker(
    location=[16.565141, 112.641903],
    popup='Quần đảo Hoàng Sa (Việt Nam)',
    icon=folium.Icon(color='red', icon='flag')
).add_to(m_final)

folium.Marker(
    location=[10.725116, 115.842338],
    popup='Quần đảo Trường Sa (Việt Nam)',
    icon=folium.Icon(color='red', icon='flag')
).add_to(m_final)

folium.LayerControl().add_to(m_final)
m_final.save('heatmap_bat_dong_san.html')
```

PHỤ LỤC 2

Commits on Dec 27, 2025

Merge pull request #9 from thanglemvp-max/main ptphat810 authored 7 hours ago	Verified	7338cad
Thang add 15 query Le Dinh Nhat Thang committed 7 hours ago		c54bfe0
Thang Add OSDS progress report Le Dinh Nhat Thang committed 7 hours ago		41394fc
Merge pull request #8 from nickeydinh/main ptphat810 authored 17 hours ago	Verified	4263c1c
update requirements nickeydinh committed 18 hours ago		979a83d
Merge branch 'main' of https://github.com/nickeydinh/vn-real-estate-scraper nickeydinh committed 18 hours ago		4f27db4
visual chart nickeydinh committed 18 hours ago		b11dcf6
update EDA nickeydinh committed 18 hours ago		49d74ad
update EDA nickeydinh committed 19 hours ago		819348d
cập nhật file requirements cho phần trực quan nickeydinh committed 19 hours ago		7857197
update path for modules ptphat810 committed yesterday		f3a7439

Commits on Dec 26, 2025

Merge branch 'main' of https://github.com/nickeydinh/vn-real-estate-scraper nickeydinh committed yesterday		e1174f8
create EDA nickeydinh committed yesterday		f303cdd
update code ptphat810 committed 2 days ago		e73e82a
remove sqlite and update push data mongoatlas ptphat810 committed 2 days ago		2081b3c
migrate data crawl to cloud ptphat810 authored 2 days ago	Verified	ddea54e
delete the scraping_status.db archive file and replace it with mongodb atlas ptphat810 authored 2 days ago	Verified	2977106
update 'verified' field boolean to string ptphat810 committed 2 days ago		d1b8e3f

Commits on Dec 25, 2025

Merge pull request #7 from nickeydinh/main ptphat810 authored 2 days ago	Verified	1fd403c
Merge branch 'ptphat810:main' into main nickeydinh authored 2 days ago	Verified	100e7e0
updata file cleaner nickeydinh committed 2 days ago		8de9be4
Merge pull request #6 from nickeydinh/main ptphat810 authored 2 days ago	Verified	527259a
add data for rent bds nickeydinh committed 2 days ago		4a71e3f
update data to drive ptphat810 committed 3 days ago		ea7c4ad

Commits on Dec 24, 2025

upadate README.md ptphat810 committed 4 days ago		47ac33e
create connect mongodb ptphat810 committed 4 days ago		b3075e4
add new records ptphat810 committed 4 days ago		2e84085
update code ptphat810 committed 4 days ago		81ae500

Commits on Dec 22, 2025

fix error result return NoneType ptphat810 committed last week		4b75eea
add project info field and final pattern for schema json ptphat810 committed last week		8fa4f6d
reupload data ptphat810 committed last week		ad1af60
Delete data/raw directory ptphat810 authored last week	Verified	4a4d026
delete wron format ptphat810 committed last week		c1da982
upadate new records ptphat810 committed last week		6803a35

Commits on Dec 21, 2025

Rename eda.ipynb to processing.ipynb ptphat810 authored last week	Verified	ba57346

Commits on Dec 21, 2025

update and add new records ptphat810 committed last week		0548823
add get project info tab and update contact agent info ptphat810 committed last week		ae314d4
update list ptphat810 committed last week		aa9c00c
add configuration file ptphat810 committed last week		7b5282e
Merge pull request #4 from thanglemvp-max/main ptphat810 authored last week	Verified	9fa9466
refactor code and add spec for rent property ptphat810 committed last week		a2237e6

Commits on Dec 20, 2025

Thang Fixed data_cleaner file Le Dinh Nhat Thang committed last week		b42bd36
update scripts run code ptphat810 committed last week		80d2224
update insert data to mongodb and add notebooks for eda ptphat810 committed last week		d8833aa
Delete src/test.py ptphat810 authored last week	Verified	c91d170
add logging module ptphat810 committed last week		3e5c654
add crawler engine, downloader and storage modules ptphat810 committed last week		44f04e0
add new records ptphat810 committed last week		b9c235a

Commits on Dec 19, 2025

Merge pull request #3 from nickeydinh/main ptphat810 authored last week	Verified	4210470
update content for README.md ptphat810 committed last week		5ef8ab0
add mongo client for database connection ptphat810 committed last week		07ccfe8
add project dependencies ptphat810 committed last week		5559514
Update the scraper file to get a wider variety of links. nickeydinh committed last week		0a49fda

Commits on Dec 18, 2025

add scraped data ptphat810 committed last week		aeec4f4
Merge pull request #2 from thanglemvp-max/main ptphat810 authored last week	Verified	647e369
Thang add data cleaning script Le Dinh Nhat Thang committed last week		35325d7

Commits on Dec 17, 2025

run code in source ptphat810 committed 2 weeks ago		aae3b08
add duplicate_filter.py to remove duplicate records ptphat810 committed 2 weeks ago		f544f39
run code scraper.py and clener.py ptphat810 committed 2 weeks ago		7c2fcf7
update extract_agent_info function ptphat810 committed 2 weeks ago		71710b1
Merge pull request #1 from nickeydinh/main ptphat810 authored 2 weeks ago	Verified	a1610fa
add function to extract agent contact information ptphat810 committed 2 weeks ago		a913b3c
Add a function to get real estate URLs nickeydinh committed 2 weeks ago		06a8648
Updated Thanhvien 2 nickeydinh committed 2 weeks ago		f4e5adc
add extract coordinate from js ptphat810 committed 2 weeks ago		bddd87e
add property detail parser ptphat810 committed 2 weeks ago		ef7f2de
Add some data fields ptphat810 committed 2 weeks ago		c726a5a
update .gitignore ptphat810 committed 2 weeks ago		ca9fc81

Commits on Dec 16, 2025

Add initial parser ptphat810 committed 2 weeks ago		af314df
Updated Thanhvien 1 ptphat810 committed 2 weeks ago		85d12d8

Commits on Dec 16, 2025

Added Thanhvien.txt ptphat810 committed 2 weeks ago		7a80177
Initial commit ptphat810 authored 2 weeks ago	Verified	221c196