NHẬP MÔN KHOA HỌC DỮ LIỆU

NHÓM 1 - LỚP N03

BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÁO CÁO DỰ ÁN
NHẬP MÔN KHOA HỌC DỮ LIỆU

Đề tài: **" PHÂN TÍCH VÀ DỰ ĐOÁN GIÁ THUÊ PHÒNG TRỌ Ở TP HCM"**

Nhóm 01 : NO3
Người hướng dẫn : Đinh Xuân Trường
Trần Hoàng Hiệp : B21DCCN346
Chu Văn Mạnh : B21DCCN510
Trần Việt Hoàng : B21DCCN057
Trần Vũ Cảnh : B21DCCN175
Nguyễn Hải Nam : B19DCCN451

HÀ NỘI - 2024

BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÁO CÁO DỰ ÁN
NHẬP MÔN KHOA HỌC DỮ LIỆU

Đề tài: **" PHÂN TÍCH VÀ DỰ ĐOÁN GIÁ THUÊ PHÒNG TRỌ Ở TP HCM"**

Nhóm 01	**: NO3**
Trần Hoàng Hiệp	**: B21DCCN346**
Chu Văn Mạnh	**: B21DCCN510**
Trần Việt Hoàng	**: B21DCCN057**
Trần Vũ Cảnh	**: B21DCCN175**
Nguyễn Hải Nam	**: B19DCCN451**

LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin được gửi lời cảm ơn chân thành nhất đến thầy Đinh Xuân Trường. Trong quá trình học tập và tìm hiểu môn Nhập môn Khoa học dữ liệu em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hướng dẫn tâm huyết và tận tình của thầy. Thầy đã giúp em tích lũy thêm nhiều kiến thức về kỹ năng quan trọng này để có thể hoàn thành được bài tiểu luận về đề tài: PHÂN TÍCH VÀ DỰ ĐOÁN GIÁ THUÊ PHÒNG TRỌ Ở TP HCM.

Trong quá trình làm bài chắc chắn khó tránh khỏi những thiếu sót. Do đó, em kính mong nhận được những lời góp ý của thầy để bài tiểu luận của em ngày càng hoàn thiện hơn. Em xin chân thành cảm ơn!

Hà Nội, ngày tháng năm 20...

Đại diện nhóm

Chu Văn Mạnh

MỤC LỤC

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API: Application Programming Interface	Giao diện lập trình ứng dụng
DT: Decision TreeC	Cây quyết định
MAE: Mean Absolute Error	Sai số trung bình tuyệt đối
MSE: Mean Squared Error	Sai số bình phương trung bình
RFM: Random Forest Model	Mô hình Rừng ngẫu nhiên
RMSE: Root Mean Squared Error	Căn bậc hai của sai số bình phương trung bình
SF: Square Footage	Diện tích theo bộ vuông

DANH MỤC HÌNH VẼ

PHẦN MỞ ĐẦU

Mục tiêu và định hướng cá nhân về quá trình KHDL: Mục tiêu bản thân trong quá trình thực tập cơ sở là củng cố kiến thức cơ sở cho sinh viên trước khi vào phần chuyên ngành, tạo tiền đề cho các sinh viên trong kỳ thực tập năm tiếp theo. Tôi mong muốn phát triển kỹ năng phân tích dữ liệu và ứng dụng công nghệ thông tin vào thực tiễn, từ đó xây dựng nền tảng vững chắc cho sự nghiệp sau này. Lý do lựa chọn đề tài này là vì nhu cầu ngày càng cao về thông tin thị trường bất động sản, đặc biệt là giá thuê phòng trọ trong bối cảnh đô thị hóa nhanh chóng tại TPHCM.

Trình bày phần đặt vấn đề liên quan đến đề tài của KHDL: Hiện nay, Thành phố Hồ Chí Minh đang đối mặt với nhiều thách thức trong việc quản lý giá thuê nhà ở, đặc biệt là phòng trọ. Với sự gia tăng dân số và nhu cầu nhà ở ngày càng cao, việc phân tích và dự đoán giá thuê phòng trọ trở thành một vấn đề cấp thiết. Công nghệ phân tích dữ liệu và học máy hiện nay đã mở ra nhiều cơ hội để giải quyết các vấn đề này, giúp người dân và các nhà đầu tư có cái nhìn tổng quan hơn về thị trường.

Trình bày phần các giải pháp hiện tại và hạn chế: Trong thực tế, thị trường phòng trọ tại TPHCM đang gặp nhiều khó khăn, bao gồm việc thiếu thông tin chính xác về giá cả và xu hướng thị trường. Các giải pháp hiện tại chủ yếu dựa vào dữ liệu thống kê từ các cơ quan nhà nước và trang web bất động sản, nhưng thường không được cập nhật thường xuyên và thiếu tính chính xác. Điều này dẫn đến việc người thuê và chủ nhà không thể đưa ra quyết định hợp lý, ảnh hưởng đến trải nghiệm thuê nhà.

Trình bày phần mục tiêu và hướng giải pháp: Mục tiêu của đề tài là phát triển các mô hình phân tích và dự đoán giá thuê phòng trọ dựa trên các yếu tố như vị trí, diện tích, tiện nghi và xu hướng thị trường. Hướng giải pháp sẽ bao gồm việc thu thập dữ liệu từ nhiều nguồn khác nhau, áp dụng các thuật toán học máy để cải thiện độ chính xác trong dự đoán, và cung cấp thông tin hữu ích cho người thuê và chủ nhà.

Trình bày phần đóng góp Đề tài: Hướng tới nhu cầu đó, đề tài có tên "***Phân tích và dự đoán giá thuê phòng trọ ở Thành phố Hồ Chí Minh***".

Nội dung trình bày trong báo cáo gồm 3 chương chính:

- Chương 1: Đặt vấn đề
- Chương 2: Báo cáo tiến độ
- Chương 3: Kết luận

CHƯƠNG 1. ĐẶT VẤN ĐỀ

*C*hương 1 giới thiệu về đề tài "Phân tích và dự đoán giá thuê phòng trọ ở thành phố Hồ Chí Minh"

1.1 Tổng quan đề tài

Khảo sát hiện trạng

Thành phố Hồ Chí Minh (TPHCM) là một trong những đô thị lớn nhất và phát triển nhanh nhất tại Việt Nam. Với sự gia tăng dân số và nhu cầu về nhà ở ngày càng cao, việc phân tích và dự đoán giá thuê phòng trọ trở thành một vấn đề cấp thiết.

- Nhu cầu thực tế: Nhiều người, đặc biệt là sinh viên và người lao động từ các tỉnh khác, tìm kiếm chỗ ở phù hợp với ngân sách của mình. Việc hiểu rõ về giá cả sẽ giúp họ đưa ra quyết định đúng đắn.
- Hỗ trợ người cho thuê: Các chủ trọ cũng cần thông tin về giá cả thị trường để định giá hợp lý, từ đó tối ưu hóa lợi nhuận.
- Hỗ trợ người dân: Đề tài này không chỉ mang lại thông tin hữu ích cho người thuê mà còn giúp các chủ nhà điều chỉnh giá cả một cách hợp lý.
- Cung cấp dữ liệu cho nhà hoạch định chính sách: Các cơ quan chức năng có thể dựa vào kết quả nghiên cứu để hoạch định chính sách phát triển nhà ở tại TPHCM.
- Góp phần vào nghiên cứu thị trường: Đề tài cũng mở ra hướng nghiên cứu mới trong lĩnh vực bất động sản, giúp các nhà đầu tư có cái nhìn tổng quan về thị trường.

Các giải pháp hiện tại và hạn chế

Các giải pháp hiện tại:

- Sử dụng dữ liệu thống kê: Các cơ quan nhà nước và tổ chức nghiên cứu thường thu thập và công bố dữ liệu giá thuê phòng trọ theo khu vực. Dữ liệu này cung cấp thông tin hữu ích cho người dân và các nhà đầu tư.
- Các trang web tìm kiếm phòng trọ: Nhiều nền tảng trực tuyến cung cấp thông tin về giá cả và thông tin chi tiết về các căn phòng cho thuê, giúp người thuê dễ dàng tìm kiếm và so sánh.
- Mô hình dự đoán đơn giản: Một số nghiên cứu đã áp dụng các mô hình hồi quy tuyến tính hoặc hồi quy logistic để dự đoán giá thuê dựa trên các yếu tố như diện tích, vị trí và tiện nghi.

Hạn chế:

- Thiếu dữ liệu chính xác: Dữ liệu thu thập được có thể không đầy đủ hoặc không được cập nhật thường xuyên, dẫn đến việc phân tích không chính xác.
- Chưa áp dụng công nghệ tiên tiến: Nhiều mô hình hiện tại chưa sử dụng các thuật toán học máy tiên tiến, như hồi quy phi tuyến, mạng nơ-ron hoặc cây quyết định, làm giảm khả năng dự đoán chính xác.
- Thiếu sự đa dạng trong các yếu tố phân tích: Nhiều nghiên cứu chỉ xem xét một số yếu tố nhất định mà không tính đến các yếu tố khác có thể ảnh hưởng đến giá thuê, như tình hình kinh tế, sự thay đổi trong nhu cầu nhà ở, hoặc các yếu tố xã hội.

- Khó khăn trong việc phân tích vùng miền: Thành phố Hồ Chí Minh có nhiều khu vực với đặc điểm khác nhau, việc áp dụng một mô hình chung cho tất cả các khu vực có thể không phản ánh đúng thực tế.

1.2 Mục tiêu và định hướng giải pháp

Mục tiêu:

- Phân Tích Tình Hình Thực Tế: Đánh giá hiện trạng giá thuê phòng trọ tại Thành phố Hồ Chí Minh, xác định các yếu tố ảnh hưởng đến giá cả.
- Dự Đoán Giá Thuê: Phát triển các mô hình dự đoán chính xác giá thuê phòng trọ dựa trên các yếu tố như vị trí, diện tích, tiện nghi, và xu hướng thị trường.
- Cung Cấp Thông Tin Hữu Ích: Cung cấp thông tin chi tiết cho người thuê và chủ nhà về giá cả thị trường, giúp họ đưa ra quyết định hợp lý.
- Hỗ Trợ Quyết Định Chính Sách: Cung cấp dữ liệu và phân tích cho các cơ quan chức năng nhằm hỗ trợ hoạch định chính sách phát triển nhà ở.

Định hướng giải pháp:

- Thu Thập Dữ Liệu: Tăng cường thu thập dữ liệu từ nhiều nguồn khác nhau, bao gồm các trang web bất động sản, khảo sát người thuê, và báo cáo của các cơ quan chức năng.
- Sử Dụng Công Nghệ Tiên Tiến: Áp dụng các thuật toán học máy hiện đại như hồi quy phi tuyến, mạng nơ-ron, và cây quyết định để cải thiện độ chính xác của mô hình dự đoán.
- Tích Hợp Nhiều Yếu Tố Phân Tích: Mở rộng phạm vi phân tích bằng cách xem xét không chỉ các yếu tố vật lý của căn phòng mà còn các yếu tố xã hội và kinh tế có thể ảnh hưởng đến giá thuê.
- Phát Triển Công Cụ Trực Tuyến: Xây dựng một công cụ trực tuyến cho phép người dùng dễ dàng tra cứu và so sánh giá thuê phòng trọ dựa trên các yếu tố mà họ quan tâm.
- Đánh Giá và Cập Nhật Mô Hình: Định kỳ đánh giá và cập nhật các mô hình dự đoán dựa trên dữ liệu mới và phản hồi từ người dùng để đảm bảo tính chính xác và khả thi.

1.3 Đóng góp của đề tài

Đề tài phân tích và dự đoán giá thuê phòng trọ tại Thành phố Hồ Chí Minh có những đóng góp quan trọng sau:

- **Cung Cấp Thông Tin Chất Lượng**: Đề tài giúp cung cấp thông tin chính xác và kịp thời về giá thuê phòng trọ, hỗ trợ người thuê và chủ nhà trong việc ra quyết định.
- **Phát Triển Các Mô Hình Dự Đoán**: Việc phát triển các mô hình dự đoán dựa trên dữ liệu thực tế không chỉ giúp cải thiện độ chính xác mà còn mở ra hướng nghiên cứu mới trong lĩnh vực bất động sản.
- **Hỗ Trợ Chính Sách Nhà Nước**: Dữ liệu và phân tích từ đề tài có thể được sử dụng để hỗ trợ các cơ quan chức năng trong việc hoạch định chính sách phát triển nhà ở, từ đó cải thiện tình hình cung cầu nhà ở tại TPHCM.
- **Nâng Cao Nhận Thức Thị Trường**: Đề tài giúp nâng cao nhận thức của người dân về tình hình thị trường bất động sản, từ đó góp phần vào việc điều chỉnh hành vi tiêu dùng và đầu tư.

- **Khuyến Khích Nghiên Cứu Tiếp Theo**: Kết quả nghiên cứu có thể là cơ sở cho các nghiên cứu tiếp theo trong lĩnh vực phân tích thị trường bất động sản, đặc biệt là trong việc áp dụng các công nghệ tiên tiến hơn.

Kết luận chương 1

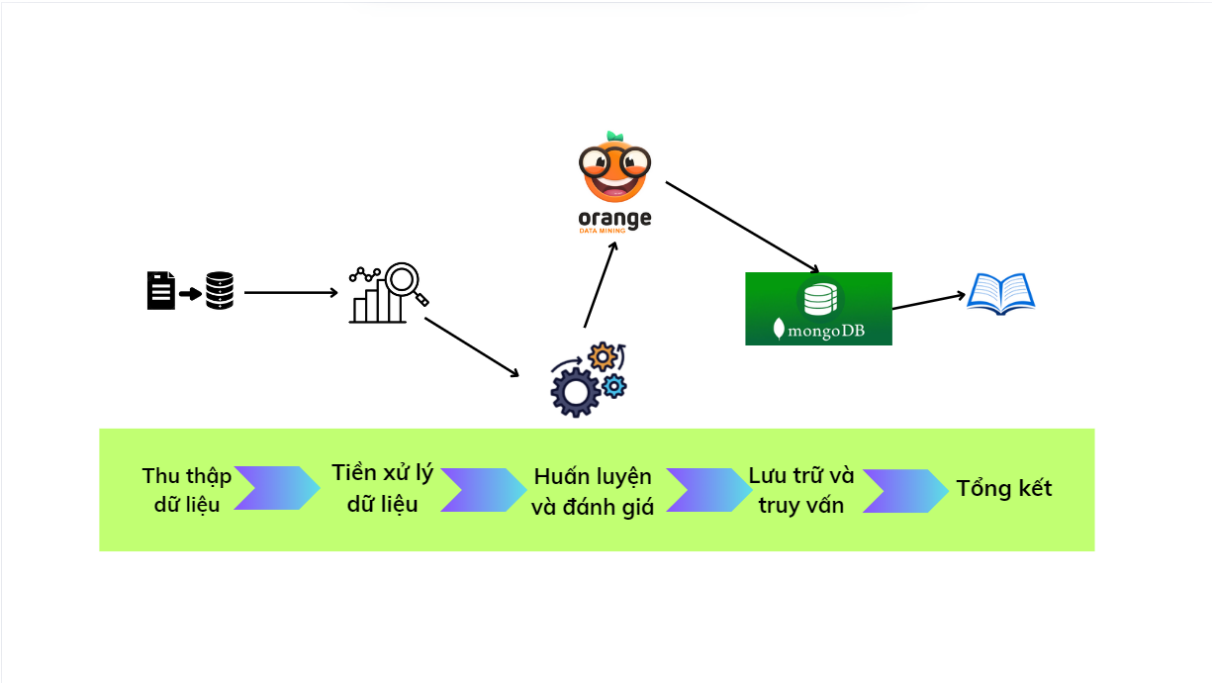
Chương giới thiệu đề tài phân tích và dự đoán giá thuê phòng trọ tại Thành phố Hồ Chí Minh đã nêu rõ lý do và tầm quan trọng của việc nghiên cứu lĩnh vực này. Với sự phát triển nhanh chóng của thành phố và nhu cầu nhà ở ngày càng tăng, việc hiểu rõ về giá thuê phòng trọ không chỉ giúp người dân đưa ra quyết định hợp lý mà còn hỗ trợ chủ nhà trong việc định giá bất động sản.

Đề tài đã xác định mục tiêu và hướng giải pháp cụ thể nhằm cải thiện độ chính xác của các mô hình dự đoán, đồng thời cung cấp thông tin hữu ích cho người thuê và cơ quan chức năng. Các giải pháp hiện tại và hạn chế cũng đã được phân tích, từ đó mở ra hướng nghiên cứu mới cho tương lai.

Những đóng góp của đề tài không chỉ có giá trị trong việc nghiên cứu lý thuyết mà còn có ý nghĩa thực tiễn lớn, hỗ trợ cho việc hoạch định chính sách phát triển nhà ở và nâng cao nhận thức của cộng đồng về thị trường bất động sản. Qua đó, đề tài hy vọng sẽ là nguồn tài liệu tham khảo quý giá cho các nghiên cứu tiếp theo trong lĩnh vực này.

CHƯƠNG 2. BÁO CÁO TIẾN ĐỘ TỪNG TUẦN

2.1 Sơ đồ project



Hình 2.1: Sơ đồ Project

2.2 Tổng quan quá trình làm dự án trong 10 tuần

Tiến độ thực hiện bài tập lớn thể hiện dưới đây:

Tuần Topic 1 (06/09 - 13/09)

Chủ đề tìm hiểu tuần topic 1: Thu thập và chuẩn bị dữ liệu.

Lý do và động lực chọn đề tài:

1. Lý do: Việc thu thập dữ liệu từ trang web thuephongtro.com, một trang web phổ biến tại Việt Nam chuyên cung cấp thông tin về các phòng trọ cho thuê
2. Động lực: Nhu cầu tìm kiếm phòng trọ tại các thành phố lớn như Hồ Chí Minh đang ngày càng tăng cao. Việc thu thập và phân tích dữ liệu từ trang web này sẽ giúp chúng tôi hiểu rõ hơn về thị trường cho thuê phòng trọ, các yếu tố ảnh hưởng đến giá thuê, và các khu vực phổ biến được người thuê ưa chuộng.

Công nghệ thực hiện: Trang web thuephongtro.com cung cấp thông tin về các phòng trọ cho thuê tại Hồ Chí Minh. Để phân tích và sử dụng dữ liệu này cho các mục đích phân tích, việc thu thập dữ liệu tự động từ trang web là cần thiết. Scrapy là một framework mạnh mẽ và linh hoạt để thực hiện nhiệm vụ này.

Loại website: Loại website: thuộc kiểu động, có dùng javascript, không cần đăng nhập, không cần captcha.

Lý do sử dụng Scrapy:

1. Hiệu suất cao: Scrapy được thiết kế để thu thập dữ liệu một cách hiệu quả và nhanh chóng. Nó có khả năng xử lý nhiều yêu cầu đồng thời, giúp giảm thời gian thu thập dữ liệu từ các trang web lớn.

2. Dễ sử dụng và mở rộng: Scrapy cung cấp một API đơn giản và dễ sử dụng để viết các spider. Nó cũng hỗ trợ mở rộng dễ dàng với các middleware và pipeline, cho phép tùy chỉnh quá trình thu thập và xử lý dữ liệu.
3. Hỗ trợ XPath và CSS Selectors: Scrapy cho phép sử dụng cả XPath và CSS selectors để trích xuất dữ liệu từ các trang web. Điều này giúp việc trích xuất dữ liệu trở nên linh hoạt và dễ dàng hơn.
4. Quản lý phiên làm việc và xử lý lỗi: Scrapy tích hợp sẵn các tính năng quản lý phiên làm việc và xử lý lỗi, giúp đảm bảo quá trình thu thập dữ liệu diễn ra suôn sẻ và không bị gián đoạn.
5. Hỗ trợ nhiều định dạng dữ liệu: Scrapy hỗ trợ xuất dữ liệu ra nhiều định dạng khác nhau như JSON, CSV, XML, giúp dễ dàng tích hợp với các hệ thống khác.

Kỹ thuật sử dụng:

1. Crawl Spiders: Scrapy hỗ trợ việc tạo ra các Spider (nhện) để điều hướng qua các trang web và thu thập dữ liệu. Bằng cách xác định các quy tắc định hình cụ thể, Spider có thể tự động thu thập dữ liệu từ nhiều trang web.
2. Middleware: Scrapy hỗ trợ Middleware cho phép xử lý các yêu cầu và phản hồi trước hoặc sau khi chúng được xử lý bởi Spider. Điều này cho phép thực hiện các bước xử lý như đổi User-Agent, xử lý Cookies, hoặc xử lý các yêu cầu bị chặn.
3. Pipeline: Scrapy sử dụng Pipeline để xử lý dữ liệu sau khi nó được cào từ trang web. Pipeline cho phép bạn lọc, xử lý, và lưu trữ dữ liệu theo cách tùy chỉnh.

Kết quả:

1. Tiến hành trích xuất dữ liệu các trường bao gồm:
    - **name**: Tiêu đề của bài viết.
    - **DistrictAndCity**: Quận và thành phố.
    - **Date**: Ngày đăng bài.
    - **price**: Giá thuê.
    - **square**: Diện tích.
    - **Id**: ID của bài viết.

Sau đó, tiến hành sử dụng XPath chung đã trích được là "/html/body/div/main/div/div[3]/div/div[2]/article[’+str(i)+’]/a/div"kết hợp CSS selectors để trích xuất các thông tin cần thiết từ mỗi mục,

Id	Title	Price	Square	District	City	Date
610	Cho thuê nhà trọ số 43/2A1 Tăng Nhơn Phú A, Qu...	Thỏa thuận	20 m²	Quận 9	Hồ Chí Minh	12/12/2014
1401	cho thuê phòng trọ	1 Triệu/tháng	8 m²	Quận 8	Hồ Chí Minh	30/03/2015
2819	Cho thuê phòng trọ 25m2 , 2.500.000	2.5 Triệu/tháng	25 m²	Tân Bình	Hồ Chí Minh	23/04/2016
2847	CHO THUÊ VĂN PHÒNG VÀ PHÒNG TRỌ GIÁ RẺ .ĐẸP .N...	1.5 Triệu/tháng	23 m²	Tân Phú	Hồ Chí Minh	23/07/2016
3049	Phòng cho thuê.Chính Chủ.Quận Bình Thạnh...	3 Triệu/tháng	25 m²	Bình Thạnh	Hồ Chí Minh	08/10/2017
...	...	...	...	...	...	...
169945	Phòng 18m2 máy lạnh,cửa sổ thoáng mát, giờ giấ...	2.8 Triệu/tháng	18 m²	Tân Bình	Hồ Chí Minh	11/09/2024
169946	🏠🏠🏠 Phòng trọ dạng Studio FULL NT cao cấp n...	6.5 Triệu/tháng	35 m²	Bình Thạnh	Hồ Chí Minh	15/09/2024
169949	Cho thuê phòng trọ riêng biệt 24m2 gần vòng xo...	2.2 Triệu/tháng	24 m²	Bình Tân	Hồ Chí Minh	17/09/2024
169950	Cho thuê phòng trọ Phường 3, Quận Gò Vấp	1.9 Triệu/tháng	15 m²	Gò Vấp	Hồ Chí Minh	17/09/2024
169955	Phòng 5 tr/tg. Dc 269 nguyễn thái bình bình, p...	5 Triệu/tháng	25 m²	Tân Bình	Hồ Chí Minh	20/09/2024

16000 rows × 6 columns

Hình 2.2: Dữ liệu tổng hợp được

-> Kết luận: Sau khi thực hiện các bước thu thập dữ liệu từ trang web *thuephongtro.com*, kết quả đạt được: (16000 dòng dữ liệu)

- **Thu thập dữ liệu thành công**: Thu thập được các thông tin cần thiết từ các bài viết trên trang web, bao gồm tiêu đề, quận và thành phố, ngày đăng bài, giá thuê, diện tích, và ID của bài viết. Dữ liệu này sẽ giúp phân tích thị trường cho thuê phòng trọ tại các thành phố lớn như Hồ Chí Minh.
- **Hiểu rõ hơn về thị trường**: Thông qua việc phân tích dữ liệu thu thập được có thể hiểu rõ hơn về các yếu tố ảnh hưởng đến giá thuê phòng trọ, các khu vực phổ biến được người thuê ưa chuộng, và xu hướng thị trường.
- **Công cụ hỗ trợ người dùng**: Dữ liệu thu thập được sẽ được sử dụng để phát triển các công cụ hỗ trợ người dùng trong việc tìm kiếm phòng trọ phù hợp với nhu cầu và ngân sách của họ. Điều này sẽ giúp người dùng tiết kiệm thời gian và công sức trong việc tìm kiếm phòng trọ.

Tên	%đóng góp	Ghi chú
Chu Văn Mạnh	30%	clean_data(code), báo cáo
Trần Việt Hoàng	30%	clean_data(code), báo cáo
Trần Hoàng Hiệp	30%	clean_data(code), báo cáo
Trần Vũ Cảnh	5%	báo cáo
Nguyễn Hải Nam	5%	báo cáo

Hình 2.3: Phần trăm đóng góp tuần 1

Tuần Topic 2 (13/09 - 20/09)

Chủ đề tìm hiểu tuần topic 2: Tiền xử lý dữ liệu.

Dữ liệu từ file inf:

Đọc dữ liệu từ file "inf.csv"và lưu kết quả vào DataFrame df-info.

```
df_info = pd.read_csv('./inf.csv')
df_info = df_info.set_index('Id').sort_index()
df_info
```

Id	Title	Price	Square	District	City	Date
610	Cho thuê nhà trọ số 43/2A1 Tăng Nhơn Phú A, Qu...	Thỏa thuận	20 m²	Quận 9	Hồ Chí Minh	12/12/2014
1401	cho thuê phòng trọ	1 Triệu/tháng	8 m²	Quận 8	Hồ Chí Minh	30/03/2015
2819	Cho thuê phòng trọ 25m2 , 2.500.000	2.5 Triệu/tháng	25 m²	Tân Bình	Hồ Chí Minh	23/04/2016
2847	CHO THUÊ VĂN PHÒNG VÀ PHÒNG TRỌ GIÁ RẺ .ĐẸP .N...	1.5 Triệu/tháng	23 m²	Tân Phú	Hồ Chí Minh	23/07/2016
3049	Phòng cho thuê.Chính Chủ.Quận Bình Thạnh...	3 Triệu/tháng	25 m²	Bình Thạnh	Hồ Chí Minh	08/10/2017
...	...	...	...	...	...	...
169945	Phòng 18m2 máy lạnh,cửa sổ thoáng mát, giờ giấ...	2.8 Triệu/tháng	18 m²	Tân Bình	Hồ Chí Minh	11/09/2024
169946	🏠🏠🏠 Phòng trọ dạng Studio FULL NT cao cấp n...	6.5 Triệu/tháng	35 m²	Bình Thạnh	Hồ Chí Minh	15/09/2024
169949	Cho thuê phòng trọ riêng biệt 24m2 gần vòng xo...	2.2 Triệu/tháng	24 m²	Bình Tân	Hồ Chí Minh	17/09/2024
169950	Cho thuê phòng trọ Phường 3, Quận Gò Vấp	1.9 Triệu/tháng	15 m²	Gò Vấp	Hồ Chí Minh	17/09/2024
169955	Phòng 5 tr/tg. Dc 269 nguyễn thái bình bình, p...	5 Triệu/tháng	25 m²	Tân Bình	Hồ Chí Minh	20/09/2024

16000 rows × 6 columns

Hình 2.4: Dữ liệu từ file inf

Xử lý dữ liệu bị lặp:

Kiểm tra xem trong tập có dòng nào bị trùng không. Nếu trùng thì trả về True, không thì là False

Xóa các dòng trùng lặp, giữ lại dòng gần nhất

```
num_duplicated_rows = df_info.duplicated().sum()
is_duplicated = (num_duplicated_rows != 0)
print('So dong lap la {}'.format(num_duplicated_rows))
print(is_duplicated)
```

```
So dong lap la 52
True
```

```
df_info = df_info.drop_duplicates()
num_duplicated_rows = df_info.duplicated().sum()
print('So dong lap la {}'.format(num_duplicated_rows))
```

```
So dong lap la 0
```

Hình 2.5: Kiểm tra và xóa dữ liệu trùng lặp

Ý nghĩa của mỗi cột:

- **ID** là để phân biệt phòng trọ, nhà trọ.

- **Title** là tiêu đề của phòng trọ, nhà trọ.

- **Price** là giá cho thuê của phòng trọ, nhà trọ theo tháng, đơn vị là Triệu/Tháng.

- **Square** là diện tích của phòng trọ, nhà trọ, đơn vị là m^2

- **District** là địa chỉ quận của phòng trọ, nhà trọ.

- **City** là địa chỉ thành phố của phòng trọ, nhà trọ.

- **Date** là ngày đăng tin cho thuê của phòng trọ, nhà trọ.

Kiểm tra dạng dữ liệu của các cột để xem có đúng với dạng mong muốn không.

```
df_info.dtypes
```

	0
Title	object
Price	object
Square	object
District	object
City	object
Date	object

dtype: object

Hình 2.6: Kiểm tra dữ liệu các cột

Các cột đều ở dạng object, điều này là không nên bởi vì nếu cột nào cũng ở dạng object thì rất khó tính toán.
Có 3 cột có thể chuyển sang dạng khác là Price (numeric), Square (numeric), Date (datetime) => tiến hành chuyển dạng của các cột này

- **Square**: loại bỏ đơn vị m^2 và chuyển sang dạng số

```
df_info["Square"] =df_info["Square"].str.replace('m²','')
df_info["Square"] = pd.to_numeric(df_info["Square"],errors='coerce')
df_info.rename(columns={'Square':'Square(m²)'},inplace=True)
df_info['Square(m²)']
```

Id	
610	20.0
1401	8.0
2819	25.0
2847	23.0
3049	25.0
...	...
169945	18.0
169946	35.0
169949	24.0
169950	15.0
169955	25.0

Hình 2.7: Xử lý dữ liệu cột square

Price: Xem cụ thể hơn các giá trị để có thể xử lý chính xác

Thấy có các dạng đơn vị là Triệu/tháng, Nghìn/tháng, Triệu/m2/tháng, Nghìn/m2/tháng, và có thêm một kiểu giá trị là Thỏa thuận.

Cần chuyển toàn bộ đơn vị sang Triệu/tháng và loại bỏ đơn vị ở sau để tiện chuyển thành số, riêng giá trị Thỏa thuận thì để là NaN

```
def convert_price(price,square):
    if 'Triệu/tháng' in price:
        price = price.replace('Triệu/tháng','')
        return float(price)
    if 'Trăm nghìn/tháng' in price:
        price = price.replace('Nghìn/tháng','')
        return float(price)/1000
    if 'Triệu/m2/tháng' in price:
        if pd.isna(square):
            return np.nan
        price = price.replace('Triệu/m2/tháng','')
        return float(price)*square
    if 'Nghìn/m2/tháng' in price:
        if pd.isna(square):
            return np.nan
        price = price.replace('Nghìn/m2/tháng','')
        price = float(price)/1000
        return price*square
    return np.nan

df_price_square = df_info[['Price','Square(m²)']]

# Thực hiện chuyển đổi
for i in df_price_square.index:
    df_price_square.at[i,'Price'] = convert_price(df_price_square.loc[i]['Price'],df_price_square.loc[i]['Square(m²)'])

# Lưu vào dataframe
df_info['Price'] =  pd.to_numeric(df_price_square['Price'])
df_info.rename(columns={'Price':'Price(Triệu/tháng)'},inplace=True)

df_info["Price(Triệu/tháng)"]
```

Hình 2.8: Xử lý dữ liệu cột price

Date: Biểu diễn ngày tháng năm nhưng lại không có dạng dữ liệu là datetime mà là object

```
df_info["Date"] = pd.to_datetime(df_info["Date"],format = "%d/%m/%Y",errors = 'raise')

df_info["Date"]
```

Id	Date
610	2014-12-12
1401	2015-03-30
2819	2016-04-23
2847	2016-07-23
3049	2017-10-08
...	...
169945	2024-09-11
169946	2024-09-15
169949	2024-09-17
169950	2024-09-17
169955	2024-09-20

15948 rows × 1 columns

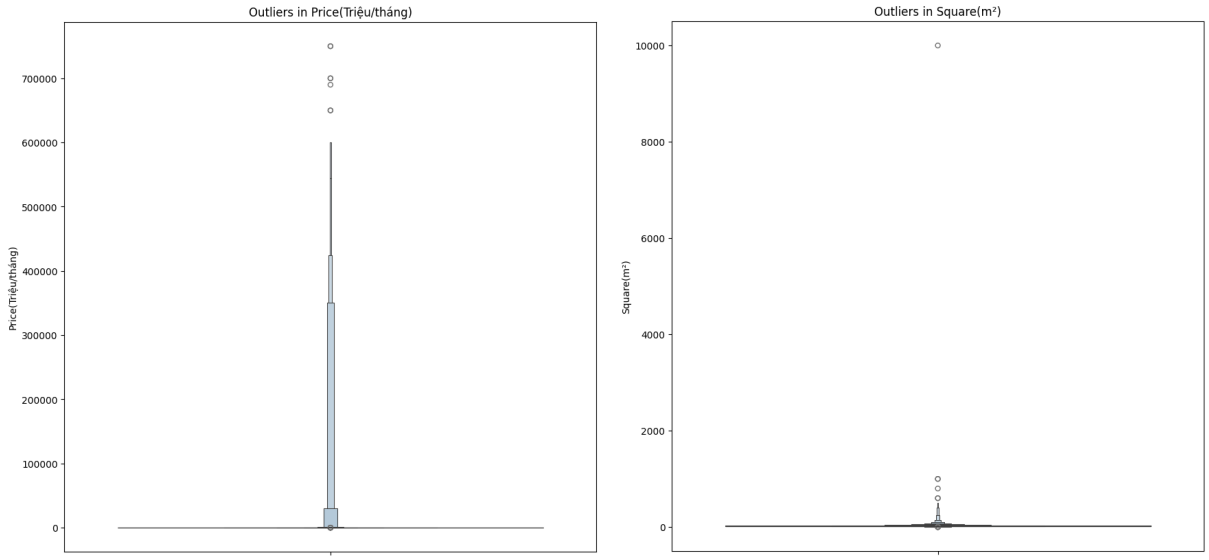
Hình 2.9: Xử lý dữ liệu cột Date

Xử lý dữ liệu NaN:

Vì đây là những thuộc tính khá khó để suy hay đoán và có rất ít dữ liệu nên khi drop nó đi cũng không ảnh hưởng khi xây dựng model sau này.

Outliers:

Biểu đồ boxplot để tìm outliers của giá và diện tích



Hình 2.10: Biểu đồ boxplot của giá và diện tích trước khi xử lý

- Thấy giá có outliers có giá trị trên 100 có thể là do người đăng tải thông tin đã nhầm đơn vị nên ta sẽ tiền hành giảm về con số đúng.
Giá đúng của những con số này là khoảng dưới 10.

- Còn những điểm trên 50 ta thu được 3 giá trị

```
outliers_price = df_info[df_info['Price(Triệu/tháng)'] > 50]
outliers_price
```

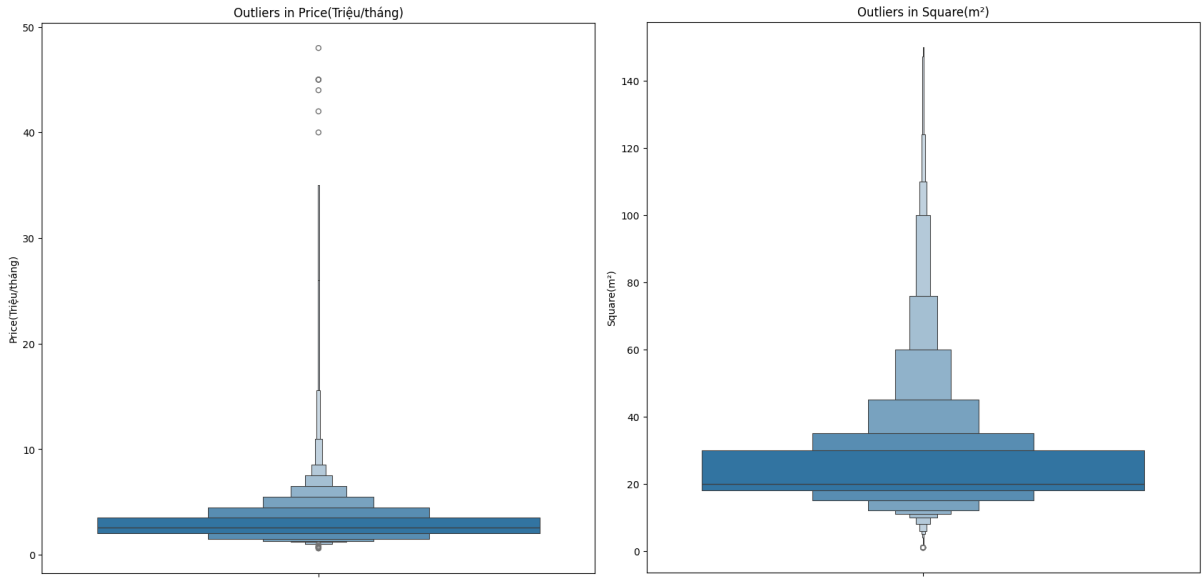
Id	Title	Price(Triệu/tháng)	Square(m²)	District	City	Date
121523	Bán nhà mặt tiền khu biệt thự Lam Sơn giá rẻ	81.0	100.0	Bình Thạnh	Hồ Chí Minh	2015-02-06
158080	Dịch vụ cho thuê căn hộ mini Q. Bình Thạnh	63.0	30.0	Bình Thạnh	Hồ Chí Minh	2020-06-29
166836	Tìm nữ ở ghép chung cư cao cấp sunrise city vi...	95.0	38.0	Quận 7	Hồ Chí Minh	2021-03-20

Hình 2.11: Xử lý các dữ liệu vô lý

Thấy dòng đầu tiên là bán, dòng tiếp là cho thuê cả căn hộ, còn dòng cuối người đăng cũng đã nhập sai giá trị thực nên ta chủ động loại bỏ hết những dòng này.

- Thấy diện tích có outliers có giá trị trên 150 sẽ rơi vào những trường hợp như kí túc xá, hay cho thuê 1 phòng đơn nhưng đăng diện tích cả nhà hoặc đăng sai giá trị thực nên ta cũng loại hết những dòng này.

Biểu đồ boxplot của giá và diện tích sau khi đã xử lý:



Hình 2.12: Biểu đồ boxplot của giá và diện tích sau khi đã xử lý

Tên	%đóng góp	Ghi chú
Chu Văn Mạnh	30%	clean_data(code), báo cáo
Trần Việt Hoàng	30%	clean_data(code), báo cáo
Trần Hoàng Hiệp	30%	clean_data(code), báo cáo
Trần Vũ Cảnh	5%	báo cáo
Nguyễn Hải Nam	5%	báo cáo

Hình 2.13: Phần trăm đóng góp tuần 2

Tuần Topic 3 (20/09 - 27/09)

Chủ đề tìm hiểu tuần topic 3: Khám phá và trực quan hóa dữ liệu.

Phân bố dữ liệu

Ta nhận thấy rằng các cột có thể tính toán sự phân bố đó là Price (Triệu/tháng), Square (m²). Ta sẽ tính các giá trị:

- Giá trị min
- Giá trị lower quartile (phân vị 25)
- Giá trị median (phân vị 50)
- Giá trị upper quartile (phân vị 75)
- Giá trị max

```
num_col_info_df = pd.DataFrame([],index= [ "min", "lower_quartile", "median", "upper_quartile", "max","mean" ])
col = ['Price(Triệu/tháng)','Square(m²)']

for i in col:
    distribu = np.nanpercentile(df_info[i],(0,25,50,75,100)) #tinh min,lower,median,upper,max
    mean = df_info[i].mean()
    data = np.append(distribu,mean)
    num_col_info_df[i] = data.round(1)
num_col_info_df
```

	Price(Triệu/tháng)	Square(m²)
min	0.6	10.0
lower_quartile	2.0	18.0
median	2.6	20.0
upper_quartile	3.5	28.0
max	10.0	80.0
mean	3.0	24.2

Hình 2.14: Các thông số

Với các cột không phải dạng số: Title, District, City, ta sẽ tính:

- Tỉ lệ % (từ 0 đến 100) các giá trị thiếu
- Số lượng giá trị khác nhau
- Giá trị xuất hiện nhiều nhất
- Giá trị xuất hiện ít nhất

```
col = ['Title','District','City']
ob_col_info_df = pd.DataFrame([],index = ['missing ratio','num diff value','Most appear','Min appear'])

for i in col:
    missing = np.array(df_info[i].isna().sum()*100/len(df_info)) #tinh phan tram gia tri thieu
    diff_value = len(df_info[i].dropna().unique())
    count_value = df_info[i].dropna().value_counts()
    most_ap = count_value[0]
    least_ap = count_value[-1]
    ob_col_info_df[i] = [missing,diff_value,most_ap,least_ap]
ob_col_info_df
```

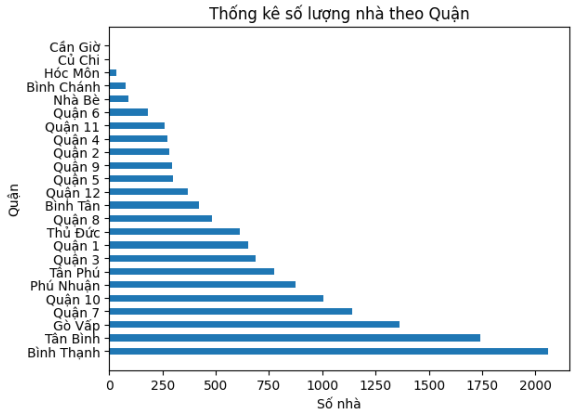
	Title	District	City
missing ratio	0.0	0.0	0.0
num diff value	12605	24	1
Most appear	58	2058	13977
Min appear	1	1	13977

Hình 2.15: Các thông số

Ta sẽ vẽ một vài biểu đồ để xem xét rõ hơn sự phân bố.

```
def draw_chart(df_col,til,x_name,y_name,typechart = 0):
    #Tính tần số và sắp xếp giảm dần
    data_num = df_col.value_counts()

    #Kiểm tra xem có phải là dạng barh không
    if typechart==1:
        plt.barh(data_num.index,data_num.values,height= 0.5)
    else:
        plt.bar(data_num.index,data_num.values)

    plt.title(til)
    plt.xlabel(x_name)
    plt.ylabel(y_name)
    plt.show()
```

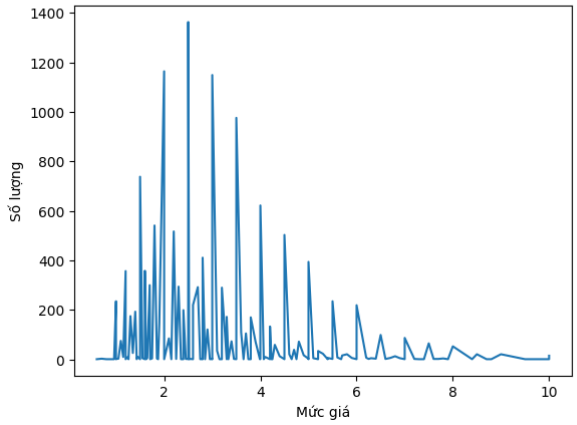
```
draw_chart(df_info["District"],"Thống kê số lượng nhà theo Quận","Số nhà","Quận",1)
```



Hình 2.16: Biểu đồ số lượng nhà theo Quận

Ta biểu diễn số lượng nhà theo giá:

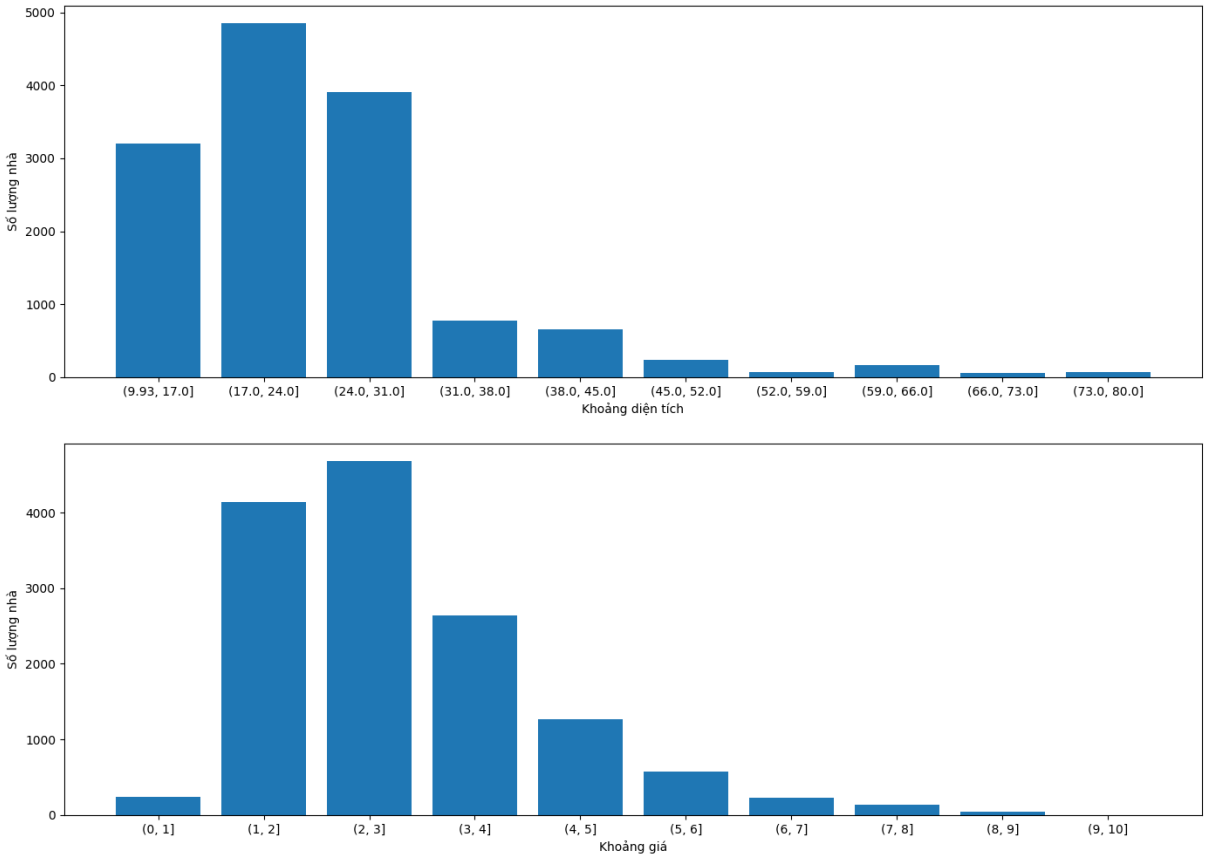
```
count_price = df_info["Price(Triệu/tháng)"].value_counts()
count_price = count_price.sort_index()

plt.plot(count_price.index,count_price.values)
plt.xlabel('Mức giá')
plt.ylabel('Số lượng')
plt.show()
```



Hình 2.17: Biểu đồ số lượng nhà theo giá

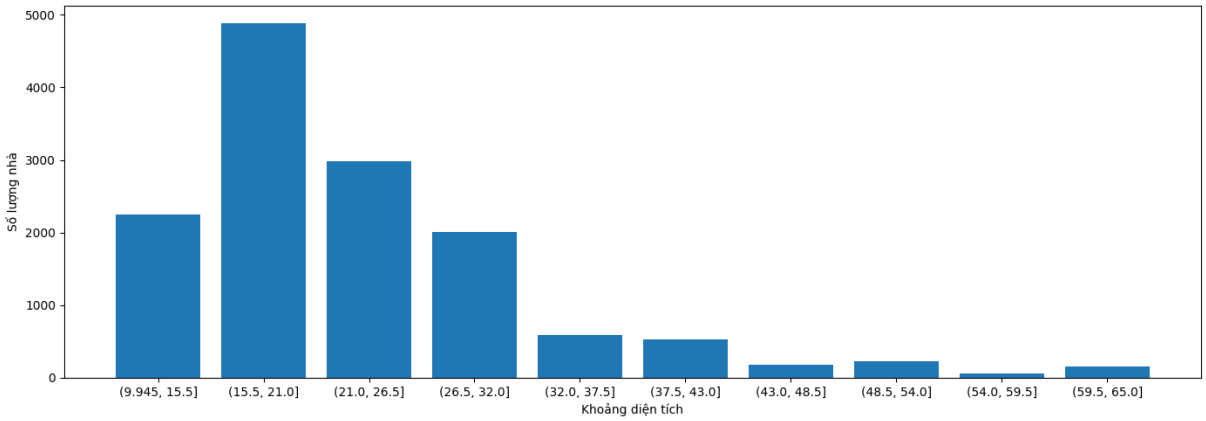
Ta thấy dữ liệu tập trung chủ yếu từ 0 đến 10 triệu, vậy ta sẽ lấy dữ liệu khoảng này để vẽ rõ hơn (vẽ bằng bar chart).

Ta cũng vẽ đồ thị biểu diễn số lượng nhà theo diện tích.



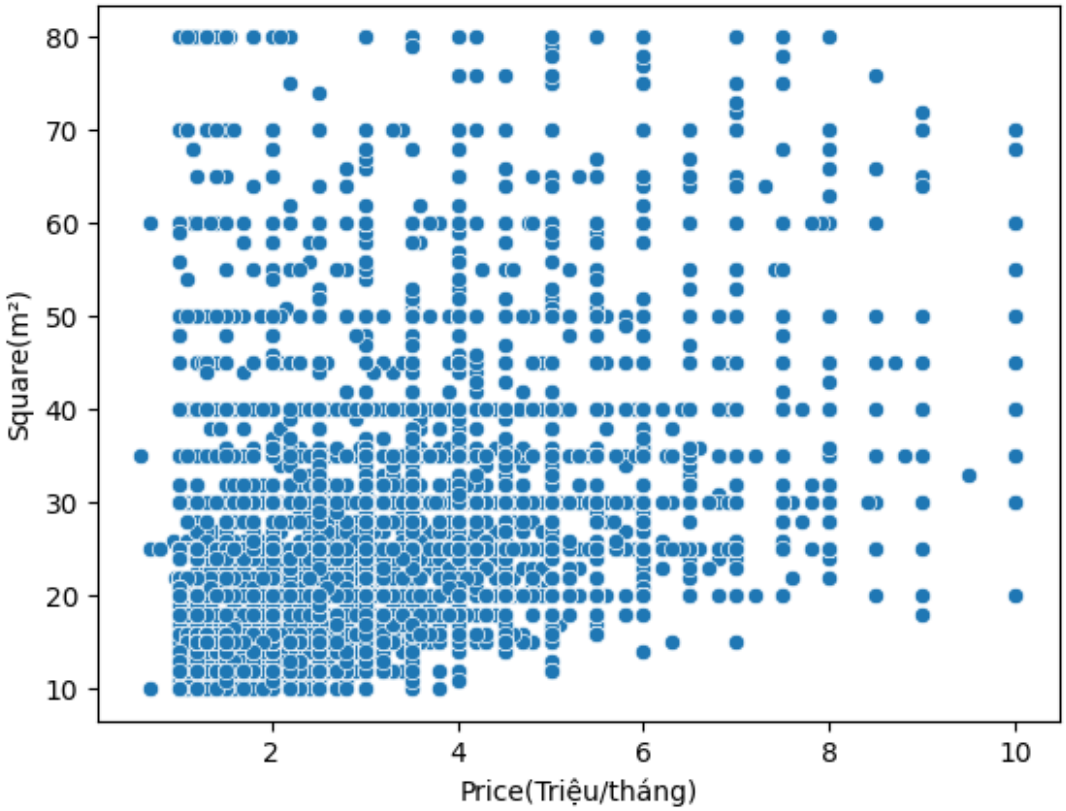
Hình 2.18: Biểu đồ số lượng nhà theo giá

Dữ liệu tập trung nhiều ở khoảng dưới 65, vậy ta sẽ dùng dữ liệu khoảng này để vẽ lại rõ hơn.



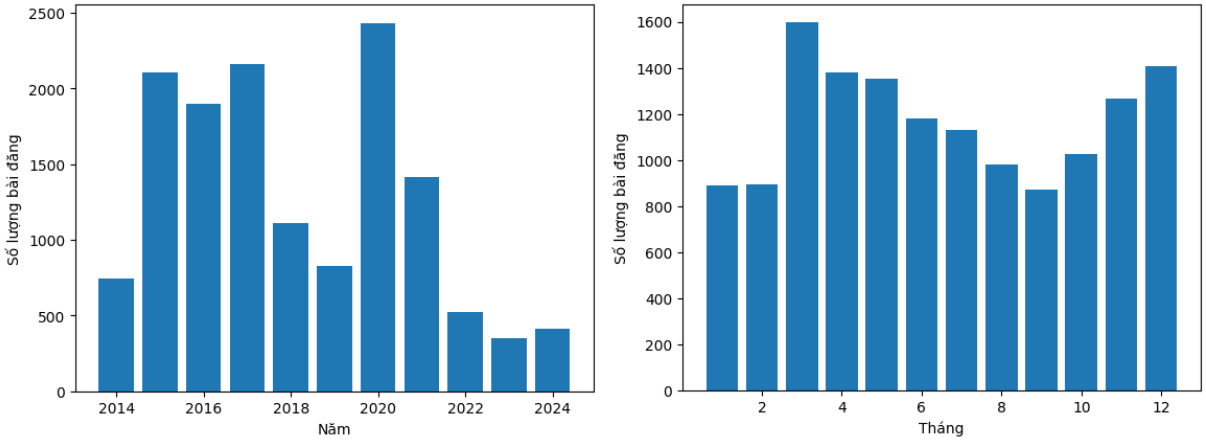
Hình 2.19: Biểu đồ số lượng nhà theo diện tích

Để xem mối liên hệ giữa diện tích và giá nhà ta sẽ dùng đồ thị scatterplot.



Hình 2.20: Đồ thị Square và Price

Thống kê số nhà theo năm và tháng.



Hình 2.21: Đồ thị Square và Price

Tên	%đóng góp	Ghi chú
Chu Văn Mạnh	30%	clean_data(code), báo cáo
Trần Việt Hoàng	30%	clean_data(code), báo cáo
Trần Hoàng Hiệp	30%	clean_data(code), báo cáo
Trần Vũ Cảnh	5%	báo cáo
Nguyễn Hải Nam	5%	báo cáo

Hình 2.22: Phần trăm đóng góp tuần 3

Tuần Topic 4 (27/09 - 04/10)

Chủ đề tìm hiểu tuần topic 4: Phân loại mô hình và Đánh giá mô hình

Phân loại bài toán

Trong học máy, các bài toán thường được phân loại thành ba loại chính:

- **Phân loại (Classification)**: Dự đoán một nhãn hoặc một lớp. Ví dụ: phân loại email là spam hoặc không spam.
- **Hồi quy (Regression)**: Dự đoán một giá trị liên tục. Ví dụ: dự đoán giá nhà, giá cổ phiếu.
- **Phân cụm (Clustering)**: Nhóm các điểm dữ liệu thành các cụm mà không có nhãn trước đó. Ví dụ: phân cụm khách hàng thành các nhóm dựa trên hành vi mua sắm.

Đặc điểm của bài toán dự đoán giá thuê phòng trọ

Bài toán của chúng tôi có các đặc điểm sau:

- **Đầu vào**: Các đặc trưng của phòng trọ: diện tích, quận, thành phố, ngày, tháng, năm,...
- **Đầu ra**: Giá thuê phòng trọ, là một giá trị liên tục.

Lý do bài toán thuộc hồi quy

- **Giá trị liên tục**: Giá thuê phòng trọ là một giá trị liên tục, không phải là một nhãn hoặc lớp. Điều này làm cho bài toán thuộc về hồi quy.
- **Thuật toán sử dụng**: Sử dụng các thuật toán hồi quy `LinearRegression`, `Ridge`, `Lasso`, `KNeighborsRegresso` và `RandomForestRegressor`. Đây là các thuật toán được thiết kế để dự đoán giá trị liên tục.
- **Mục tiêu dự đoán**: Mục tiêu là dự đoán một giá trị cụ thể (giá thuê phòng trọ) dựa trên các đặc trưng đầu vào, không phải là phân loại phòng trọ vào các nhóm hoặc cụm.

Kết luận

Dựa trên các đặc điểm và mục tiêu của bài toán, kết luận rằng bài toán dự đoán giá thuê phòng trọ tại thành phố Hồ Chí Minh là một bài toán hồi quy. Sử dụng các thuật toán hồi quy để xây dựng mô hình dự đoán giá trị liên tục của giá thuê phòng trọ dựa trên các đặc trưng đầu vào.

Từ đó ta sẽ làm việc với 5 thuật toán như ở **topic** trước để xây dựng mô hình.

Tên	Đóng góp
Chu Văn Mạnh	30%
Trần Việt Hoàng	30%
Trần Hoàng Hiệp	30%
Trần Vũ Cảnh	5%
Nguyễn Hải Nam	5%

Hình 2.23: Phần trăm đóng góp tuần 4

Tuần Topic 5 (04/10 - 11/10)

Chủ đề tìm hiểu tuần topic 5: Xây dựng và Đánh giá mô hình

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import date
import seaborn as sns
from datetime import datetime
import warnings
warnings.filterwarnings('ignore')
```

Hình 2.24: Import các thư viện

Tiếp tục tiền xử lý dữ liệu:

- Loại bỏ các cột không cần thiết

- Chuyển đổi trường Date sang kiểu Numberic

	Id	Title	Price(Triệu/tháng)	Square(m²)	District	City	Date
0	169955	Phòng 5 tr/tg. Dc 269 nguyễn thái bình bình, p...	5.0	25.0	Tân Bình	Hồ Chí Minh	2024-09-20
1	129967	Cho thuê phòng trọ dạng chung cư mini máy lạnh...	1.8	18.0	Quận 8	Hồ Chí Minh	2024-09-20
2	151366	PHÒNG SẠCH THOÁNG, SÂN RỘNG MÁT, LỐI ĐI RIÊNG ...	1.5	12.0	Tân Bình	Hồ Chí Minh	2024-09-20
3	169844	Cho thuê CHDV 25-30m2 nội thất cao cấp, ban cô...	5.5	25.0	Gò Vấp	Hồ Chí Minh	2024-09-19
4	168697	Cho thuê phòng có nội thất giá rẻ gần chợ Tân...	5.5	30.0	Quận 1	Hồ Chí Minh	2024-09-19
...	...	...	...	...	...	...	...
13972	110296	PHÒNG VIP CHO THUÊ - BX MIỀN ĐÔNG	2.0	25.0	Bình Thạnh	Hồ Chí Minh	2014-08-19
13973	122097	PHÒNG MỚI, ĐẸP, TIỆN NGHI, AN NINH	2.5	15.0	Quận 4	Hồ Chí Minh	2014-08-19
13974	122087	CHO THUÊ PHÒNG Ở CAO CẤP ĐƯỜNG CMT8, Q.10, PHÙ...	4.0	20.0	Quận 10	Hồ Chí Minh	2014-08-19
13975	122109	Cần 1 nữ ở ghép	1.5	15.0	Quận 7	Hồ Chí Minh	2014-08-19
13976	121875	Cho thuê phòng trọ phường Tân Kiểng, Quận 7	2.5	25.0	Quận 7	Hồ Chí Minh	2014-08-18

```
train_df_temp.drop('City', axis= 1, inplace = True)
train_df_temp['Date'] = pd.to_datetime(train_df_temp['Date'])
train_df_temp['Date'] = train_df_temp['Date'].map(pd.Timestamp.toordinal)
```

Hình 2.25: Xử lý dữ liệu

Tạo thêm đặc trưng cho bảng dữ liệu

Dựa vào đặc trưng: 'Title' để tạo thêm 6 đặc trưng: isNew(phòng mới hay cũ) , hasFurniture (có nội thất không), liveTogether(ở ghép/ktx), isWholeHouse (có phải nhà nguyên căn không), isApartment (có phải căn hộ không),

isNearCenter (có gần trung tâm không)

```
def isWholeHouse(df):
    df["Title"] = df["Title"].str.lower()
    e = ['nguyên']
    df['isWholeHouse'] = df["Title"].apply(lambda x: 'yes' if any(i in x for i in e) else 'no')
    return df
def isLiveTogether(df):
    df["Title"] = df["Title"].str.lower()
    e = ['ghép','share',"ktx",'túc']
    df['isLiveTogether'] = df["Title"].apply(lambda x: 'yes' if any(i in x for i in e) else 'no')
    return df
def isApartment(df):
    df["Title"] = df["Title"].str.lower()
    e = ['hộ','cư', 'chdv']
    df['isApartment'] = df["Title"].apply(lambda x: 'yes' if any(i in x for i in e) else 'no')
    return df
def isNew(df):
    df["Title"] = df["Title"].str.lower()
    e = ['mới', 'vừa', 'xây']
    df['isNew'] = df["Title"].apply(lambda x: 'yes' if any(i in x for i in e) else 'no')
    return df
def isNearCenter(df):
    df["Title"] = df["Title"].str.lower()
    e = ['tâm', 'tam']
    df['isNearCenter'] = df["Title"].apply(lambda x: 'yes' if any(i in x for i in e) else 'no')
    return df
def hasFurniture(df):
    df["Title"] = df["Title"].str.lower()
    e = ['nội thất', 'noi that', 'tiện nghi','tiện ích','cao cấp','cao cap','hiện đại','hien dai']
    df['hasFurniture'] = df["Title"].apply(lambda x: 'yes' if any(i in x for i in e) else 'no')
    return df
train_df_temp = isNew(train_df_temp)
train_df_temp = isNearCenter(train_df_temp)
train_df_temp = hasFurniture(train_df_temp)
train_df_temp = isWholeHouse(train_df_temp)
train_df_temp = isLiveTogether(train_df_temp)
train_df_temp = isApartment(train_df_temp)
```

Hình 2.26: Trích xuất các đặc trưng

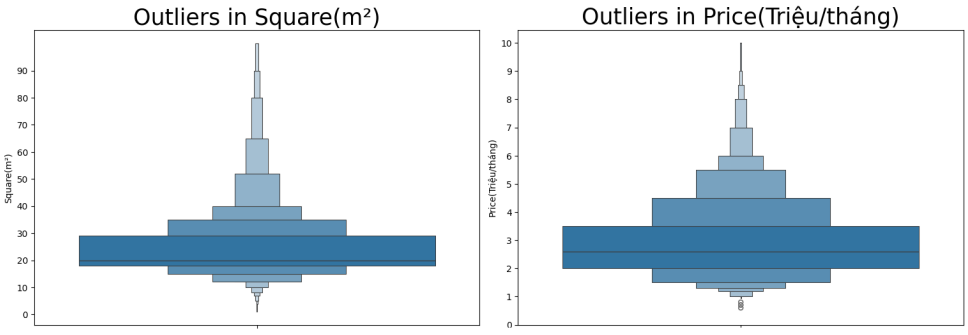
Bỏ đi những cột có tương quan thấp với giá

- Ta thấy ở thuộc tính giá dữ liệu tập trung dưới 10 triệu và ở diện tích dữ liệu tập trung dưới 100 m² nên ta sẽ loại những dòng trên 10 triệu (giá) và 100 m² (Diện tích).

```
corr_price = df_new.corr()['Price']
drop_columns = corr_price[abs(corr_price)<0.01].index
df_new.drop(columns = drop_columns,inplace = True)

df_new.drop(columns=['Price']).columns.tolist()
```

Hình 2.27: Xử lý dữ liệu

Loại bỏ outliers



Hình 2.28: Outliers in Square và Outliers in Price

Xây dựng mô hình

Chia tập train/test với tỷ lệ là 75:25

Chọn các mô hình

Ta sẽ dùng phương pháp K Fold Cross Validation để đánh giá và chọn ra các mô hình tốt nhất. Sau khi chạy thử với nhiều k thì tụi em chọn k=7 vì nó có hiệu quả tốt nhất.

Lý do dùng: Bởi vì tập dữ liệu khá nhỏ, nếu chia ra các tập train/val/test thì không thật sự đo được độ hiệu quả của mô hình (vì mỗi tập quá ít dữ liệu). Dùng K Fold Cross Validation thì mọi phần trong tập train sẽ được dùng cho cả việc train và test, nhờ đó ta sẽ đánh giá chính xác hơn độ hiệu quả của mô hình.

Ta sẽ chọn các thuật toán là Linear Regression, Ridge, Lasso, K-Neighbors Regressor, Random Forest

```
model = LinearRegression()
scores = cross_val_score(model, X_train, y_train, cv=7)
scores.mean().round(5)
```

```
0.49211
```

```
model = Ridge()
scores = cross_val_score(model, X_train, y_train, cv=7)
scores.mean().round(5)
```

```
0.49266
```

```
model = Lasso()
scores = cross_val_score(model, X_train, y_train, cv=7)
scores.mean().round(5)
```

```
0.22409
```

```
model = KNeighborsRegressor()
scores = cross_val_score(model, X_train, y_train, cv=7)
scores.mean().round(5)
```

```
0.2067
```

```
model = RandomForestRegressor()
scores = cross_val_score(model, X_train, y_train, cv=7)
scores.mean().round(5)
```

```
0.53331
```

Hình 2.29: Đánh giá các thuật toán

Ta sẽ chọn ra 3 mô hình có điểm cao nhất là Linear Regression, Ridge, Random Forest

Chọn các mô hình

Cho các mô hình này lần lượt chạy trên tập train và test để đo độ hiệu quả của chúng.
Ta sẽ tính các giá trị đo độ lỗi như sau:

MAE: lỗi trị tuyệt đối trung bình MSE: lỗi bình phương trung bình RMSE: sai số bình phương trung bình căn R^2score: mức độ phù hợp của mô hình với tập dữ liệu

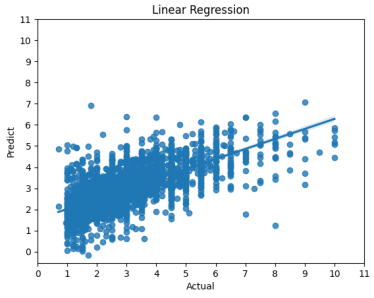
Linear Regression

```
model_LN = LinearRegression()
model_LN.fit(X_train,y_train)

print("======TRAIN======")
y_pre_train = model_LN.predict(X_train)
score_train = mean_absolute_error(y_train,y_pre_train).round(5)
mse_score = mean_squared_error(y_train,y_pre_train).round(5)
rmse = mean_squared_error(y_train,y_pre_train,squared = False).round(5)
score_r2 = round(r2_score(y_train,y_pre_train), 5)
print("Train error (MAE): {}".format(score_train))
print("Train error (MSE): {}".format(mse_score))
print("Train error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
print("======TEST======")
y_pre_test = model_LN.predict(X_test)
score_test = mean_absolute_error(y_test,y_pre_test).round(5)
mse_score = mean_squared_error(y_test,y_pre_test).round(5)
rmse = mean_squared_error(y_test,y_pre_test,squared = False).round(5)
score_r2 = round(r2_score(y_test,y_pre_test), 5)
print("Test error (MAE): {}".format(score_test))
print("Test error (MSE): {}".format(mse_score))
print("Test error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
sns.regplot(x = y_test, y = y_pre_test)
plt.title('Linear Regression')
plt.xlabel('Actual')
plt.ylabel('Predict')
plt.xticks(range(0,12,1))
plt.yticks(range(0,12,1))
plt.show()
```

```
======TRAIN======
Train error (MAE): 0.70183
Train error (MSE): 0.96888
Train error (RMSE): 0.98432
R2 score : 0.49727
======TEST======
Test error (MAE): 0.70749
Test error (MSE): 1.00085
Test error (RMSE): 1.00043
R2 score : 0.48302
```

Hình 2.30: Code mô hình Linear Regression

Trên tập train Linear thấy độ lỗi không cao nhưng $R^2 score$ lại thấp và khi cho chạy với tập test thì lại có kết quả không tốt hơn tập train.



Hình 2.31: Mô hình Linear Regression

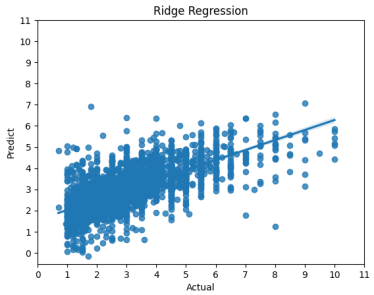
Ridge

```
model = Ridge()
model.fit(X_train,y_train)

print("======TRAIN======")
y_pre_train = model.predict(X_train)
score_train = mean_absolute_error(y_train,y_pre_train).round(5)
mse_score = mean_squared_error(y_train,y_pre_train).round(5)
rmse = mean_squared_error(y_train,y_pre_train,squared = False).round(5)
score_r2 = round(r2_score(y_train,y_pre_train), 5)
print("Train error (MAE): {}".format(score_train))
print("Train error (MSE): {}".format(mse_score))
print("Train error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
print("======TEST======")
y_pre_test = model.predict(X_test)
score_test = mean_absolute_error(y_test,y_pre_test).round(5)
mse_score = mean_squared_error(y_test,y_pre_test).round(5)
rmse = mean_squared_error(y_test,y_pre_test,squared = False).round(5)
score_r2 = round(r2_score(y_test,y_pre_test), 5)
print("Test error (MAE): {}".format(score_test))
print("Test error (MSE): {}".format(mse_score))
print("Test error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
sns.regplot(x = y_test, y = y_pre_test)
plt.title('Ridge Regression')
plt.xlabel('Actual')
plt.ylabel('Predict')
plt.xticks(range(0,12,1))
plt.yticks(range(0,12,1))
plt.show()
```

```
======TRAIN======
Train error (MAE): 0.7019
Train error (MSE): 0.96898
Train error (RMSE): 0.98437
R2 score : 0.49722
======TEST======
Test error (MAE): 0.7075
Test error (MSE): 1.00089
Test error (RMSE): 1.00045
R2 score : 0.483
```

Hình 2.32: Code mô hình Ridge

Tương tự thì ở tập train Ridge có độ lỗi không cao nhưng $R^2 score$ lại thấp và khi cho chạy với tập test thì lại có kết quả không tốt hơn tập train.



Hình 2.33: Mô hình Ridge

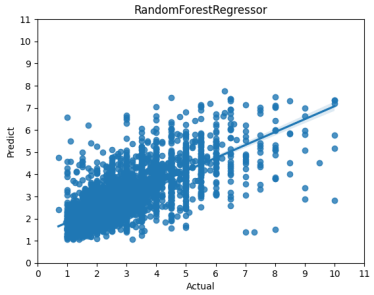
Random Forest Ridge

```
model = RandomForestRegressor()
model.fit(X_train,y_train)

print("======TRAIN======")
y_pre_train = model.predict(X_train)
score_train = mean_absolute_error(y_train,y_pre_train).round(5)
mse_score = mean_squared_error(y_train,y_pre_train).round(5)
rmse = mean_squared_error(y_train,y_pre_train,squared = False).round(5)
score_r2 = round(r2_score(y_train,y_pre_train), 5)
print("Train error (MAE): {}".format(score_train))
print("Train error (MSE): {}".format(mse_score))
print("Train error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
print("======TEST======")
y_pre_test = model.predict(X_test)
score_test = mean_absolute_error(y_test,y_pre_test).round(5)
mse_score = mean_squared_error(y_test,y_pre_test).round(5)
rmse = mean_squared_error(y_test,y_pre_test,squared = False).round(5)
score_r2 = round(r2_score(y_test,y_pre_test), 5)
print("Test error (MAE): {}".format(score_test))
print("Test error (MSE): {}".format(mse_score))
print("Test error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
sns.regplot(x = y_test, y = y_pre_test)
plt.title('RandomForestRegressor')
plt.xlabel('Actual')
plt.ylabel('Predict')
plt.xticks(range(0,12,1))
plt.yticks(range(0,12,1))
plt.show()
```

```
======TRAIN======
Train error (MAE): 0.24622
Train error (MSE): 0.12716
Train error (RMSE): 0.3566
R2 score : 0.93402
======TEST======
Test error (MAE): 0.67437
Test error (MSE): 0.95575
Test error (RMSE): 0.97762
R2 score : 0.50631
```

Hình 2.34: Code mô hình Random Forest

Ngược lại thì tập train Random Forest cho kết quả rất tốt, độ lỗi thấp và R^2score rất cao. Tuy nhiên thực tế lại đưa ra kết quả thấp hơn 40/100 so với lúc train (tuy nhiên vẫn cao hơn 2 mô hình kia). Mô hình này đã rơi vào trường hợp overfitting.



Hình 2.35: Mô hình Ridge

Đánh giá mô hình

Nhìn chung, cả 3 mô hình được chọn để xây dựng: Linear Regression, Ridge, Random Forest và kể cả phương pháp Ensemble(stacking) kết hợp nhiều mô hình nhưng vẫn chỉ cho R^2score ở mức 0.5. Chưa đạt kì vọng ban đầu nhóm đề ra.

Nguyên nhân chủ quan có thể là do xử lý dữ liệu chưa triệt để (cũng rất khó vì nhóm đã làm rất kỹ phần xử lý tiền dữ liệu). Nguyên nhân khách quan là do giá nhà ở trang web thu thập cung cấp thông tin chưa thực sự sát với thị trường, còn mang nhiều sự cảm tính, và biến động giá nhà ở Việt Nam là rất lớn.

Phương án chính: Loại bỏ nhiễu bằng cách quan sát biểu đồ, và loại bỏ nhiễu bằng thủ công sau đó cho chạy trên 3 mô hình: Linear Regression, Ridge, Random Forest thì cho kết quả ở mức $R^2 score$ là khoảng 0.5. Kết hợp với những đánh giá cụ thể từ từng mô hình đã đi đến quyết định chọn mô hình Random Forest cho việc dự đoán giá nhà.

Tên	Đóng góp
Chu Văn Mạnh	30%
Trần Việt Hoàng	30%
Trần Hoàng Hiệp	30%
Trần Vũ Cảnh	5%
Nguyễn Hải Nam	5%

Hình 2.36: Phần trăm đóng góp tuần 5

Tuần Topic 6 (11/10 - 18/10)

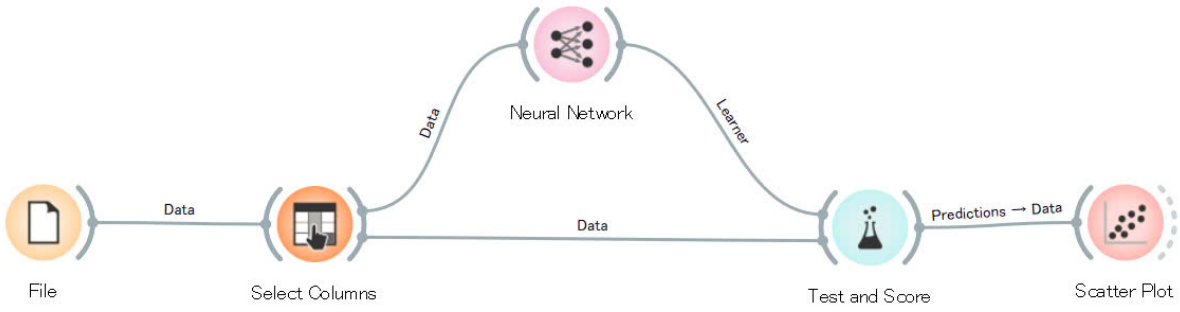
Chủ đề tìm hiểu tuần 6: Xây dựng Neural Network với mô hình dự đoán.

Neural Network

Neural Network (Mạng nơ-ron nhân tạo) là một mô hình toán học được xây dựng dựa trên cấu trúc và hoạt động của bộ não con người, được thiết kế để giúp máy tính học hỏi từ dữ liệu và thực hiện các tác vụ như phân loại, dự đoán, và nhận diện mẫu. Mạng nơ-ron là nền tảng của trí tuệ nhân tạo (AI) và học máy (Machine Learning).

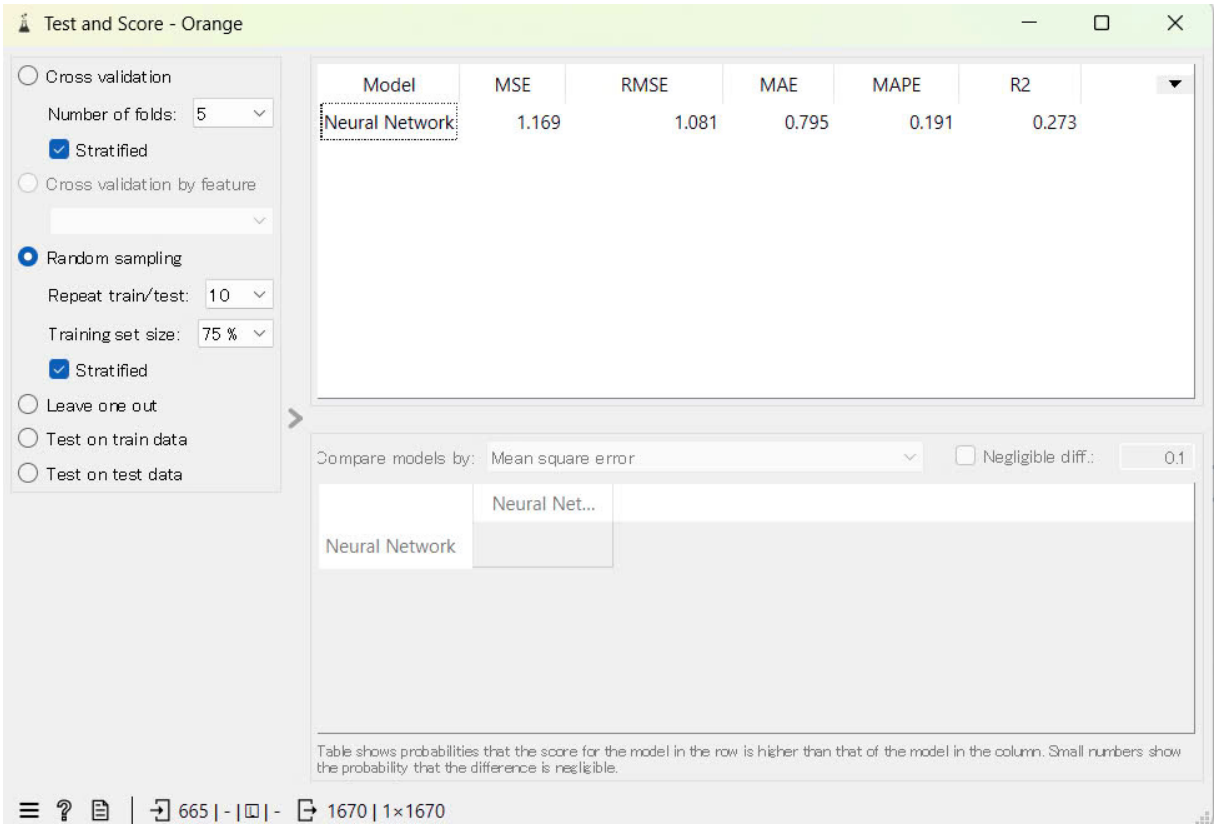
Trong Orange, có thể sử dụng Neural Network để huấn luyện mô hình trên tập dữ liệu và dự đoán kết quả dựa trên dữ liệu mới.

Mô hình trong Orange Data Mining



Hình 2.37: Mô hình trong Orange Data Mining

Kết quả thu được



Hình 2.38: Kết quả thu được

Đánh giá Mô hình Dự đoán

MSE (Mean Squared Error) Giá trị này là 1.169, cho thấy mức độ sai số trung bình bình phương giữa giá trị dự đoán và giá trị thực tế. MSE càng nhỏ, mô hình càng chính xác, nhưng giá trị hiện tại cho thấy còn sai số.

RMSE (Root Mean Squared Error) Đây là căn bậc hai của MSE, giá trị là 1.081. Chỉ số này giúp diễn giải dễ dàng hơn vì nó ở cùng đơn vị với giá trị dự đoán. Giá trị càng nhỏ thì dự đoán càng chính xác.

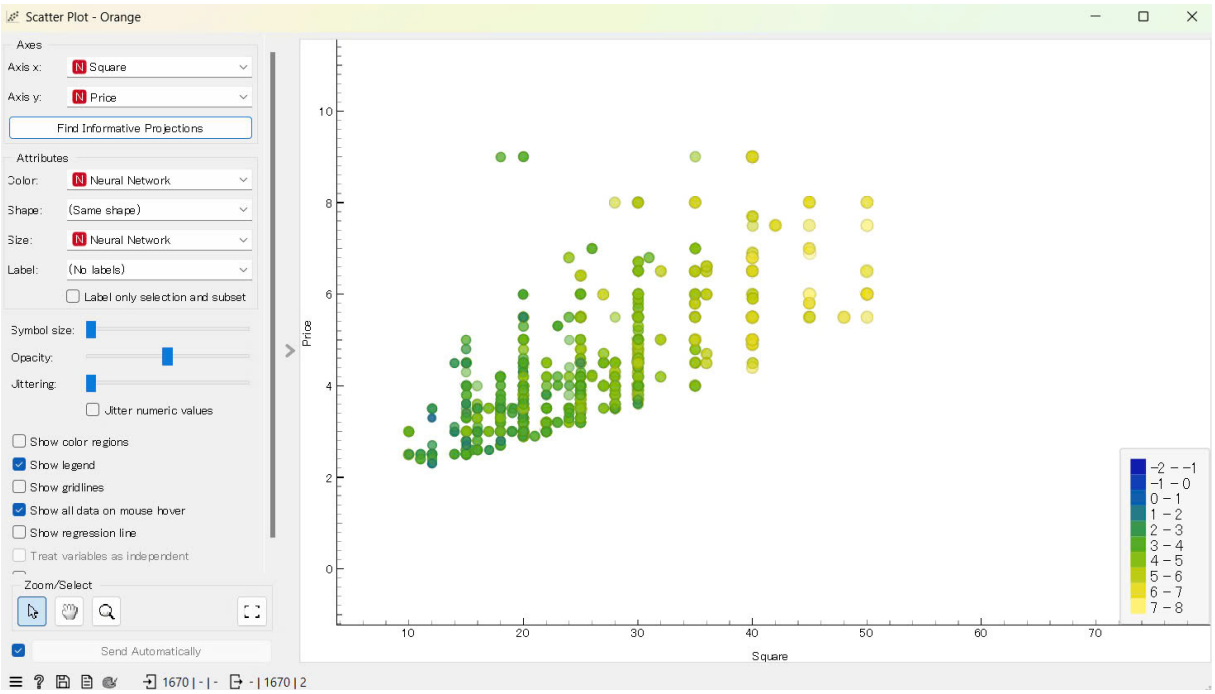
MAE (Mean Absolute Error) Với giá trị 0.795, đây là độ lớn trung bình của lỗi dự đoán không xét dấu. Nó cho thấy sai số trung bình giữa giá trị dự đoán và giá trị thực tế, giúp dễ dàng hiểu hơn về độ chính xác của mô hình.

MAPE (Mean Absolute Percentage Error) Giá trị 0.191 đại diện cho tỷ lệ phần trăm sai số so với giá trị thực tế, và càng nhỏ càng tốt.

R^2 (R-squared) Giá trị 0.273 cho thấy mô hình này giải thích được 27.3% phương sai trong dữ liệu. Giá trị R^2 thấp cho thấy mô hình không thể giải thích tốt mối quan hệ giữa đầu vào và đầu ra.

Nhận xét chung Các chỉ số lỗi như MSE, RMSE, và MAE đều không quá lớn, nhưng chưa đạt mức lý tưởng, cho thấy mô hình cần được cải thiện thêm để dự đoán chính xác hơn. R^2 thấp (0.273) là dấu hiệu cho thấy mô hình này chưa tốt trong việc giải thích sự biến thiên của dữ liệu và có thể cần tinh chỉnh các tham số của Neural Network hoặc sử dụng các phương pháp tiền xử lý dữ liệu tốt hơn.

Đồ thị Price theo Square



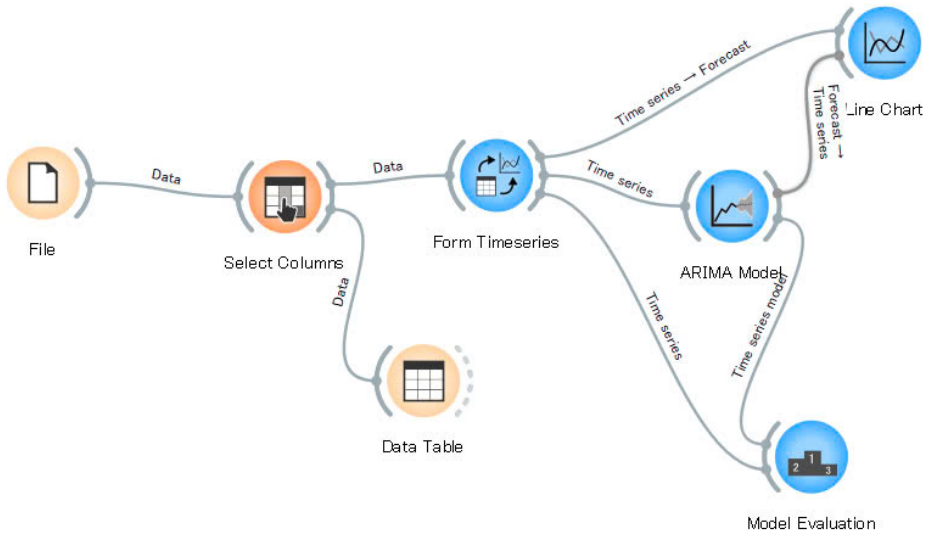
Hình 2.39: Đồ thị Price theo Square

Tên	Đóng góp
Chu Văn Mạnh	30%
Trần Việt Hoàng	30%
Trần Hoàng Hiệp	30%
Trần Vũ Cảnh	5%
Nguyễn Hải Nam	5%

Hình 2.40: Phần trăm đóng góp tuần 6

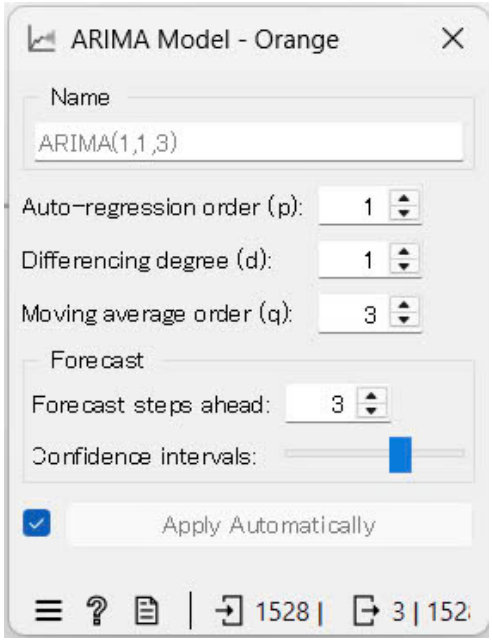
Tuần Topic 7 (18/10 - 25/10)

Chủ đề tìm hiểu tuần topic 7: Sử dụng time series để xử lý bài toán



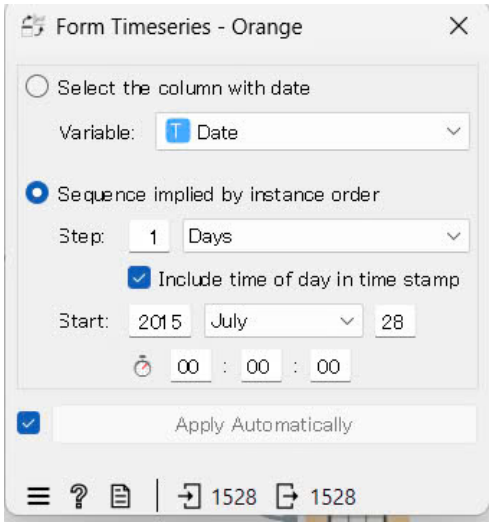
Hình 2.41: Đánh giá mô hình

Quy trình này giúp bạn thực hiện các bước từ việc chuẩn bị dữ liệu đến xây dựng và đánh giá mô hình dự báo ARIMA. Sơ đồ cho thấy một cách tiếp cận có hệ thống, từ việc xử lý dữ liệu đến đánh giá mô hình, nhằm đảm bảo rằng dự báo cuối cùng có độ chính xác cao. Phần hiển thị kết quả qua biểu đồ giúp dễ dàng hình dung sự khác biệt giữa giá trị thực tế và giá trị dự báo.



Hình 2.42: ARIMA Model

Mô hình ARIMA trong ảnh được cấu hình với các tham số ARIMA(1,1,3), có nghĩa là mô hình sẽ dựa vào 1 giá trị trước đó của chuỗi, sử dụng sai phân bậc 1 để xử lý dữ liệu không dừng, và sử dụng 3 giá trị của nhiễu trắng để dự đoán. Mô hình dự báo trước 3 bước và có dải dự báo với khoảng tin cậy có thể điều chỉnh.



Hình 2.43: Form Timeseries

Cửa sổ này thiết lập các tham số cơ bản cho việc chuyển đổi dữ liệu thành chuỗi thời gian. Tùy chọn đã chọn trong hình cho thấy dữ liệu chuỗi thời gian được xây dựng với tần suất hàng ngày, bắt đầu từ 28 tháng 7 năm 2015, và bao gồm cả thời gian trong ngày.

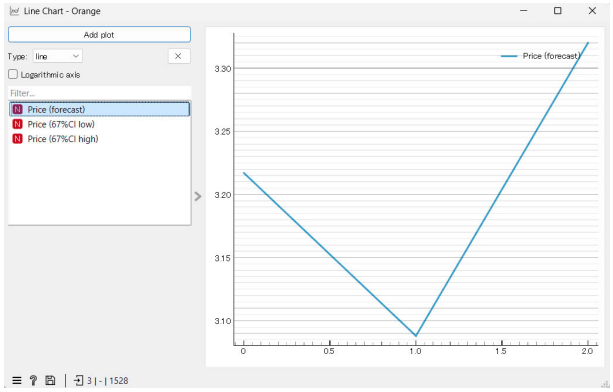
	RMSE	MAE	MAPE	POCID	R^2	AIC	BIC
ARIMA(1,1,3)	1.541	0.611	0.292	44.1	-0.039	4992.8	5019.3
ARIMA(1,1,3) (in-sample)	1.327	0.809	0.324	42.0	0.014	5219.6	5246.2

Hình 2.44: Các chỉ số đánh giá

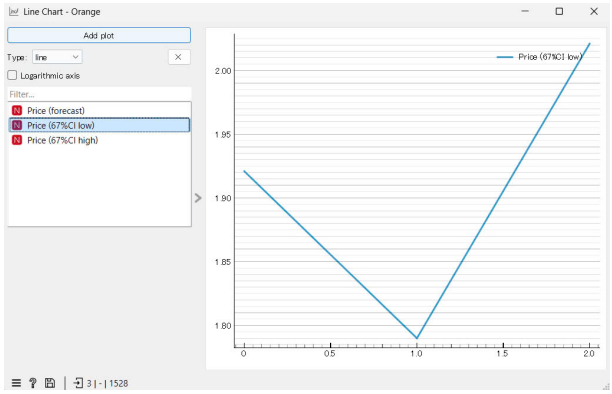
Dựa trên các chỉ số từ bảng kết quả của mô hình ARIMA(1,1,3), chúng ta có các nhận xét như sau:

- Mô hình in-sample có RMSE thấp hơn (1.327) so với out-sample (1.541), cho thấy mô hình hoạt động tốt hơn trên tập huấn luyện nhưng kém hơn trên tập kiểm tra.
- Giá trị MAE của in-sample cao hơn so với out-sample, cho thấy trên tập kiểm tra, các sai số giữa giá trị dự đoán và giá trị thực tế trung bình là nhỏ hơn.
- MAPE của mô hình in-sample cao hơn, có thể cho thấy rằng mô hình hơi khớp quá tốt trên tập huấn luyện nhưng không đạt hiệu quả tương tự trên tập kiểm tra.
- Giá trị POCID của cả hai trường hợp không quá cao, cho thấy mô hình không dự đoán tốt xu hướng thay đổi của giá trị thực.
- Giá trị R^2 âm hoặc gần 0 cho thấy mô hình không giải thích được tốt sự biến thiên của dữ liệu. R^2 âm (-0.039) cho thấy mô hình ARIMA(1,1,3) hoạt động kém hiệu quả trên tập kiểm tra, thậm chí không tốt hơn so với việc sử dụng giá trị trung bình làm dự đoán.

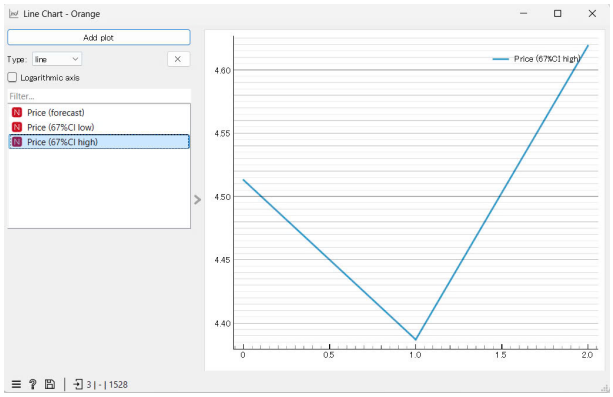
kết quả cảu quá trình xử lý:



Hình 2.45: Kết quả



Hình 2.46: Kết quả



Hình 2.47: Kết quả

Các hình ảnh này minh họa 3 kịch bản khác nhau về xu hướng giá phòng trọ theo thời gian, bao gồm một dự báo trung bình, một dự báo thận trọng (low CI), và một dự báo lạc quan (high CI).

Tên	Đóng góp
Chu Văn Mạnh	30%
Trần Việt Hoàng	30%
Trần Hoàng Hiệp	30%
Trần Vũ Cảnh	5%
Nguyễn Hải Nam	5%

Hình 2.48: Phần trăm đóng góp tuần 7

Tuần Topic 8 (25/10 - 01/11)

Chủ đề tìm hiểu tuần topic 8: Giảm chiều dữ liệu và Tối ưu hoá hàm lồi

Cả 3 mô hình được chọn để xây dựng: Linear Regression, Ridge, Random Forest và kể cả phương pháp Ensemble(stacking) kết hợp nhiều mô hình nhưng vẫn chỉ cho R2score ở mức 0.5. Chưa đạt kì vọng ban đầu nhóm đề ra.

Nguyên nhân chủ quan có thể là do xử lý dữ liệu chưa triệt để (cũng rất khó vì nhóm đã làm rất kỹ phần xử lý tiền dữ liệu). Nguyên nhân khách quan là do giá nhà ở trang web thu thập cung cấp thông tin chưa thực sự sát với thị trường, còn mang nhiều sự cảm tính, và biến động giá nhà ở Việt Nam là rất lớn.

Phương án chính: Loại bỏ nhiễu bằng cách quan sát biểu đồ, và loại bỏ nhiễu bằng thủ công sau đó cho chạy trên 3 mô hình: Linear Regression, Ridge, Random Forest thì cho kết quả ở mức R2score là khoảng 0.5. Kết hợp với những đánh giá cụ thể từ từng mô hình đã đi đến quyết định chọn mô hình **Random Forest** cho việc dự đoán giá nhà vì mô hình này phù hợp với bài toán gồm cả biến tuyến tình lẫn biến nhị phân.

Giảm chiều dữ liệu

Đánh giá các phương pháp giảm chiều:

- Principal Component Analysis (PCA): Thích hợp nếu bạn muốn giảm chiều dữ liệu mà vẫn giữ lại được nhiều thông tin nhất có thể. PCA sẽ giúp tìm các trục chính của dữ liệu và loại bỏ các chiều có phương sai thấp.
- Linear Discriminant Analysis (LDA): Nếu tập dữ liệu có nhãn phân loại, LDA sẽ hữu ích trong việc tối đa hóa sự khác biệt giữa các lớp.
- Nếu bạn có một lượng lớn dữ liệu phi tuyến, một mạng autoencoder có thể giúp giảm chiều dữ liệu mà vẫn giữ lại các đặc trưng chính.

Vì mô hình sử dụng **Random Forest** và dữ liệu có nhiều cột nhị phân, phương pháp phù hợp là PCA (Principal Component Analysis) để chọn lọc các đặc trưng quan trọng.

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Chuẩn hóa dữ liệu
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Áp dụng PCA để giảm chiều dữ liệu, chọn số thành phần chính dựa trên phần trăm phương sai
pca = PCA(n_components=0.95)  # Giữ lại 95% phương sai
X_reduced = pca.fit_transform(X_scaled)

# Kiểm tra số thành phần chính
print("Số thành phần chính sau PCA:", X_reduced.shape[1])
```

```
Số thành phần chính sau PCA: 28
```

Hình 2.49: Code phương pháp PCA

Chia lại các tập dữ liệu sau khi giảm và huấn luyện lại mô hình với RandomForest sau khi giảm chiều

```
# Chia lại dữ liệu sau khi giảm chiều
X_train, X_test, y_train, y_test = train_test_split(X_reduced, y, test_size=0.25, random_state=42)
```

```
model = RandomForestRegressor()
model.fit(X_train,y_train)

print("======TRAIN======")
y_pre_train = model.predict(X_train)
score_train = mean_absolute_error(y_train,y_pre_train).round(5)
mse_score = mean_squared_error(y_train,y_pre_train).round(5)
rmse = mean_squared_error(y_train,y_pre_train,squared = False).round(5)
score_r2 = round(r2_score(y_train,y_pre_train), 5)
print("Train error (MAE): {}".format(score_train))
print("Train error (MSE): {}".format(mse_score))
print("Train error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
print("======TEST======")
y_pre_test = model.predict(X_test)
score_test = mean_absolute_error(y_test,y_pre_test).round(5)
mse_score = mean_squared_error(y_test,y_pre_test).round(5)
rmse = mean_squared_error(y_test,y_pre_test,squared = False).round(5)
score_r2 = round(r2_score(y_test,y_pre_test), 5)
print("Test error (MAE): {}".format(score_test))
print("Test error (MSE): {}".format(mse_score))
print("Test error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
```

```
======TRAIN======
Train error (MAE): 0.24448
Train error (MSE): 0.12586
Train error (RMSE): 0.35477
R2 score : 0.93469
======TEST======
Test error (MAE): 0.66328
Test error (MSE): 0.92714
Test error (RMSE): 0.96288
R2 score : 0.52109
```

Hình 2.50: Huấn luyện mô hình

Tối ưu hóa hàm lỗi

Sau khi giảm chiều dữ liệu, bạn có thể sử dụng các kỹ thuật tối ưu hóa để tìm giá trị tối ưu cho mô hình.

Đánh giá các phương pháp tối ưu hóa:

- Gradient Descent hoặc Stochastic Gradient Descent (SGD): Phù hợp cho các mô hình lớn và không cần tính toán ma trận Hessian.
- Quasi-Newton Methods (ví dụ: L-BFGS): Thường được dùng trong các mô hình học máy như Logistic Regression và SVM.

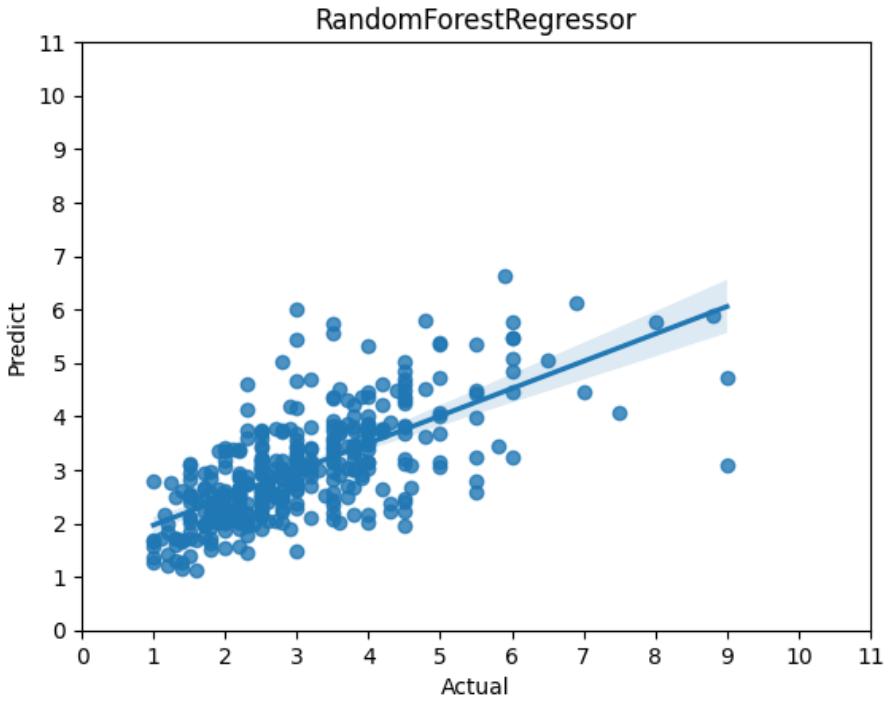
- Grid Search: Dùng Grid Search để tìm các siêu tham số tốt nhất cho mô hình trên tập dữ liệu đã giảm chiều.

Tối ưu hóa Random Forest bằng Grid Search để cải thiện hiệu suất trên dữ liệu đã giảm chiều.

```python
from sklearn.model_selection import GridSearchCV

# Định nghĩa các giá trị cho siêu tham số để tìm kiếm
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30],
    'min_samples_split': [2, 5, 10]
}

# Thiết lập GridSearchCV với Random Forest
grid_search = GridSearchCV(estimator=RandomForestRegressor(random_state=42), param_grid=param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)

# Huấn luyện Grid Search với tập train
grid_search.fit(X_train, y_train)

# Kết quả siêu tham số tốt nhất
print("Best parameters found: ", grid_search.best_params_)
# print("Best score: ", grid_search.best_score_)

# Dùng mô hình với siêu tham số tối ưu
best_model = grid_search.best_estimator_

# Đánh giá mô hình
y_pred = best_model.predict(X_test)
mae_score = mean_absolute_error(y_test, y_pred).round(5)
print("Test MAE with optimized model: ", mae_score)
```

```
Best parameters found:  {'max_depth': 20, 'min_samples_split': 10, 'n_estimators': 300}
Test MAE with optimized model:  0.65422
```

Hình 2.51: Code tối ưu hóa hàm lỗi



Hình 2.52: Tối ưu hóa hàm lỗi

Nhận xét

Ta có thể thấy kết quả của mô hình sau khi giảm chiều và tối ưu hóa hàm lỗi không thay đổi đáng kể. Điều này xảy ra là do:

- Dữ liệu đầu vào là rất tốt: không có đặc trưng nào được chỉnh sửa nhiều khi giảm chiều.
- Lựa chọn mô hình hợp lý: Random Forest tự động chọn lọc những đặc trưng quan trọng trong quá trình xây dựng cây, do đó việc giảm chiều có thể không mang lại nhiều cải thiện đáng kể.

Tên	Đóng góp
Chu Văn Mạnh	30%
Trần Việt Hoàng	30%
Trần Hoàng Hiệp	30%
Trần Vũ Cảnh	5%
Nguyễn Hải Nam	5%

Hình 2.53: Phần trăm đóng góp tuần 8

Tuần Topic 9 (01/11 - 08/11)

Chủ đề tìm hiểu tuần Topic 9: MongoDB

Chèn cơ sở dữ liệu vào MongoDB

```python
import pandas as pd
from pymongo import MongoClient

# Sử dụng URI đã cung cấp
uri = 'mongodb+srv://GROUP_1:12345@cluster0.cjavi.mongodb.net/'

# Tạo client MongoDB
client = MongoClient(uri)

# Chọn cơ sở dữ liệu và bộ sưu tập
db = client['DB_G1']
collection = db['RentalRoom']

# Đọc dữ liệu từ file CSV
df = pd.read_csv('clean_data.csv')

# Chuyển DataFrame thành danh sách các từ điển
data = df.to_dict(orient='records')

# Chèn dữ liệu vào bộ sưu tập
result = collection.insert_many(data)

# In ID của các tài liệu đã chèn
print('Inserted document IDs:', result.inserted_ids)
```

Hình 2.54: Lưu trữ dữ liệu

Giải thích: *Import các thư viện cần thiết:

- **pandas**: Thư viện dùng để xử lý và phân tích dữ liệu, trong trường hợp này để đọc file CSV.
- **MongoClient** từ **pymongo**: Được sử dụng để kết nối với cơ sở dữ liệu MongoDB.

*Khởi tạo URI kết nối MongoDB:

- Đây là chuỗi kết nối đến một cơ sở dữ liệu MongoDB từ xa. Bạn cần cung cấp tên người dùng, mật khẩu và địa chỉ của MongoDB cluster.

*Tạo client MongoDB:

- **MongoClient** sẽ sử dụng **uri** để kết nối tới MongoDB. Thông qua client, bạn có thể truy cập các database và collection trong MongoDB.

*Chọn cơ sở dữ liệu và bộ sưu tập:

- **db**: Là một đối tượng đại diện cho cơ sở dữ liệu có tên **DB_G1**.
- **collection**: Là một đối tượng đại diện cho collection (bộ sưu tập) có tên **RentalRoom** trong cơ sở dữ liệu **DB_G1**.

*Đọc dữ liệu từ file CSV:

- Đọc file CSV (**clean_data.csv**) vào một **DataFrame df**.

*Chuyển đổi DataFrame thành danh sách các từ điển:

- **to_dict(orient='records')**: Chuyển **DataFrame** thành một danh sách các từ điển. Mỗi từ điển đại diện cho một hàng trong **DataFrame**, giúp dễ dàng chèn vào MongoDB.

*Chèn dữ liệu vào MongoDB:

- **insert_many(data)**: Chèn toàn bộ danh sách **data** vào collection **RentalRoom**.

*In ID của các tài liệu đã chèn:

RentalRoom

	Title String	Price(Triệu/tháng) Double	Square(m²) Double	District String	City String
1	"Phòng 5 tr/tg. Dc 269 ng…	5	25	"Tân Bình"	"Hồ Chí Minh"
2	"Cho thuê phòng trọ dạng …	1.8	18	"Quận 8"	"Hồ Chí Minh"
3	"PHÒNG SẠCH THOÁNG, SÂN R…	1.5	12	"Tân Bình"	"Hồ Chí Minh"
4	"Cho thuê CHDV 25-30m2 nộ…	5.5	25	"Gò Vấp"	"Hồ Chí Minh"
5	"Cho thuê phòng có nội th…	5.5	30	"Quận 1"	"Hồ Chí Minh"
6	"Cho thuê phòng trọ Quận …	3.5	18	"Quận 1"	"Hồ Chí Minh"
7	"Phòng đẹp như căn hộ stu…	4.2	25	"Bình Thạnh"	"Hồ Chí Minh"
8	"Phòng siêu đẹp có ban cô…	2.9	20	"Gò Vấp"	"Hồ Chí Minh"

Hình 2.55: kết quả run code

Tên	Đóng góp
Chu Văn Mạnh	30%
Trần Việt Hoàng	30%
Trần Hoàng Hiệp	30%
Trần Vũ Cảnh	5%
Nguyễn Hải Nam	5%

Hình 2.56: Phần trăm đóng góp tuần 9

Tuần Topic 10 (08/11 - 15/11)

Chủ đề tìm hiểu tuần topic 10: Huấn luyện mô hình Random Forest, truy vấn và demo mô hình Random Forest

Huấn luyện mô hình Random Forest

```python
import pandas as pd
from pymongo import MongoClient
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Kết nối MongoDB
uri = 'mongodb+srv://GROUP_1:12345@cluster0.cjavi.mongodb.net/'
client = MongoClient(uri)
db = client['DB_G1']
collection = db['RentalRoom']

# Lấy dữ liệu từ MongoDB và chuyển thành DataFrame
data = list(collection.find())
df = pd.DataFrame(data)

# Xóa cột _id do không cần dùng trong quá trình huấn luyện
df = df.drop(columns=['_id'])

# Tách dữ liệu thành X và y
X = df.drop(columns=['Price'])  # Giả sử cột mục tiêu là 'Price'
y = df['Price']

# Chia dữ liệu thành tập huấn luyện và kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

# Huấn luyện mô hình Random Forest
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)

print("======TRAIN======")
y_pre_train = model.predict(X_train)
score_train = mean_absolute_error(y_train,y_pre_train).round(5)
mse_score = mean_squared_error(y_train,y_pre_train).round(5)
rmse = mean_squared_error(y_train,y_pre_train,squared = False).round(5)
score_r2 = round(r2_score(y_train,y_pre_train), 5)
print("Train error (MAE): {}".format(score_train))
print("Train error (MSE): {}".format(mse_score))
print("Train error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
print("======TEST======")
y_pre_test = model.predict(X_test)
score_test = mean_absolute_error(y_test,y_pre_test).round(5)
mse_score = mean_squared_error(y_test,y_pre_test).round(5)
rmse = mean_squared_error(y_test,y_pre_test,squared = False).round(5)
score_r2 = round(r2_score(y_test,y_pre_test), 5)
print("Test error (MAE): {}".format(score_test))
print("Test error (MSE): {}".format(mse_score))
print("Test error (RMSE): {}".format(rmse))
print("R2 score : {}".format(score_r2))
```

Hình 2.57: Code huấn luyện

Giải thích code

- Import các thư viện cần thiết

- `pandas`: Thư viện xử lý dữ liệu phổ biến trong Python, dùng để tạo và thao tác với `DataFrame`.
- `MongoClient` từ `pymongo`: Dùng để kết nối với cơ sở dữ liệu MongoDB.

- `RandomForestRegressor` từ `sklearn.ensemble`: Mô hình học máy sử dụng thuật toán Random Forest để hồi quy, giúp dự đoán giá trị mục tiêu (ở đây là giá phòng cho thuê).
- `train_test_split` từ `sklearn.model_selection`: Hàm dùng để chia dữ liệu thành tập huấn luyện và tập kiểm tra.
- Các hàm đánh giá từ `sklearn.metrics`:
  - `mean_absolute_error (MAE)`: Đo độ chênh lệch trung bình giữa giá trị dự đoán và giá trị thực tế.
  - `mean_squared_error (MSE)`: Tính độ lệch bình phương trung bình giữa giá trị dự đoán và thực tế.
  - `r2_score`: Hệ số xác định R^2, biểu thị mức độ phù hợp của mô hình với dữ liệu (từ 0 đến 1, càng gần 1 càng tốt).

- Kết nối với MongoDB

- `uri`: Đường dẫn kết nối đến MongoDB (thường bao gồm nhóm, cụm, và thông tin đăng nhập nếu có).
- `client`: Tạo một kết nối đến MongoDB từ `uri`.
- `db`: Lấy đối tượng cơ sở dữ liệu có tên là `DB_G1`.
- `collection`: Truy cập vào collection `RentalRoom` trong `DB_G1`, nơi chứa dữ liệu phòng cho thuê.

- Truy xuất dữ liệu từ MongoDB

- `collection.find()`: Lấy toàn bộ dữ liệu từ collection `RentalRoom`.
- `data = list(collection.find())`: Dữ liệu được chuyển thành danh sách, mỗi phần tử là một dòng dữ liệu từ MongoDB.
- `df = pd.DataFrame(data)`: Chuyển danh sách `data` thành một `DataFrame df` của pandas để dễ dàng xử lý.

- Xóa cột _id

- `df.drop(columns=['_id'])`: Xóa cột `\id`, vì cột này là định danh tự động từ MongoDB, không có ý nghĩa trong việc huấn luyện mô hình.

- Tách dữ liệu thành hai phần X và y

- `X = df.drop(columns=['Price'])`: `X` chứa tất cả các cột trừ cột `Price`, đại diện cho các đặc trưng (features) trong mô hình.
- `y = df['Price']`: `y` là biến mục tiêu (target), chứa giá phòng (Price) mà chúng ta muốn dự đoán.

- Chia dữ liệu thành tập huấn luyện và tập kiểm tra

- `train_test_split(X, y, test_size=0.25, random_state=42)`: Chia dữ liệu `X` và `y` thành hai phần:
  - `X_train` và `y_train`: Dữ liệu huấn luyện (75%).

  - `X_test` và `y_test`: Dữ liệu kiểm tra (25%).

- `random_state=42`: Đảm bảo việc chia dữ liệu là ngẫu nhiên nhưng có thể tái tạo được khi chạy lại mã.

- Khởi tạo và huấn luyện mô hình Random Forest

- `model = RandomForestRegressor(random_state=42)`: Tạo mô hình hồi quy Random Forest với `random_state=42` để đảm bảo tính tái lập.
- `model.fit(X_train, y_train)`: Huấn luyện mô hình trên tập huấn luyện `X_train` và `y_train`.

- Đánh giá mô hình trên tập huấn luyện

- `y_pre_train = model.predict(X_train)`: Dự đoán `y` trên tập huấn luyện.
- Tính toán các chỉ số đánh giá:
  - MAE (Mean Absolute Error): Sai số trung bình giữa giá trị thực tế và dự đoán.
  - MSE (Mean Squared Error): Bình phương của sai số trung bình.
  - RMSE (Root Mean Squared Error): Căn bậc hai của MSE, thường dễ hiểu hơn vì cùng đơn vị với biến mục tiêu.
  - R2 Score: Hệ số xác định, giá trị càng gần 1 càng cho thấy mô hình phù hợp với dữ liệu.
- Kết quả của từng chỉ số được in ra để phân tích.

- Đánh giá mô hình trên tập kiểm tra

- Tương tự như tập huấn luyện, mô hình sẽ được đánh giá trên tập kiểm tra `(X_test, y_test)` bằng các chỉ số MAE, MSE, RMSE, và R2 Score.
- Việc đánh giá trên tập kiểm tra giúp kiểm tra độ tổng quát của mô hình. Nếu các chỉ số này trên tập kiểm tra khác biệt lớn so với tập huấn luyện, có thể mô hình bị overfitting (quá khớp với dữ liệu huấn luyện) hoặc underfitting (chưa khớp đủ với dữ liệu huấn luyện).

Truy vấn dữ liệu

- Tìm Các Phòng Trọ Ở Một Quận/Huyện Cụ Thể

```
from pymongo import MongoClient
client = MongoClient('mongodb+srv://GROUP_1:12345@cluster0.cjavi.mongodb.net/')
db = client['DB_G1']
collection = db['RentalRoom']
query = {"District_Quận 1": True}
results = collection.find(query)
for room in results:
    print(room)
```

- Tìm Các Phòng Trọ Với Diện Tích Trong Khoảng 20m² - 40m²

```
from pymongo import MongoClient
client = MongoClient('mongodb+srv://GROUP_1:12345@cluster0.cjavi.mongodb.net/')
```

```
db = client['DB_G1']
collection = db['RentalRoom']
query = {"Square": {"$gte": 20, "$lte": 40}}
results = collection.find(query)
for room in results:
    print(room)
```

Tìm Các Phòng Trọ Gần Trung Tâm Và Có Nội Thất

```
from pymongo import MongoClient
client = MongoClient('mongodb+srv://GROUP_1:12345@cluster0.cjavi.mongodb.net/')
db = client['DB_G1']
collection = db['RentalRoom']

query = {
    "isNearCenter_yes": True,
    "hasFurniture_yes": True
}
results = collection.find(query)
for room in results:
    print(room)
```

- Đếm Số Lượng Phòng Trọ Trong Mỗi Quận/Huyện

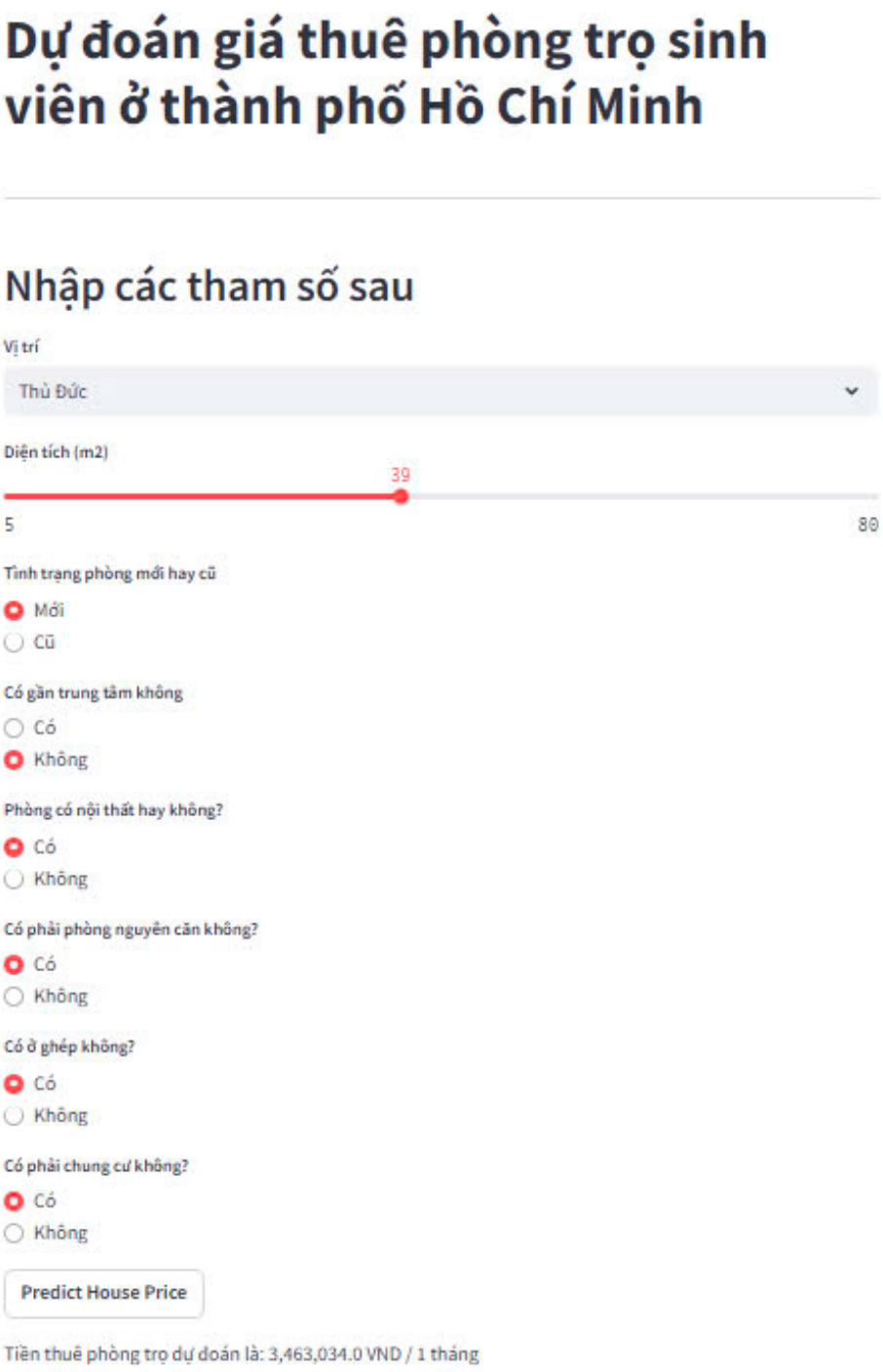
```
from pymongo import MongoClient
client = MongoClient('mongodb+srv://GROUP_1:12345@cluster0.cjavi.mongodb.net/')
db = client['DB_G1']
collection = db['RentalRoom']
pipeline = [
    {"$group": {"_id": "District_Quận 1", "count": {"$sum": 1}}},
    {"$sort": {"count": -1}}
]
results = collection.aggregate(pipeline)
for result in results:
    print(f"Quận/Huyện: {result['_id']}, Số lượng phòng trọ: {result['count']}")
```

- Tính Giá Thuê Trung Bình Theo Quận/Huyện

```
from pymongo import MongoClient
client = MongoClient('mongodb+srv://GROUP_1:12345@cluster0.cjavi.mongodb.net/')
db = client['DB_G1']
collection = db['RentalRoom']
pipeline = [
    {"$group": {"_id": "District_Quận 1", "average_price": {"$avg": "$Price"}}},
    {"$sort": {"average_price": -1}}
```

```
]
results = collection.aggregate(pipeline)
for result in results:
    print(f"Quận/Huyện: {result['_id']}, Giá thuê trung bình: {result['average_price']
```

Demo mô hình Random Forest



Hình 2.58: Demo mô hình Random Forest

CHƯƠNG 3. KẾT LUẬN

3.1 Kết luận

Dự án "Phân tích và dự đoán giá thuê phòng trọ ở TP Hồ Chí Minh"đã cho phép ứng dụng các kiến thức về khoa học dữ liệu và học máy vào một bài toán có ý nghĩa thực tiễn trong bối cảnh đô thị hóa và nhu cầu nhà ở ngày càng gia tăng. Trong quá trình thực hiện dự án, nhóm đã triển khai thành công việc thu thập dữ liệu từ các nguồn trực tuyến, làm sạch, xử lý dữ liệu và áp dụng nhiều mô hình dự đoán khác nhau để xác định giá thuê phòng trọ dựa trên các yếu tố như vị trí, diện tích và thời gian.

So với các nghiên cứu hoặc công cụ tương tự, dự án tập trung vào việc khai thác chi tiết các yếu tố địa lý và đặc trưng cụ thể của thị trường phòng trọ tại TP Hồ Chí Minh, đồng thời sử dụng các thuật toán học máy tiên tiến như Random Forest, hồi quy tuyến tính và ARIMA cho phân tích chuỗi thời gian. Kết quả dự báo đạt độ chính xác trung bình, phản ánh được xu hướng biến động của thị trường. Tuy nhiên, mô hình dự báo vẫn còn hạn chế trong việc xử lý dữ liệu có tính biến động cao và có thể cải thiện thêm độ chính xác.

Trong quá trình thực hiện dự án, nhóm đã đạt được các thành tựu như:

- Hoàn thiện quy trình thu thập và xử lý dữ liệu để làm sạch và chuẩn hóa, giúp cải thiện đầu vào cho các mô hình dự đoán. Áp dụng thành công nhiều mô hình học máy khác nhau và đánh giá hiệu quả của từng mô hình. Đưa ra các chỉ số và biểu đồ trực quan, giúp người dùng hiểu rõ hơn về xu hướng giá thuê tại từng khu vực. - Bên cạnh đó, cũng gặp một số hạn chế, chẳng hạn như độ chính xác của mô hình bị ảnh hưởng bởi dữ liệu thiếu nhất quán và các yếu tố khách quan không thể kiểm soát, chẳng hạn như tác động của chính sách nhà ở hoặc biến động kinh tế. Qua dự án, nhóm đã rút ra những bài học quý giá về việc xử lý và tối ưu hóa dữ liệu, phân tích mô hình và quan trọng hơn là kỹ năng làm việc nhóm và quản lý tiến độ.

3.2 Hướng phát triển

Để hoàn thiện và mở rộng dự án, nhóm định hướng phát triển theo hai khía cạnh: hoàn thiện chức năng đã có và mở rộng với các tính năng mới.

Trước hết, để tăng độ chính xác của mô hình, cần cải tiến các bước xử lý dữ liệu bằng cách thu thập thêm dữ liệu từ nhiều nguồn hơn và cập nhật dữ liệu theo thời gian thực để phản ánh chính xác hơn tình hình thị trường. Việc tối ưu hóa tham số của các mô hình hiện tại và thử nghiệm với các thuật toán học máy tiên tiến hơn như mạng nơ-ron đa tầng hoặc mô hình boosting cũng sẽ giúp nâng cao hiệu quả dự đoán.

Ngoài ra, có thể triển khai các tính năng bổ sung như tích hợp yếu tố kinh tế-xã hội (GDP, tỷ lệ lạm phát) vào mô hình để phân tích sâu hơn các yếu tố tác động. Xây dựng một giao diện trực tuyến cho phép người dùng tương tác, tra cứu và so sánh giá thuê phòng trọ theo từng khu vực cũng là một hướng đi tiềm năng, giúp mở rộng phạm vi ứng dụng của dự án từ một công cụ nghiên cứu thành một sản phẩm hỗ trợ người dùng thực tiễn.

TÀI LIỆU THAM KHẢO

- E. H. Hovy, “Automated discourse generation using discourse structure relations,” *Artificial Intelligence*, vol. 63, no. 1-2, pp. 341–385, 1993.
- L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*, Elsevier, 2007.
- Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow".
- M. Poesio and B. Di Eugenio, “Discourse structure and anaphoric accessibility,” in *ESSLLI Workshop on Information Structure, Discourse Structure and Discourse Semantics*, Copenhagen, Denmark, 2001, pp. 129–143.
- A. Knott, “A data-driven methodology for motivating a set of coherence relations,” Ph.D. dissertation, The University of Edinburgh, UK, 1996.
- Available: `https://machinelearningcoban.com/` (visited on Dec 28, 2016).