

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Applying MLOps for Enhanced AI Prediction in
Real Estate

Lê Thành Long
long.lt194099@sis.hust.edu.vn

Ngành: Khoa học máy tính

Giảng viên hướng dẫn: TS. Trần Văn Đặng

Chữ ký GVHD

Khóa: 64

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2024

LỜI CẢM ƠN

Em xin bày tỏ cảm ơn sâu sắc đến thầy Tiến sĩ Trần Văn Đặng và sự hướng dẫn tỉ mỉ của thầy không những ở khía cạnh kiến thức chuyên môn mà còn các kỹ năng mềm trong quá trình làm đồ án tại Trường Công nghệ thông tin và Truyền Thông, Đại Học Bách Khoa Hà Nội. Bên cạnh em xin cảm ơn các thầy cô như những người lái đò giúp bản thân em phát triển mạnh mẽ hơn và có được những kiến thức cho chặng đường sắp tới. Em đã có được những sự trưởng thành, mạnh mẽ và chững chạc hơn và sự quyết tâm hơn cho giai đoạn tiếp theo của sự nghiệp.

Em xin cảm ơn bạn bè, người thân, những vấp ngã trong các cuộc thi đọ xát ở Đại học Bách Khoa Hà Nội đã khiến em phát triển hoàn thiện hơn. Những sự ủng hộ động viên không ngừng nghỉ đó đã giúp em biết mình phải làm gì, chuẩn bị và sắp xếp thời gian của mình sao cho hợp lý. Những kiến thức không chỉ nằm vón vẹn trên ghế nhà trường mà còn là những kinh nghiệm quý báu trong cuộc sống em đã được những thầy cô tâm huyết chỉ bảo. Đó là những món quà mà em rất hân diện có được cho đến tận thời điểm này, em đã chuẩn bị hành trang đầy đủ cho sau này.

Cuối cùng em xin cảm ơn ngôi nhà chung Đại Học Bách Khoa Hà Nội nói chung và ngôi nhà riêng khoa học máy tính 06 đã là nơi để em sẽ trở về và ôn lại những kỷ niệm, những bài thi, những cái ôm, những cái tên mà em sẽ nhớ mãi.

Lời Cam Đoan

Họ và tên sinh viên: Lê Thành Long

MSSV: 20194099

Điện thoại liên lạc: 0969973012

Email: long.lt194099@sis.hust.edu.vn

Lớp: IT1-06-K64

Chương trình đào tạo: Khoa Học Máy Tính

Tôi - Lê Thành Long - cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của TS. Trần Văn Đặng. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày tháng năm

Tác giả ĐATN

(Chữ ký và tên đầy đủ)

TÓM TẮT NỘI DUNG ĐỒ ÁN

Việc phát triển và triển khai các mô hình AI trong thế giới thực đặt ra những thách thức đáng kể do hạn chế về thời gian và nguồn lực. Độ tin cậy là tối quan trọng, đòi hỏi độ trễ thấp, thời gian hoạt động cao và khả năng chịu lỗi mạnh mẽ. Ngoài ra, việc cập nhật liên tục với kiến thức mới và triển khai tự động là rất quan trọng để duy trì độ chính xác dự đoán với dữ liệu đang phát triển. Các quy trình tích hợp truyền thống thường thiếu sự gắn kết, bao gồm các giai đoạn rời rạc từ chuẩn bị dữ liệu đến giám sát hiệu quả mô hình và tự động hóa. Sự xuất hiện gần đây của MLOps (Quy trình học máy) cung cấp một phương pháp tiếp cận chuẩn hóa cho vòng đời học máy, giải quyết những hạn chế này. Trong nghiên cứu này, tôi sẽ đề xuất một quy trình MLOPS để nâng cao tính tự động hóa và hiệu quả của hệ thống AI. Cùng với quy trình MLOps, tôi tập trung giải quyết vấn đề dự đoán giá bất động sản tại thị trường Việt Nam.

Gần đây, vấn đề giá bất động sản được quan tâm nhiều hơn khi có nhiều hiện tượng lạm phát giá. Do đó, định giá bất động sản đã trở thành một trong những câu hỏi mà người dùng quan tâm. Nhận thấy tiềm năng của thị trường, nhiều chuyên gia trong lĩnh vực thống kê và trí tuệ nhân tạo đã tích cực tham gia và đề xuất vô số giải pháp cho các nền tảng tin tức và đơn vị cung cấp dịch vụ bất động sản. Dữ liệu bất động sản được cập nhật hàng ngày đòi hỏi các dịch vụ trí tuệ nhân tạo cũng phải được cập nhật tương ứng. Nhiều dịch vụ AI trong lĩnh vực bất động sản vẫn chưa nhận ra tầm quan trọng của vấn đề này và đã ảnh hưởng đến trải nghiệm của người dùng về độ tin cậy của dự đoán giá bất động sản. Ngoài ra, quá trình xây dựng và triển khai các mô hình AI trong nhiều sản phẩm không đảm bảo tính toàn vẹn và độ tin cậy và ảnh hưởng trực tiếp đến người dùng cuối như: các mô hình triển khai được người xây dựng mô hình đánh giá chủ quan, quá trình đào tạo không được giám sát đúng cách. Do đó, việc các mô hình AI đưa ra kết quả dự đoán không đáng tin cậy, ảnh hưởng trực tiếp đến người dùng là điều không thể tránh khỏi. Hơn nữa, việc cập nhật kiến thức về giá bất động sản ở thời điểm hiện tại là cần thiết trong việc định giá nhà cho người dùng và đơn vị cung cấp dịch vụ.

Để giải quyết những thách thức này, nghiên cứu này đề xuất một quy trình tự động bao gồm chuẩn bị dữ liệu, phát triển mô hình, xử lý hậu kỳ, đánh giá tự động và giám sát. Quy trình này nhằm: (i) tăng cường tự động hóa quy trình trong tích hợp mô hình AI trong quá trình phát triển phần mềm mà không ảnh hưởng đến trải nghiệm của người dùng, (ii) triển khai các cơ chế giám sát và đánh giá hiệu quả trong suốt vòng đời sản phẩm, (ii) đảm bảo độ tin cậy và tính toàn vẹn của hệ thống

trong các ứng dụng thực tế. Bằng cách tận dụng các nguyên tắc MLOps, nghiên cứu này tìm cách tối ưu hóa các hệ thống dự đoán bất động sản do AI thúc đẩy, điều chỉnh chúng theo động lực thị trường đang thay đổi và kỳ vọng của người dùng. Cuối cùng nghiên cứu xây dựng hệ thống BKPrice thành công và chứng thực bằng các thông số thực nghiệm và hoàn thành tốt những yêu cầu đã đặt ra.

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

ABSTRACT

The development and deployment of real-world AI models pose significant challenges due to time and resource constraints. Reliability is paramount, demanding low latency, high uptime, and robust fault tolerance. Additionally, continuous updates with new knowledge and automated deployments are crucial for maintaining predictive accuracy with evolving data. Traditional integration processes often lack cohesion, encompassing fragmented stages from data preparation to model effectiveness monitoring and automation. The recent emergence of MLOps (Machine Learning Operations) offers a standardized approach to the machine learning lifecycle, addressing these limitations. In this research, I will propose an MLOps process to improve the automation and efficiency of the AI system. Together with the MLOps process, I focus on solving the problem of real estate quotation in the Vietnamese market.

Referencing this standard process with existing solutions, we can immediately see the major shortcomings in manual AI application systems. In this study, I will propose an MLOPS process to improve the automation and efficiency of the AI system. Along with the MLOps process, I focus on solving the problem of predicting real estate prices in the Vietnamese market. Recently, the issue of real estate prices has received more attention when there have been many price inflation phenomena. Therefore, real estate valuation has become one of the questions that users are interested in. Realizing the potential of the market, many experts in the fields of statistics and artificial intelligence have actively participated and proposed countless solutions for news platforms and real estate service providers. Real estate data is updated daily requires artificial intelligence services to be updated accordingly. Many AI services in the real estate sector have not yet recognized the importance of this issue and have affected the user experience in terms of the reliability of real estate price predictions. In addition, the process of building and deploying AI models in many products does not ensure integrity and reliability and directly affects end users such as: deployed models are subjectively evaluated by the model builder, the training process is not properly monitored. Therefore, it is inevitable that AI models give unreliable prediction results, directly affecting users. Moreover, updating knowledge about real estate prices at the present time is necessary in pricing houses for users and service providers. To address these challenges, this study proposes an automated process encompassing data preparation, model development, post-processing, automatic evaluation, and monitoring. This MLOps-driven approach aims to: (i) increase process automation in AI model integration within software

development without compromising user experience, (ii) implement effective monitoring and evaluation mechanisms throughout the product lifecycle, (ii) ensure system reliability and integrity in real-world applications. By leveraging MLOps principles, this research seeks to optimize AI-driven real estate prediction systems, aligning them with evolving market dynamics and user expectations.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	3
1.3 Định hướng giải pháp.....	4
1.4 Đóng góp của đồ án	5
1.5 Bố cục của đồ án	6
1.6 Các giải pháp hiện tại và hạn chế	7
1.6.1 Xây dựng mô hình không tin cậy	7
1.6.2 Quy trình xây dựng mô hình thủ công	8
1.7 Mục tiêu và định hướng giải pháp	8
1.7.1 Mục tiêu.....	8
1.7.2 Định hướng giải pháp	8
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH TỔNG QUAN.....	10
2.1 Khảo sát	10
2.2 Phân tích chức năng tổng quan hệ thống	12
2.2.1 Tổng quan hệ thống	12
2.2.2 Các thành phần chính của hệ thống	13
CHƯƠNG 3. CƠ SỞ LÝ THUYẾT	15
3.1 Giải thuật và tính hiệu quả của hệ thống.....	15
3.2 Framework và tính tự động của hệ thống trong lưu trữ và xây dựng mô hình	22
CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT.....	30
4.1 Tổng quan giải pháp.....	30

4.2 Hệ thống thu thập dữ liệu tự động	30
4.2.1 Data Components.....	31
4.2.2 Data Pipeline	32
4.2.3 Xử lý và lưu trữ dữ liệu	35
4.3 Hệ thống dự đoán giá bất động sản tin cậy	37
4.3.1 Giải thuật định giá bất động sản	37
4.3.2 Hệ thống huấn luyện và triển khai mô hình tự động	48
4.3.3 Triển khai quá trình MLOPs bằng mô hình ngôn lớn	57
CHƯƠNG 5. ĐÁNH GIÁ VÀ THỰC NGHIỆM	60
5.1 Các tham số đánh giá	60
5.1.1 Tập tham số đánh giá hiệu năng mô hình dự đoán	60
5.1.2 Tập tham số đánh giá quy trình MLOps.....	61
5.2 Phương pháp thí nghiệm.....	61
5.2.1 Cấu hình thiết bị sử dụng	61
5.2.2 Môi trường lập trình thử nghiệm.....	61
5.2.3 Cấu hình siêu tham số	62
5.2.4 Cấu hình tập thuộc tính.....	63
5.2.5 Tiền hành thí nghiệm.....	64
CHƯƠNG 6. GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	77
6.0.1 Tính mới và tính sáng tạo.....	77
6.0.2 Tính module hóa và tái sử dụng	77
CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI	79
7.1 Kết luận	79
7.2 Hướng phát triển trong tương lai	80

CHƯƠNG 8. PHỤ LỤC.....	82
8.0.1 Các trường hợp dự đoán giá bất động sản trong thực tế	82
8.0.2 Quản lý và đánh giá mô hình.....	84
8.0.3 Cơ sở dữ liệu	86
8.0.4 Biểu đồ benchmark kết quả	87
REFERENCE	92

DANH MỤC HÌNH VẼ

Hình 1.1	Lãi suất vay ngân hàng tháng 5/2024	1
Hình 1.2	Mức độ quan tâm bất động sản	1
Hình 2.1	Màn hình định giá nhà đất của biggee	10
Hình 2.2	Màn hình định giá ngôi nhà ở OneHousing	11
Hình 2.3	Kết quả định giá ở OneHousing	12
Hình 2.4	Sơ đồ hoạt động tổng quan của hệ thống	13
Hình 3.1	AI Model Benchmark	16
Hình 3.2	Mô tả cấu trúc dữ liệu Quadtree	21
Hình 3.3	Feast Framework	23
Hình 3.4	Kiến trúc của MLflow	24
Hình 3.5	Airflow architecture	25
Hình 3.6	Sự khác nhau giữa coupled và decoupled architecture	27
Hình 3.7	Debezium Architecture	28
Hình 4.1	BKPrice Data System	30
Hình 4.2	Dữ liệu từ trang muaban.net	33
Hình 4.3	Dữ liệu từ trang meyland.com	33
Hình 4.4	Dữ liệu từ trang batdongsan.com	34
Hình 4.5	Thông tin đường phố Việt Nam	36
Hình 4.6	Định dạng cuối cùng của dữ liệu	37
Hình 4.7	Danh sách các đặc trưng sử dụng trong mô hình	38
Hình 4.8	Danh sách các đặc trưng sử dụng trong mô hình	39
Hình 4.9	BKPrice Prediction Algorithm	40
Hình 4.10	Danh sách các địa điểm nổi tiếng	42
Hình 4.11	Mô hình hóa mật độ của thuộc tính <i>num_of_school_in_1000m_radius</i> trước và sau khi khớp Gaussian Mixture Model	43
Hình 4.12	Mô hình hóa mật độ của thuộc tính <i>num_of_marketplace_in_1000m_radius</i> trước và sau khi khớp Gaussian Mixture Model	44
Hình 4.13	Biểu đồ đơn và tích lũy phương sai của các chiều dữ liệu	44
Hình 4.14	BKPrice Model	46
Hình 4.15	Giá khu vực quận 11 tuân theo phân bố gaussian	47
Hình 4.16	Phân bố giá dự đoán chưa khớp với phân bố giá thực tế	47
Hình 4.17	Phân bố giá dự đoán đã gần khớp với phân bố giá thực tế	48
Hình 4.18	BKPrice Prediction System	49

Hình 4.19 Tiếp nhận và huấn luyện mô hình tự động	50
Hình 4.20 Màn hình quản lý tập huấn luyện và các đặc trưng tương ứng .	51
Hình 4.21 Giao diện chính quản lý các tác vụ huấn luyện mô hình	52
Hình 4.22 Giao diện quản lý các tác trong mỗi DAG	53
Hình 4.23 Màn hình quản lý thời các thời điểm huấn luyện mô hình	55
Hình 4.24 Thông tin của một thời điểm huấn luyện mô hình XGB trên bộ dữ liệu thành phố Hồ Chí Minh	55
 Hình 5.1 Benchmark Featureset and Model Performance on Ha Noi Dataset	64
Hình 5.2 Benchmark Featureset Version and Model Performance on Ho Chi Minh Dataset	64
Hình 5.3 Độ quan trọng của thuộc tính	65
Hình 5.4 Thống kê trung bình thời gian huấn luyện mô hình và tỷ lệ thành công khi huấn luyện	68
Hình 5.5 Trạng thái dung lượng bộ nhớ sử dụng và mức độ sử dụng CPU%	69
Hình 5.6 Mô tả thời gian hoàn thành giai đoạn thu thập, làm sạch lưu trữ và xây dựng tập huấn luyện - Phrase 1	70
Hình 5.7 Mô tả thời gian hoàn thành giai đoạn huấn luyện và đánh giá mô hình - Phrase 2	70
Hình 5.8 Mô tả tổng thời gian hoàn thành toàn bộ luồng MLOps	71
Hình 5.9 Chat Phrase 1	72
Hình 5.10 Chat Phrase 2	72
Hình 5.12 Chat Phrase 4	73
Hình 5.11 Chat Phrase 3	73
Hình 5.13 Chat Phrase 5	74
Hình 5.14 Chat Phrase 6	74
Hình 5.17 Chat Phrase 9	75
Hình 5.15 Chat Phrase 7	75
Hình 5.16 Chat Phrase 8	75
 Hình 8.1 Test Case 1	82
Hình 8.2 Test Case 2	82
Hình 8.3 Test Case 3	83
Hình 8.4 Test Case 4	83
Hình 8.5 Test Case 5	84
Hình 8.6 Thông tin huấn luyện và đánh giá mô hình	84

Hình 8.7	Biểu đồ độ lỗi dự đoán trích xuất tự động	85
Hình 8.8	Thông tin mô hình sau khi huấn luyện kết thúc	85
Hình 8.9	Cơ sở dữ liệu bất động sản	86
Hình 8.10	Cơ sở dữ liệu bất động sản	86
Hình 8.11	HN Data - RMSE	87
Hình 8.12	HN Data - Explained Variance Score	87
Hình 8.13	HN Data - Max Error	88
Hình 8.14	HCM Data - RMSE	88
Hình 8.15	HCM Data - Explained Variance Score	89
Hình 8.16	HCM Data - Max Error	89

DANH MỤC BẢNG BIỂU

Bảng 4.1	Danh sách tiện tích công	41
Bảng 4.2	Danh sách mô hình cơ sở trong BKPrice Prediction System . .	45
Bảng 4.3	Function Calling List	59
Bảng 5.1	Cấu hình thiết bị sử dụng	61
Bảng 5.2	Cấu hình môi trường trong hệ thống BKPrice	62
Bảng 5.3	Cấu hình siêu tham số của mô hình AI	62
Bảng 5.4	Cấu hình phiên bản thuộc tính	63
Bảng 5.5	Độ lỗi RMSE trên tập huấn luyện Hà Nội	67
Bảng 5.6	Độ lỗi RMSE trên tập huấn luyện Hồ Chí Minh	67

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Application Programming Interface
CPU	Central Processing Unit
GMM	Gaussian mixture model
GPU	Graphic Processing Unit
MLOps	Machine learning operations
MLP	Multilayer perceptron
PCA	Principal component analysis
RAM	Random Access Memory

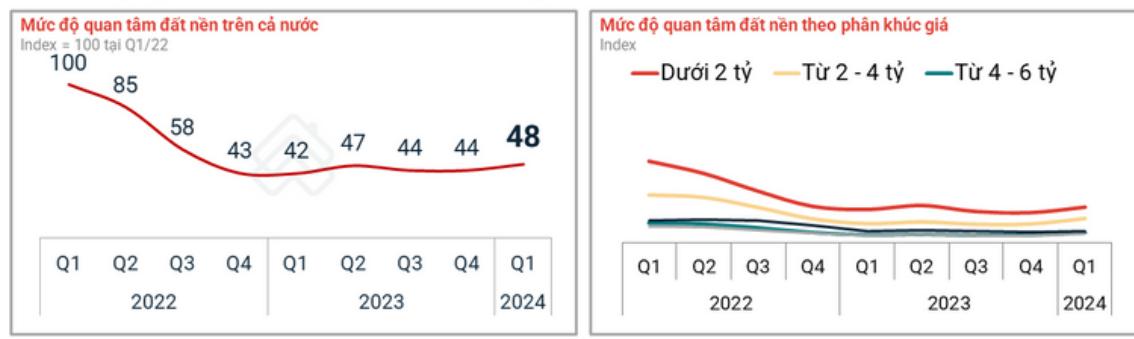
CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Ngân hàng	Lãi suất ưu đãi (%/năm)	Tỷ lệ cho vay tối đa (%)	Kỳ hạn vay tối đa (năm)	Biên độ (%)	Phí phạt trả nợ trước hạn (%)
BIDV	5-5,5	100	30	3,7	1
Vietinbank	5,8	80	20	3,5	2
Vietcombank	6,4	70	20	3,5	1-2
Agribank	6,5	100	5	3	1-4
Woori Bank	5,1-5,7	80	30	3,8	0-3
Shinhan Bank	5,5-6	70	30	0,3	0,5-2
BVBank	5-7,5	75	20	2	
Hong Leong Bank	5,5-7,5	80	25	1,5	3
SHB	5,79	75	25		
VIB	5,9	75	30	3,9	2,5
VPBank	5,9	75	25	3	4
Standard Chartered	6	75	25		0-2
UOB	6	75	25	3	0,75
GPBank	6	70	15		
Sacombank	6,5	100	30	3,5	2-5
MSB	6,5	90	35	3,5	0-3
SeABank	6,5	90	25	3,35	
HDBank	6,8	85	25	4,5	
TPBank	7,5	100	30	3,3-3,6	1-3,5

Hình 1.1: Lãi suất vay ngân hàng tháng 5/2024

Mức độ quan tâm đất nền cải thiện



Hình 1.2: Mức độ quan tâm bất động sản

Trong thị trường bất động sản hiện nay ở Việt Nam, giá nhà đất đang bị thổi giá một cách bất thường đầu năm 2024 và mối quan tâm về bất động sản đã tăng trở lại từ năm 2024, trong khi mặc dù lãi suất đang chứng kiến một xu hướng giảm ở đầu năm 2024. Hình 1.1 mô tả được lãi suất vay ở các ngân hàng đang ở mức thấp. Vào tháng tháng 5/2024 lãi suất cho vay của Vietinbank là 5-5.5 phần trăm giảm khoảng 20 phần trăm so với con số lãi suất 6.4 phần trăm vào đầu năm 2024. Các ngân hàng đang mong muốn cho người dân vay nhiều hơn. Hình 1.2 mô tả được sự quan tâm bất động sản bắt đầu dấu hiệu hồi phục từ đầu năm 2024 sau khi lao dốc từ năm 2022 và 2023. Điều này khảng định rõ ràng rằng nhu cầu mua bất động sản của người dùng [1]. Mặc dù có tương đối nhiều dịch vụ tư vấn và thẩm định bất động sản và có nhiều kênh mạng xã hội giúp người dùng có cách nhìn về định giá bất động sản hơn, nhưng đối diện với hành vi thổi giá như vậy, nếu việc định giá không được diễn ra nghiêm ngặt và đúng thì có thể dẫn tới việc người dân khó khăn trong việc xuống tiền mua bất động sản, mua không đúng phân khúc giá của miếng đất.

Việc xây dựng một hệ thống tính toán và định giá bất động sản theo thời gian rất là cần thiết. Việc tự động hóa và xây dựng mô hình định giá bất động sản mang lại cho người dân và các doanh nghiệp cung cấp dịch vụ bất động sản lợi ích lớn. Thông thường các doanh nghiệp hay các bên thứ 3 cung cấp cách dịch vụ bất động sản sẽ có bộ chuyên viên tư vấn và thẩm định bất động sản. Tuy nhiên để có một đội ngũ có chuyên môn về bất động sản, cập nhật dữ liệu thường xuyên thì mất công sức và tiền bạc, và có đủ thông tin để đưa ra được một sự tư vấn đáng và mang tính khách quan cho người dùng. Bên cạnh đó việc tự động hóa mô hình tính toán và dự đoán giá bất động sản cung cấp cho người dùng góc nhìn tổng thể và khách quan hơn về vị trí bất động sản mà người dùng quan tâm và doanh nghiệp cũng tiết kiệm được chi phí để xây dựng đội ngũ chuyên viên tư vấn bất động sản hùng hậu.

Nhìn xa hơn nữa, việc áp dụng tự động hóa định giá bất động sản người dùng có thể tìm hiểu được nhiều hơn trước đi đưa ra quyết định xuống tiền với bất động sản nào. Phía doanh nghiệp cũng nhẹ nhàng và có thể dành thời gian chăm sóc khách hàng ở khía cạnh khác nhiều hơn mà không cần phải trả lời các câu hỏi: 'Nhà này trị giá bao nhiêu" và gia tăng được hiểu qua công việc.

Trí tuệ nhân tạo là một trong những lĩnh vực được nhiều người biết đến hơn sau khi GPT được ra đời. Bên cạnh đó Machine Learning Operations (MLOps) là một lĩnh vực mới đặc biệt ở thị trường Việt Nam. MLOps có vai trò tự động hóa quản lý các dịch vụ AI tốt hơn giúp cho các sản phẩm chạy thực tế một cách mượt mà hơn. Do đó, việc có thể tự động hóa hệ thống tính toán và định giá bất động sản thì

áp dụng kết hợp cả trí tuệ nhân tạo và MLOps mang nhiều lợi ích. Trí tuệ nhân tạo giúp cho máy tính có thể phân tích và xử lý được dữ liệu bất động sản tốt và chính xác hơn. MLOps giúp cho quá trình xây dựng hệ thống định giá một cách tự động theo thời gian trôi nên mượt mà và đáp ứng được nhiều người dùng tốt hơn. Điều này quan trọng khi mà đầu năm 2024 mối quan tâm bất động sản của người dùng tăng trở lại và dự đoán tăng tiếp trong năm 2024. Một hệ thống tự động cập nhật dữ liệu bất động sản, tính toán và định giá bất động giúp doanh nghiệp có một sản phẩm mượt mà hơn và làm hài lòng trải nghiệm người dùng. Sự kết hợp giữa trí tuệ nhân tạo và MLOPs mang lại tiềm năng cho một tương lai phát triển các sản phẩm thông minh về bất động sản.

Vì vậy, tôi quyết định chọn đề tài "Áp dụng MLOPs để nâng cao dịch vụ dự đoán giá bất động sản"

1.2 Mục tiêu và phạm vi đề tài

Để giải quyết vấn đề tự động hóa trong việc định giá bất động sản thì các doanh nghiệp cung cấp dịch vụ định giá bất động sản một cách tự động mà không phụ thuộc hoàn toàn một cách thủ công vào các chuyên gia tư vấn và thẩm định giá bất động sản. Công ty cổ phần Biggee đã đưa ra một giải pháp đáng chú ý là xây dựng dịch vụ dự đoán giá bất động sản - một tính năng được tích hợp vào trang <https://biggee.vn/>. Tính năng này cho phép người dùng định giá nhanh khu vực bất động sản và định giá theo thông tin cơ bản của bất động sản [2]. Tuy nhiên cá nhân tôi thấy thì việc định giá theo thông tin cơ bản thì bất động sản yêu cầu nhập nhiều thông tin mà rất có thể người dùng khó có thể biết được như là: tờ thửa và số thửa của bất động sản. Điều này rất làm mất trải nghiệm người dùng vì thông tin tờ thửa và số thửa được công bố ở Việt Nam rất ít. Dẫn tới việc khó khăn đối với người dùng mới và có ít kiến thức chuyên môn về tờ thửa.

Bên cạnh đó có một giải pháp tới từ OneMount được tài trợ bởi VinGroup và Techcombank [3]. Giải pháp OneHousing cho phép người dùng định giá bất động sản dựa vào địa chỉ do người dùng nhập vào. Giải pháp này tiện lợi cho người dùng ở khía cạnh nhập ít thông tin hơn để có thể ra được kết quả định giá. Tuy nhiên nhược điểm lớn nhất mà tôi có thể thấy là người dùng rất khó có thể tìm được thông tin định giá của các ngôi nhà tại một vùng với các thông tin cơ bản khác nhau: nhà rộng 5m hoặc 10m thì trang OneHousing sẽ cho ra kết quả giống nhau.

Hơn thế nữa nhược điểm lớn nhất ở 2 giải pháp trên đó chính là kết quả không thay đổi theo thời gian. Sở dĩ điều này không đúng vì giá bất động sản từ đầu năm 2024 tới giữa tháng 6/2024 được nhận định là tăng [1]. Do vậy tôi có một nhận định rằng cả 2 giải pháp trên đều không đáp ứng được tính tự động của mô hình

định giá bất động sản về khía cạnh dữ liệu mới và khía cạnh thời gian.

Dựa vào những đánh giá phân tích và đánh giá nói trên, các giải pháp chỉ đang dừng ở mức đưa ra con số chứ chưa đảm bảo tính tăng trưởng của giá nhà theo thời gian và tính tự động cập nhật mô hình với dữ liệu bất động sản mới hàng ngày. Vì vậy tôi quyết định tập trung vào việc tối ưu thuật toán định giá nhà cùng với tính tự động trong hệ thống định giá bất động sản.

Để có thể đạt được những yếu tố này thì hệ thống cần có khả năng dự đoán được giá bất động sản theo khu vực, theo các thông tin nhập vào như Bigge nhưng đơn giản hơn. Tối ưu việc định giá bất động sản cho người dùng như OneHousing nhưng hệ thống cũng phải đảm bảo được tính tự động. Vì vậy hệ thống cần có những chức năng sau:

- Cập nhật thông tin mới tự động cho mô hình Mô hình phải được tự động cập nhật với dữ liệu bất động sản mới theo thời gian
- Mô hình dự đoán giá bất động sản theo thời gian, vị trí và các thông người dùng mong muốn
- Quá trình huấn luyện và tracking các phiên bản của mô hình là tự động và việc đánh giá các mô hình, deploy các mô hình cũng đáp ứng độ tự động

Hệ thống do tôi đề xuất kế thừa những điểm mạnh của các giải pháp trên và đảm bảo được tính mới của mô hình dự đoán bất động sản.

Đối với đồ án, tôi quyết định tập trung vào thị trường bất động sản Việt Nam, định giá bất động sản Việt Nam. Tuy nhiên đồ án vẫn xây dựng một hệ thống để có thể linh hoạt các miền dữ liệu bất động sản ở các quốc gia khác.

1.3 Định hướng giải pháp

Với những mục tiêu đã được xác định rõ ở phần trước, tôi xin đưa ra định hướng cho từng mục tiêu như sau:

- Cập nhật thông tin mới tự động cho mô hình Để đảm được rằng hệ thống luôn được tính toán và phân tích dựa trên dữ liệu mới thì một service nhỏ của hệ thống sẽ đi thu thập dữ liệu từ các trang mua bán bất động sản, làm sạch và tích hợp dữ liệu để tạo ra một nguồn dữ liệu thống nhất cho việc training phía sau
- Mô hình dự đoán giá bất động sản theo thời gian, vị trí và các thông người dùng mong muốn Để đảm bảo được mô hình dự đoán giá bất động sản theo thời gian thì trong quá trình xây dựng mô hình phải có yếu tố thời gian. Bên cạnh đó để cho mô hình có hiểu quả tốt hơn thì việc phân tích độ tương quan

và mối liên hệ giữa các bất động sản cùng một khu vực rất quan trọng trong việc xác định giá bất động ở khía cạnh khu vực. Đặc biệt mô hình phải đa dạng hóa và phân tích được mức độ ảnh hưởng của các thuộc tính lên giá bất động sản để từ đó xây dựng được một mô hình dự đoán giá tốt hơn

- Quá trình huấn luyện và tracking các phiên bản của mô hình là tự động và việc đánh giá các mô hình, deploy các mô hình cũng đáp ứng độ tự động

Để đảm bảo được tính tự động huấn luyện của mô hình thì việc xây dựng message queue quan trọng việc khi nào thì mô hình sẽ được huấn luyện lại với dữ liệu mới. Bên cạnh đó tôi sử dụng metric Root Mean Squared Error (RMSE) trong bài toán này để đánh giá mô hình sau khi tranining xong và nếu mô hình vượt qua được các tập an toàn (tập dữ liệu đảm bảo mô hình phải tốt trên tập này được gọi là trustworthy model) thì quyết định deploy mô hình ở giai đoạn sản phẩm.

Củ thể hơn đối với việc cập nhật dữ liệu mới cho mô hình thì hệ thống lập lịch để thu thập dữ liệu, làm sạch dữ liệu, tích hợp dữ liệu và đẩy vào cơ sở dữ liệu, sẵn sàng cho quá trình training mô hình. Để xây dựng được hệ thống định giá bất động sản dự đoán chính xác theo thời gian và vị trí thì hệ thống sẽ xây dựng machine learning, deep learning và thực hiện engineer feature các thông tin về vị trí, thời gian và thực hiện ensemble các mô hình để hệ thống có thể mô hình hóa được dữ liệu bất động sản tốt hơn. Sau khi mô hình được huấn luyện xong, hệ thống sẽ đánh giá mô hình mới bằng tập trustworthy dataset. Nếu đảm bảo mức độ an toàn trên tập này thì mô hình có thể được deploy còn nếu không thì mô hình sẽ không được deploy ở production phrase. Bên cạnh đó hệ thống cũng tracking được những trường hợp bất động sản có giá bất thường so với hệ thống định giá. Điều đó giúp hệ thống có thể phân tích, tìm hiểu và điều chỉnh mô hình.

1.4 Đóng góp của đồ án

Dựa vào những đề xuất đã trình bày ở mục trên, đồ án có những đóng góp, công hiến chính:

- Hệ thống hoàn chỉnh từ việc thu thập dữ liệu, xử lý dữ liệu và lưu trữ dữ liệu thành một thể thống nhất vào cơ sở dữ liệu
- Đánh giá giải thuật đề xuất và so sánh với các giải thuật đã có dựa trên miền dữ liệu bất động sản thu thập tại Việt Nam
- Training mô hình tự động, đánh giá mô hình và deploy mô hình một cách tin cậy
- Xây dựng một hệ thống sẵn sàng cho việc định giá bất động sản một cách tự động

- Bước đệm tốt cho các MLOps pipeline trong tương lai

1.5 Bố cục của đồ án

Dựa vào những mục tiêu và giải pháp tôi đề xuất ở các mục trên, phần còn lại của đồ án được tổ chức như sau

Chương 2 của nghiên cứu tập trung vào việc phân tích các giải pháp hiện có. Bên cạnh đó đánh giá tổng quan hiệu quả, ưu và nhược điểm của các giải pháp. Từ đó đề ra các yêu cầu của hệ thống về khía cạnh chứ năng và phi chức năng.

Chương 3 của nghiên cứu tập trung vào việc phân tích cơ sở lý thuyết đã được sử dụng trong hệ thống. Từ đó có được nền tảng tốt hơn cho các mục đích sau này của hệ thống.

Chương 4 của nghiên cứu tập trung vào việc phân tích các thành phần của hệ thống giải quyết vấn đề chính của nghiên cứu như thế nào. Phase đầu tiên của hệ thống đó chính là thu thập dữ liệu, tiền xử lý và tích hợp dữ liệu. Đồng thời cải thiện các điểm chưa tốt của hệ thống để đảm bảo hệ thống đủ tin cậy trong quá trình thu thập dữ liệu. Tiếp xây dựng mô hình AI, đánh giá và cải tiến mô hình AI đã có trong bài toán dự đoán giá bất động sản. Engineer features và lựa chọn features cho việc training. Optimize các điểm yếu của các mô hình đơn bằng các mô hình tổng hợp. Bên cạnh đó phân tích insight của các features, tiền xử lý dữ liệu và đánh giá performance của mô hình. Bên cạnh đó đề cập đến việc quản lý các mô hình AI, traning tự động, quản lý các feature của mô hình và các thông số của từng feature, việc monitor hệ thống, quản lý lỗi của các service trong hệ thống, trực quan hóa để dễ dàng trong việc phân tích và xử lý các task vụ lỗi khi cần thiết.

Chương 5 của nghiên cứu tập trung vào việc đánh giá hiệu năng của hệ thống và hiệu năng của giải thuật: độ ổn định và throughput của hệ thống. Mô tả những thực nghiệm đã thử nghiệm trên hệ thống. Trả lời các câu hỏi từ vấn đề đã được đề cập ở các chương trên

Chương 6 của nghiên cứu tập trung vào trình bày tính riêng biệt của hệ thống, tạo nên những điểm khác biệt của đồ án trong lĩnh vực bất động sản. Ở chương này tôi xin đề cập tới những điểm nhận diện đặc trưng của đồ án và những điểm mới lạ.

Chương 7 của nghiên cứu tập trung vào các kết quả nhận được của dự án và đề xuất hướng phát triển tiếp theo của sản phẩm trong tương lai.

Chương 8 của nghiên cứu đưa ra lời kết luận đối với các vấn đề chính xuyên suốt nghiên cứu

Trong những năm gần đây, các mô hình dự đoán đã đạt được những tiến bộ nhất định. Sự ra đời của các mô hình trí tuệ nhân tạo (AI), phân tích thông kê cùng với

dữ liệu lớn đã giúp cho việc dự đoán có phần tốt hơn. Trong lĩnh vực bất động sản cũng không phải ngoại lệ. Cụ thể, các mô hình Machine Learning, Deep Learning đã có khả năng bao quát hóa việc dự đoán hơn so với việc thậm định giá bằng con người - mất tương đối khá nhiều thời gian và mang tính chủ quan. Vì thế đối với các doanh nghiệp, họ không những cung cấp cho người dùng tìm kiếm bất động sản mà còn các dịch vụ AI nói chung và dự đoán giá nhà nói riêng.

Trong lĩnh vực bất động sản, bài toán dự đoán giá nhà tập trung vào việc xây dựng một hệ thống dự đoán giá của bất động sản dựa vào các thông tin được nhận từ yêu cầu của người dùng: số phòng ngủ, số phòng tắm, vị trí. Điều này đóng vai trò rất quan trọng trong việc định giá bất động sản giúp người dùng và cung cấp có một góc nhìn tổng qua hơn về giá của vị trí ngôi nhà người dùng mong muốn trong thời điểm thổi giá bất động sản tại thị trường Việt Nam hiện nay.

Hiện tại các phương pháp dự đoán giá nhà tuy vẫn sử dụng thông tin cơ bản của ngôi nhà để xây dựng mô hình nhưng còn rất cơ bản và còn thủ công, không mang tính tự động. Vì thế để có một mô hình dự đoán giá chính xác cùng với một hệ thống tự động training trở nên vô cùng quan trọng hơn đối với doanh nghiệp.

1.6 Các giải pháp hiện tại và hạn chế

1.6.1 Xây dựng mô hình không tin cậy

Hầu hết các giải pháp dự đoán giá bất động sản đều dựa vào các thông tin cơ bản của ngôi nhà. Việc đưa ra kết quả dự đoán giá cũng giống như việc hệ thống dự đoán giá tiếp thu được lượng lớn ngôi nhà tổng quát hóa và lựa chọn giá hợp lý cho ngôi nhà mà người dùng quan tâm. Để mô hình dự đoán giá nhà hiệu quả cần có sự kết hợp của các điều kiện:

- Mô hình phải mô hình hóa tốt dữ liệu. Để làm được điều này, mô hình cần phải được huấn luyện với lượng dữ liệu vừa đủ và đa dạng.
- Mô hình phải có mức độ tổng quát hóa cao. Với các yêu cầu định giá nhà của người dùng thì mô hình phải học được những thông tin đã có và đưa ra kết quả phù hợp nhất dựa vào những tri thức đã được học.

Tuy nhiên, phương pháp nêu trên vẫn còn nhiều điểm còn rất hạn chế:

- Do các mô hình hiện tại chỉ sử dụng các thông tin cơ bản của ngôi nhà mà bỏ quên một vấn đề rằng: Các ngôi nhà trong 1 vùng có mối liên hệ đặc biệt với nhau. Một mô hình chỉ học tốt trên một vài phân bố của dữ liệu vì thế mà hạn chế lớn nhất của các mô hình là chưa tận dụng được mixture của các mô hình con để tạo được một mô hình tốt hơn và tổng quát hơn
- Do mô hình dự đoán giá được huấn luyện với dữ liệu tại thời điểm đó nên khi

được yêu cầu dự đoán với dữ liệu có phân bố khác với tập huấn luyện thì mô hình khó có thể đưa ra được một kết quả tốt và phù hợp. Bên cạnh đó, các mô hình được huấn luyện mới với dữ liệu mới có thể không tốt với dữ liệu đã được training trước đó. Do đó quá trình production gặp những vấn đề không mong muốn, đặc biệt ảnh hưởng trực tiếp tới người dùng nếu không có quy trình quản lý và giám sát mô hình chuẩn chỉ.

1.6.2 Quy trình xây dựng mô hình thủ công

Hầu hết việc build các service AI còn mang tính thủ công. Điều đó có nghĩa là sau khi huấn luyện mô hình với dữ liệu thì sử dụng mô hình đã được huấn luyện và dùng cho quá trình inference ở production phrase. Mỗi khi có một lượng data mới, việc quyết định training với dữ liệu mới và chọn version nào để phục vụ ở production phrase là mang tính chủ quan của đội developer và không có số liệu cụ thể, khó để tracking. Việc thủ công chuẩn bị dữ liệu cũng ảnh hưởng đến hệ thống về mặt tự động: (i) xử lý dữ liệu thủ công sau khi thu thập quy trình build mô hình trở nên kém tính tự động hóa, (ii) việc build mô hình tự động khiến việc tracking và rollback các version tranining mô hình, lựa chọn features nào được chạy inference ở giai đoạn sản phẩm

1.7 Mục tiêu và định hướng giải pháp

1.7.1 Mục tiêu

Để giải quyết vấn đề mô hình AI bị lỗi thời với các dữ liệu mới và chưa được quản lý một cách tự động thì đồ án tập trung phát triển một hệ thống MLOPS trong bài toán dự đoán giá bất động sản tại thị trường Việt Nam. Mục tiêu chính là tạo ra một mô hình có thể dự đoán chính xác, tự động và dễ dàng hơn các giải pháp đã có: (i) Hệ thống hoàn chỉnh từ quá trình thu thập dữ liệu realtime, xử lý dữ liệu, xây dựng các thuật toán AI, training mô hình tự động đến quá trình public AI API, (ii) đánh giá và giám sát hệ thống khách quan và dễ dàng. (iii) người dùng có thể tương tác với chatbot và thực hiện từng bước trong luồng MLOps: thu thập dữ liệu, làm sạch dữ liệu, tạo dữ liệu huấn luyện, huấn luyện mô hình và định giá bất động sản.

1.7.2 Định hướng giải pháp

Để đáp ứng các mục tiêu đề ra, đề xuất giải pháp sau:

(i) Hệ thống thu thập và xử lý dữ liệu thời gian thực Triển khai các công cụ thu thập dữ liệu từ nhiều nguồn đa dạng như website bất động sản. Sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) và khai phá dữ liệu (data mining) để trích xuất và chuẩn hóa thông tin relevant. Áp dụng các thuật toán học máy để tự động phân loại, lọc và tổng hợp dữ liệu, đảm bảo chất lượng và tính cập nhật.

(ii) Xây dựng và huấn luyện mô hình AI tự động: Phát triển các mô hình học máy tiên tiến và học sâu (deep learning) để dự đoán giá bất động sản. Triển khai quy trình huấn luyện mô hình tự động hóa, bao gồm lựa chọn hyperparameter, tối ưu hóa mô hình và đánh giá hiệu suất. Cập nhật mô hình liên tục với dữ liệu mới để đảm bảo độ chính xác và khả năng thích ứng với thị trường thay đổi.

(iii) Cung cấp API AI công khai: Thiết kế và phát triển API AI RESTful dễ sử dụng, cho phép truy cập các tính năng dự đoán giá bất động sản. Cung cấp tài liệu hướng dẫn và ví dụ mã để hỗ trợ tích hợp API vào các ứng dụng và dịch vụ khác nhau. Đảm bảo bảo mật và quyền truy cập dữ liệu cho API, đáp ứng các tiêu chuẩn và quy định hiện hành.

(iv) Hệ thống đánh giá và giám sát: Thiết lập hệ thống giám sát tự động để theo dõi hiệu suất mô hình, phát hiện lỗi và cảnh báo bất thường. Sử dụng các bảng điều khiển trực quan và báo cáo chi tiết để hiển thị hiệu suất mô hình, độ tin cậy của dữ liệu và các chỉ số quan trọng khác. Cung cấp khả năng truy cập và phân tích dữ liệu giám sát để hiểu rõ hơn về hành vi của mô hình và cải thiện hiệu suất.

(v) Tương tác với chatbot và thực hiện luồng MLOps: Phát triển chatbot hỗ trợ người dùng tương tác và thực hiện từng bước trong luồng MLOps. Cho phép người dùng thu thập dữ liệu, làm sạch dữ liệu, tạo dữ liệu huấn luyện, huấn luyện mô hình và dự đoán giá bất động sản thông qua chatbot. Cung cấp hướng dẫn và hỗ trợ trong suốt quá trình, đảm bảo trải nghiệm người dùng đơn giản và hiệu quả. Bằng cách triển khai giải pháp toàn diện này, hệ thống sẽ cung cấp một nền tảng mạnh mẽ cho việc dự đoán giá bất động sản, hỗ trợ ra quyết định đầu tư sáng suốt và thúc đẩy sự phát triển của thị trường bất động sản.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH TỔNG QUAN

Đến với chương thứ 2 thì tôi sẽ khảo sát qua các giải pháp hiện có và phân tích yêu cầu và các chức năng một cách tổng quan của hệ thống

2.1 Khảo sát

Biggee bên cạnh là một nền tảng bất động sản: cung cấp và mua bán bất động sản thì còn cung cấp công cụ để định giá bất động sản [2]. Công cụ này hỗ trợ người dùng định giá nhà đất dựa trên các thông tin nhập vào của người dùng.

The screenshot shows the Biggee website's land price estimation feature. At the top, there is a logo for 'biggee' with the tagline 'bigger than the biggest' and a search bar labeled 'ĐỊNH GIÁ NHÀ ĐẤT'. Below the search bar are four buttons: a magnifying glass for search, a plus sign for add, a map icon for location, and a user icon. To the right of the search bar is a button to 'Tim vị trí hoặc tọa độ' (Search location or coordinates). The main form consists of several input fields and dropdown menus:

- Địa chỉ gợi ý nhớ:** Ốc Vườn Chuối, Vườn Chuối, Cư xá Đô Thành, phường 4, District 3,
- Số tờ:** Nhập số tờ
- Số thửa:** Nhập số thửa
- Khu vực:** Phường 4, Quận 3, Hồ Chí Minh
- Diện tích (m²):** (empty field)
- Đặc điểm tốt:**
 - Tiếp giáp trên 2 mặt tiền đường hoặc hẻm
 - Gần chợ, siêu thị phạm vi 100m
 - Nở hậu
 - Hướng Nam hoặc Đông Nam
- Đặc điểm bất lợi:**
 - BDS tọa lạc tại hẻm cụt
 - BDS có đường hướng vào nhà
 - Gần mồ mả, nghĩa trang, nhà tang lễ phạm vi 100m
 - BDS có hình dáng xấu
- Mặt tiền (m):** 5
- Chiều dài (m):** 12
- Diện tích đất xây dựng (m²):** 60
Nhập 0 nếu là đất trống.
- Số tầng:** 3
Nhập 0 nếu là đất trống.

Hình 2.1: Màn hình định giá nhà đất của biggee

Hình vẽ 2.1 mô tả công cụ định giá nhà đất của biggee. Như chúng ta được thấy

để định giá được nhà đất thì biggee yêu cầu các thông số như nhà đất ở khu vực nào, diện tích ngôi nhà, các đặc điểm của ngôi nhà, mặt tiền rộng mấy mét, ... Bên cạnh đó biggee còn yêu cầu các thông tin như số tờ và số thửa. Một trong những thông tin mà biggee yêu cầu nữa đó chính là tờ thửa và số thửa.

Như ta thấy thì việc một người dùng vào để định giá bất động sản thì việc các thông tin số tờ thửa rất khó để người dùng hiểu và việc đó có thể khiến cho kết quả dự đoán sai lệch ảnh hưởng tới thông tin mà người dùng tiếp nhận. Điều này làm trải nghiệm của người dùng trở nên xấu đi.

Hơn thế nữa một nhược điểm lớn là mô hình định giá nhà đất của biggee không thay đổi trong khoảng thời gian tương đối dài . Tôi có thử cùng 1 câu tìm kiếm định giá nhà đất tại đường "Nguyễn Bỉnh Khiêm quận 1, thành phố Hồ Chí Minh" thời điểm cách nhau khoảng 6 tháng trước thì kết quả không thay đổi. Trong khi đó giá trung bình của con đường này tăng 15-20 phần trăm so với cùng kỳ.

Đối với giải pháp thứ hai đến từ sản phẩm OneHousing. Cho phép người dùng tra được giá của ngôi nhà chỉ dựa vào mỗi địa chỉ [3].



Hình 2.2: Màn hình định giá ngôi nhà ở OneHousing

Hình vẽ 2.2 thể hiện thông tin màn hình định giá ngôi nhà ở OneHousing.

Ưu điểm của giải pháp này là nhanh gọn và tiết kiệm thời gian đối với những người dùng nào muốn định giá nhanh qua các ngôi nhà thuộc khu vực họ chọn. Tuy nhiên nhược điểm đó chính là khách hàng muốn định giá ngôi nhà có bao nhiêu phòng ngủ, diện tích tại khu vực đó giá thế nào thì OneHousing không thể đáp ứng được. Do tính chưa xác thực của ngôi nhà nên là việc định giá ngôi nhà của OneHousing chưa đủ tin cậy.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH TỔNG QUAN

The screenshot shows a property listing for a house at Số 75 - Ngõ 75 - Giải Phóng, P. Đồng Tâm - Q. Hai Bà Trưng - TP. Hà Nội. The listing includes a price range of 1 tỷ 692 triệu - 1 tỷ 862 triệu, a note about Techcombank financing, and a 'Xác thực để xem giá chính xác' button. It also shows a 'Chưa xác minh căn nhà' button. A sidebar on the right asks if the user is the owner ('Bạn có phải là chủ sở hữu căn hộ này?') and provides a 'Tôi là chủ nhà' button. Below this, it says 'Bạn đang có nhu cầu gì về bất động sản này?' and 'Hiểu đúng nhu cầu sẽ giúp OneHousing hiển thị những thông tin phù hợp nhất với bạn'. Navigation buttons for Mua, Bán, Thuê, and Cho thuê are visible.

Hình 2.3: Kết quả định giá ở OneHousing

Màn hình 2.3 mô tả thông tin kết quả định giá của giải pháp OneHousing.

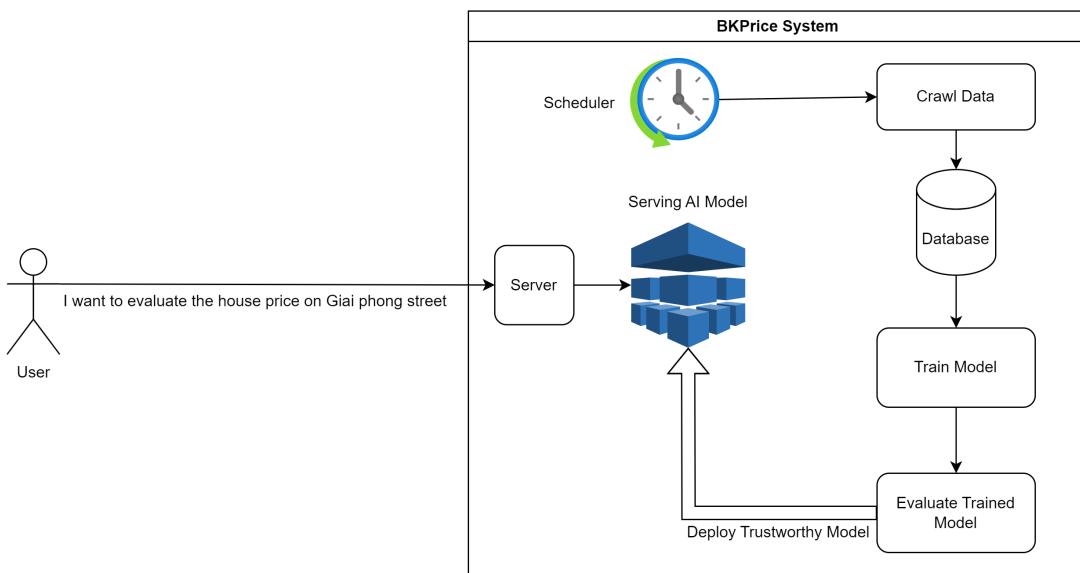
Sau khi khảo sát qua 2 giải pháp trên thì ta thấy rằng mỗi giải pháp lại có mỗi ưu và nhược điểm riêng. Điểm mạnh của giải pháp này là điểm yếu của giải pháp kia và ngược lại. Vì vậy, cần có một phương pháp tối ưu hơn để giải quyết các vấn đề còn tồn đọng trên.

Với sự tìm tòi và học hỏi, trong nghiên cứu này, tôi xin đề xuất một quy trình MLOps tự động nâng cao chất lượng định giá bất động sản, có tên là BKPrice System, một giải pháp kết hợp những điểm mạnh của 2 giải pháp trên. Quy trình MLOps đề xuất cải thiện được tính tự động của hệ thống định giá bất động sản và giải thuật định giá đa mức thông tin cải thiện độ tin cậy của kết quả định giá.

2.2 Phân tích chức năng tổng quan hệ thống

2.2.1 Tổng quan hệ thống

Phần này sẽ phân tích tổng quan hệ thống và những chức năng chính của hệ thống.



Hình 2.4: Sơ đồ hoạt động tổng quan của hệ thống

Hình vẽ 2.4 minh họa sơ đồ tổng quan của hệ thống BKPrice. Hệ thống nhận yêu cầu định giá bất động sản với các thông tin: số phòng ngủ, vị trí, ... từ người dùng. Phía server sẽ nhận yêu cầu từ người dùng và chuyển tới AI service để thực hiện quá trình tính toán và trả về kết quả dự đoán cho người dùng. Điểm đặc biệt của BKPrice đó chính là khả năng tự động huấn luyện mô hình với dữ liệu mới được thu thập theo hàng ngày từ bộ lập lịch (Scheduler). Để đảm bảo tính tin cậy của các mô hình AI được huấn luyện tự động thì trước khi triển khai, mô hình trí tuệ nhân tạo sẽ được đánh giá chất lượng bởi tập dữ liệu tin cậy để đảm bảo mô hình AI không bị quá tệ sau khi traning. Việc quyết định triển khai mô hình sẽ trở nên dễ dàng hơn khi có sự so sánh giữa các phiên bản mô hình với nhau. Bên cạnh đó việc quản lý đặc trưng của dữ liệu và sử dụng đa dạng tập thuộc tính cũng sẽ được giám sát một cách rõ ràng.

Vì vậy nhờ khả năng tự động và tin cậy, hệ thống BKPrice đã thể hiện được sự vượt trội so với giải pháp biggee, onehousing về khía cạnh linh hoạt hơn về các yếu tố định giá thông tin ngôi nhà và khả năng cập nhật tri thức của mô hình định giá bất động sản.

2.2.2 Các thành phần chính của hệ thống

- Scheduler Component: Có chức năng lên lịch thu thập dữ liệu từ các nguồn trang bất động sản
- Crawl Data Component: Có vai trò thu thập dữ liệu theo lịch trình từ Scheduler module, xử lý dữ liệu và tích hợp dữ liệu và lưu trữ dữ liệu sạch vào database

- Train Model Component: Đảm nhận nhiệm vụ quản lý tập thuộc tính, huấn luyện mô hình và quản lý mô hình
- Evaluate Trained Model Component: Có vai trò đánh giá mô hình vừa được huấn luyện xong, benchmark mô hình tự động. Từ đó thực hiện quá trình cập nhập mô hình định gia bất động sản.

Tóm lại: Qua chương số 2 thì chúng ta đã hiểu rõ tổng quan cách hệ thống hoạt động và đáp ứng yêu cầu của người dùng. Bên cạnh đó có cách nhìn khách quan hơn về các thành phần cơ bản của hệ thống.

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

Ở chương thứ 2, nghiên cứu cung cấp một góc nhìn tổng quan về các giải pháp cho bài toán định giá bất động sản tại thị trường Việt Nam. Bên cạnh nghiên cứu cũng phân tích các ưu nhược điểm của các giải pháp đó. Từ đó đề xuất ra giải pháp mang tên hệ thống định giá bất động sản tự động BKPrice. Đồng thời chương trước ta đã thấy được các thành phần của hệ thống BKPrice. Do đó để làm rõ hơn các thành phần đó thì chương này sẽ trình bày các công nghệ mà hệ thống sử dụng và vì sao tôi lại chọn công nghệ đó. Ở chương này sẽ được chia làm các phần nhỏ đề cập tới lý thuyết, thuật toán, framework và các thư viện hỗ trợ quá trình tạo nên hệ thống BKPrice.

3.1 Giải thuật và tính hiệu quả của hệ thống

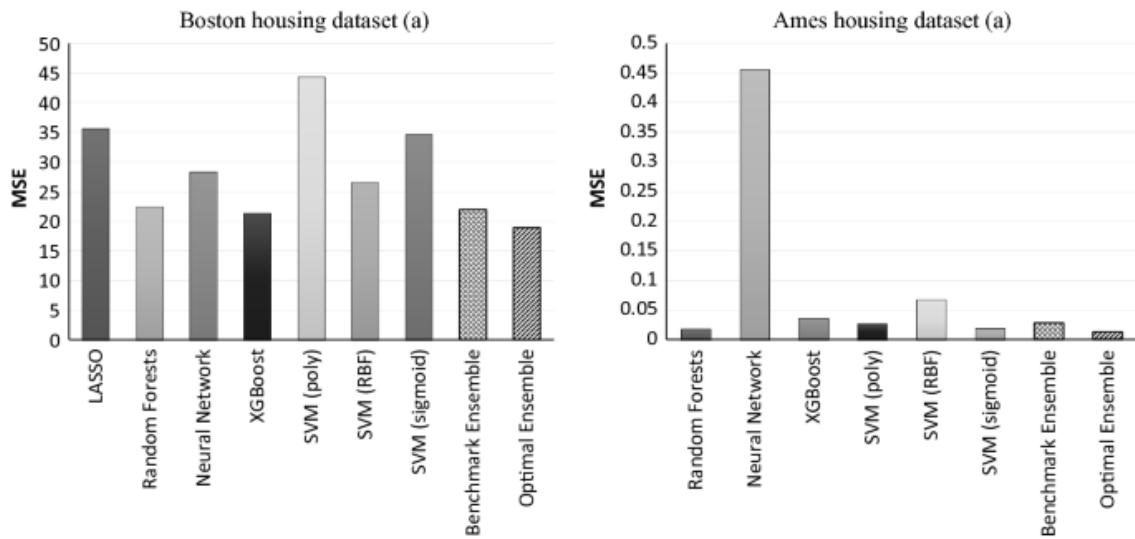
Ở trong phần này, tôi sẽ đưa ra những giải thuật để giải quyết một trong những vấn đề lớn của nghiên cứu này. Đó chính là tính khả thi của hệ thống nói chung và tính sẵn sàng của hệ thống dự đoán giá bất động sản nói riêng. Với sự tự động về mặt dữ liệu và triển khai mô hình, việc xây dựng giải thuật dự đoán giá bất động sản phải đảm bảo được các yếu tố sau:

- Tính tổng quát hóa của giải thuật.
- Tính hiệu quả trong quá trình trích xuất đặc trưng và huấn luyện mô hình.
- Tính tin cậy của kết quả dự đoán.

Các mục tiếp theo sẽ đi giải quyết các vấn đề trên.

a, Mixtral of experts - Ensemble Learning

Với việc một mô hình AI học trên tập dữ liệu thì hành vi của mô hình sẽ tốt trên một vài phân bố dữ liệu nhưng lại tệ trên một vài phân bố dữ liệu khác. Vì vậy nhiều nghiên cứu chỉ ra rằng việc kết hợp nhiều mô hình lại với nhau tạo ra được một mô hình vượt trội so với đơn lẻ các mô hình. Điều này rất quan trọng trong việc đảm bảo độ an toàn dự đoán của hệ thống.



Hình 3.1: AI Model Benchmark

Hình vẽ trên mô tả độ đo MSE (Mean squared error) giữa các mô hình trong bài toán dự đoán. Như ta thấy thì lời giải ứng dụng Ensemble đang cho kết quả tốt hơn so với các mô hình khác.

Có 3 cách ứng dụng ensemble trong việc tạo ra một mô hình tổng [4]:

- Bagging: Xây dựng một lượng lớn các model (thường là cùng loại) trên những subsamples khác nhau từ tập training dataset (random sample trong 1 dataset để tạo 1 dataset mới). Những model này sẽ được train độc lập và song song với nhau nhưng đầu ra của chúng sẽ được trung bình cộng để cho ra kết quả cuối cùng.
- Boosting: Xây dựng một lượng lớn các model (thường là cùng loại). Mỗi model sau sẽ học cách sửa những errors của model trước (dữ liệu mà model trước dự đoán sai) từ đó tạo thành một chuỗi các model mà model sau sẽ tốt hơn model trước bởi trọng số được update qua mỗi model. Trọng số của những dữ liệu dự đoán đúng sẽ không đổi, còn trọng số của những dữ liệu dự đoán sai sẽ được tăng thêm. Kết quả của model cuối cùng trong chuỗi model này làm kết quả trả về
- Stacking: Xây dựng một số model (thường là khác loại) và một meta model (supervisor model), train những model này độc lập, sau đó meta model sẽ học cách kết hợp kết quả dự báo của một số mô hình một cách tốt nhất.

Một trong những lý do module dự đoán giá nhà của BKPrice sử dụng Ensemble learning là vì:

- Phân phối giá ở các vùng khác nhau, kiểu nhà cũng sẽ khác nhau. Vì vậy việc xây dựng nhiều mô hình và học trên từng tập dữ liệu nhỏ và tổng thể sẽ bao

quát hết được đa phần phân phối của dữ liệu. Từ đó việc ensemble các mô hình này lại sẽ tạo được một mô hình có đầy đủ kiến thức từ các mô hình con. (Mixtral of experts cũng là một thuật ngữ mới nổi lên cũng đề cập tới điều này)

- Việc ensemble các mô hình lại sẽ có khả năng giảm variance và bias của từng mô hình. Do vậy việc ensemble từ nhiều mô hình sẽ tạo ra một mô hình cân bằng và mang tính tổng quát hóa cao.

b, Principal component analysis - PCA

Việc sử dụng nhiều chiều dữ liệu trong các giải thuật đang là một trong những thách thức lớn khi xây dựng các mô hình trí tuệ nhân tạo. Vì vậy làm sao để mô hình có thể đảm bảo được mô hình học đầy đủ thông tin của dữ liệu với số lượng chiều ít hơn. Điều này đảm bảo tính hiểu quả về mặt thời gian khi xây dựng mô . Để giải quyết bài toán này trong học máy có một khái niệm tên là phân tích thành phần chính (Principal component analysis - PCA) [5]. PCA là một kỹ thuật giảm chiều dữ liệu phổ biến trong học máy. Nó được sử dụng để biến đổi một tập dữ liệu có nhiều biến thành một tập dữ liệu mới có ít biến hơn, trong khi vẫn giữ lại phần lớn thông tin ban đầu.

Nguyên tắc giảm chiều của PCA được diễn ra như sau:

- Tìm ma trận hiệp phương sai: Ma trận hiệp phương sai thể hiện mối tương quan giữa các biến trong tập dữ liệu.
- Tính toán giá trị riêng và vectơ riêng: Giá trị riêng đại diện cho mức độ biến thiên của dữ liệu dọc theo các hướng tương ứng, vectơ riêng là các hướng của sự biến thiên đó.
- Chọn các thành phần chính: Lựa chọn một số vectơ riêng có giá trị riêng lớn nhất để tạo thành tập dữ liệu mới có chiều thấp hơn.

Do đó PCA mang lại những ưu điểm sau: (i) Giảm nhiễu trong dữ liệu do lựa chọn các thành phần chính, (ii) Cải thiện hiệu suất của các thuật toán học máy, (iii) dễ dàng thực hiện và giải thích.

Tuy nhiên nếu việc lựa chọn số lượng chiều dữ liệu không chuẩn thì có thể dẫn đến ván đề mất đi những thông tin quan trọng trong bộ dữ liệu gốc.

Để giải quyết bài toán trên có một khái niệm là *Explained Variance (EV)* [6]. Củ thể đây là tỷ lệ phương sai được giải thích, là một thước đo trong Phân tích thành phần chính (PCA) thể hiện tỷ lệ phần trăm biến đổi dữ liệu được giữ lại bởi một số lượng thành phần chính nhất định. EV cao cho thấy các thành phần chính đó capture được nhiều thông tin quan trọng trong dữ liệu gốc. EV thấp cho thấy

các thành phần chính đó thể hiện ít thông tin quan trọng, và có thể loại bỏ để giảm thiểu kích thước dữ liệu mà không ảnh hưởng đáng kể đến hiệu suất của mô hình học máy.

Ta có tập dữ liệu S . Giả sử có thể mô hình hóa thông tin của tập dữ liệu S bởi k chiều. Gọi $\lambda_1, \lambda_2, \dots, \lambda_k$ là k trị riêng của k chiều. Khi đó giá trị *Explained Variance* (EV) của thành phần thứ t có công thức như sau:

$$EV_t = \frac{\lambda_t}{\sum_{i=1}^k \lambda_i}$$

Không mất tính tổng quát giả sử: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{k-1} \geq \lambda_k$.

Gọi O phần trăm là ngưỡng thông tin tối thiểu mà phương PCA bảo toàn được. Do đó số lượng thành phần tối ưu d^* của PCA định nghĩa là số nhỏ nhất thỏa mãn:

$$\sum_{i=1}^{d^*} EV_i \geq O$$

Ý nghĩa của công thức trên là: số lượng thành phần chính trong PCA sẽ là số bé nhất thỏa mãn rằng: Tổng tích lũy phương sai của những thành phần đó ít nhất phải giải thích được O phần trăm dữ liệu ban đầu.

c, Mô hình xác suất

Việc sử mô hình học máy, học sâu trong dự đoán giá bất động sản luôn đi kèm với giả thuyết. Ví dụ:

- Sử dụng mô hình hồi quy tuyến tính (Linear Regression Model) thì giả thuyết đặt ra là dữ liệu đang xấp xỉ một hàm tuyến tính nào đó.
- Sử dụng các mô hình cây (Tree-based Model) thì giả thuyết đặt ra là dữ liệu đang xấp xỉ các biểu diễn điều kiện nếu không thì trong các mô hình cây

Các thông số của bất động sản cũng có những giả thuyết về phân bố dữ liệu. Dó đó các mô hình xác suất sẽ giải quyết vấn đề này. Việc nắm bắt được phân bố dữ liệu không những tạo ra được nhiều thuộc tính giúp mô hình hóa tốt dữ liệu hơn mà còn hậu xử lý kết quả mà mô hình đưa ra. Ở trong nghiên cứu này mô hình xác suất có hai vai trò sau: (i) phân loại mẫu dữ liệu theo từng thành phần trong phân phối hỗn hợp, (ii) hiểu chỉnh kết quả dự đoán nằm trong phân bố xác suất giá (quận, kiểu nhà, ...)

Trong mô hình xác suất có các loại biến khác nhau và mỗi loại biến lại có những đặc điểm riêng:

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

- Biến quan sát được (Observed variable) là biến để mô tả những gì quan sát được (số lượng phòng ngủ, diện tích bất động sản, ...)
- Biến ẩn (Hidden variable) là biến để mô tả những thuộc tính ẩn của dữ liệu mà chúng ta không thể thu thập được (Mức độ thuận tiện của bất động sản, ...)

Trong mô hình xác suất, chúng ta sẽ đặt ra các giả thuyết các biến sẽ tuân theo các giả thuyết nhất định. Một mô hình xác suất có thể có nhiều biến cũng đồng nghĩa với việc có thể có nhiều giả thuyết.

Tuy nhiên mỗi biến có thể tuân theo một phân bố hoặc nhiều phân bố khác nhau. Khi đó có một mô hình xác suất *GMM* (Gaussian Mixture Model) [7] để giải quyết vấn đề một biến tuân theo nhiều phân bố xác suất. GMM là một mô hình thống kê được sử dụng để mô hình hóa phân bố của dữ liệu. Nó giả định rằng dữ liệu được tạo ra bởi hỗn hợp của một số phân phối chuẩn (Gaussian) với các tham số khác nhau.

Giả sử dữ liệu của chúng ta được sinh từ k phân bố chuẩn (Gaussian) và mỗi điểm dữ liệu chỉ được sinh từ một trong k phân bố chuẩn. Và xét mỗi điểm dữ liệu x chỉ có một trường thuộc tính (Số phòng ngủ, diện tích hoặc giá, ...)

Ta có: k bộ $\{(\mu_1, \sigma_1), (\mu_2, \sigma_2), (\mu_3, \sigma_3), \dots, (\mu_k, \sigma_k)\}$ của k phân bố chuẩn

Gọi Z là biến tuân theo phân bố đa thức có giá trị là chỉ số của phân bố khi chọn ngẫu nhiên một phân bố trong k phân bố. Các giá trị của Z có thể là: $[1, 2, 3, \dots, k]$

Ta có hàm xác suất của z là:

$$p(z = i) = p_i \quad (3.1)$$

p_i là xác suất khi chọn ngẫu nhiên một điểm dữ liệu thì điểm dữ liệu đó được sinh ra bởi phân bố thứ i .

$p = [p_1, p_2, \dots, p_k]$, $p_i \geq 0$ là tham số của phân bố đa thức thỏa mãn $p_1 + p_2 + \dots + p_n = 1$.

Hàm mật độ GMM được định nghĩa như sau:

$$p_1 f(x|\mu_1, \sigma_1^2) + p_2 f(x|\mu_2, \sigma_2^2) + \dots + p_n f(x|\mu_n, \sigma_n^2) \quad (3.2)$$

Trong đó: $f(x|\mu_i, \sigma_i^2)$ là hàm mật độ của phân phối chuẩn có công thức như sau:

$$f(x|\mu_i, \sigma_i^2) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \quad (3.3)$$

Với hàm mất độ GMM được định nghĩa ở công thức (1), để có thể phân loại được điểm dữ liệu x thuộc phân bố nào ta thực hiện như sau:

- Tính toán giá trị hàm mật độ xác suất $f(x|\mu_i, \sigma_i^2)$, $1 \leq i \leq k$
- Hàm phân bố nào có giá trị lớn nhất thì điểm x thuộc về hàm phân bố đó

Do đó việc biết được trường thuộc tính F của mẫu dữ liệu bất động sản thuộc vào phân bố nào, thì việc xây dựng mô hình học máy, học sâu và dự đoán sẽ trở nên chính xác hơn.

d, Quadtree - Tối ưu tốc độ tìm kiếm trong không gian

Ở phần tiếp theo tôi sẽ đề cập tới vấn đề trích xuất đặc trưng để xây dựng tập huấn luyện mô hình AI. Trong đó có trích xuất đặc trưng từ các bất động sản lân cận. Tuy nhiên chúng ta có vấn đề sau:

Bài toán: Gọi M là số lượng bất động sản ($M \geq 300000$), N là số nguyên dương ($N \geq 1000$). Thực hiện N lần tìm kiếm bất động sản lân cận.

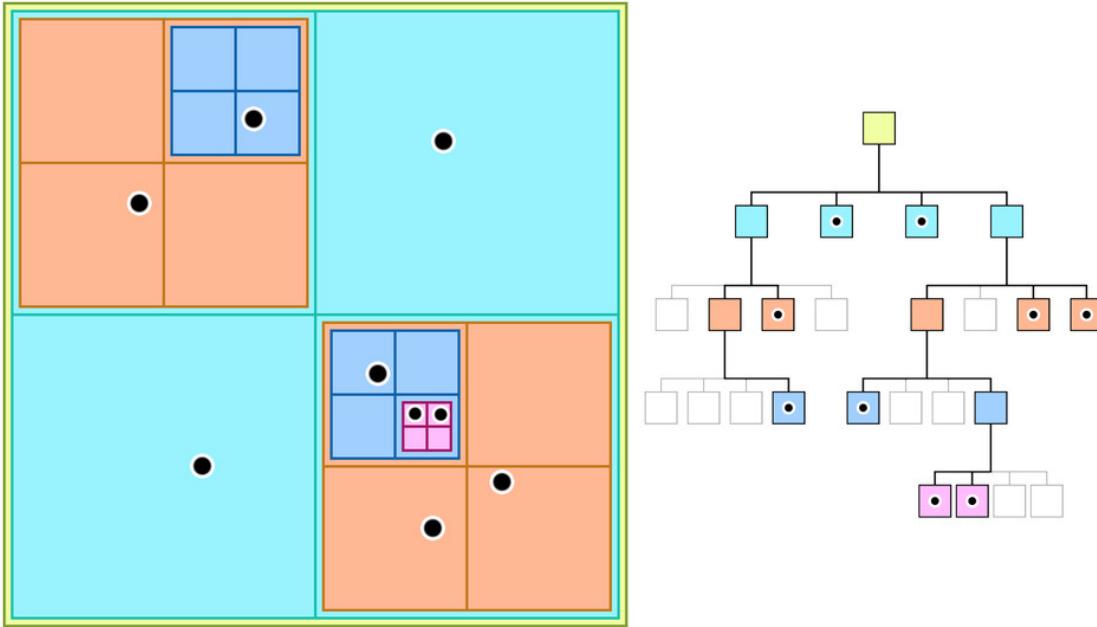
Ta có thể tìm kiếm trong cơ sở dữ liệu như sau:

- $Q_i : (lat_i, lon_i)$ là câu query thứ i , $1 \leq i \leq N$
- $A_j : (lat_j, lon_j)$ là bất động sản thứ $1 \leq j \leq M$
- D_{ij} là khoảng cách giữa 2 điểm không gian (lat_i, lon_i) và (lat_j, lon_j) với $1 \leq i \leq N, 1 \leq j \leq M$
- Với mỗi câu query $1 \leq i \leq N$, gọi A_k là bất động sản có khoảng cách $d = \min_u D_{iu}, 1 \leq u \leq M$

Đối với mỗi query thì hệ thống phải tìm kiếm toàn bộ M bất động sản (độ phức tạp là $O(M)$), do đó ta có thể thấy ngay độ phức tạp của giải thuật là $O(N \times M)$, $N \geq 1000, M \geq 100000$. Như vậy giải thuật này không hiệu quả nhất là trong pha sản phẩm vì độ phức tạp lớn khi số lượng bất động sản lớn.

Tôi đề xuất sử dụng một cấu trúc dữ liệu có tên là *Quadtree* [8].

Quadtree là một cấu trúc dữ liệu phân nhánh dạng cây, được sử dụng để phân hoạch vùng không gian hai chiều hiện tại thành các vùng nhỏ và dễ quản lý hơn. Hình vẽ 3.2 mô tả cấu trúc cây quadtree. Khác với cây nhị phân, mỗi lần phân hoạch, Quadtree chia vùng hiện tại thành 4 vùng. Do đó mỗi node trong Quadtree có 4 node con hoặc không có node con nào. Việc sử dụng Quadtree giúp quản lý các bất động sản trong không gian một cách hiệu quả, bằng cách phân chia vùng cần xử lý các đối tượng thành những vùng con, các đối tượng sẽ được đưa vào các vùng tương ứng và được quản lý riêng biệt. Quá trình phân chia cứ thế tiếp diễn



Hình 3.2: Mô tả cấu trúc dữ liệu Quadtree

cho đến khi mỗi vùng chỉ chứa một số lượng bất động sản nhất định hoặc mức độ phân chia đạt mức chấp nhận được (Khoảng cách giữa 2 bất động sản trong vùng luôn bé hơn d).

Trong nghiên cứu này tôi mô phỏng M bất động sản thành không gian 2 chiều như trên. Mỗi ô nhỏ sẽ chứa một vài bất động sản.

Do đó với mỗi câu query (lat_i, lon_i) , ta thực hiện quá trình tìm kiếm như sau:

- Tìm kiếm vị trí của ô hợp lệ cho điểm (lat_i, lon_i) . Do mỗi node trong quadtree có 4 node con tương ứng với 4 vùng không gian, vì vậy cứ mỗi lần lặp tìm kiếm số lượng bất động sản sẽ giảm dần theo cấp số nhân.
- Tìm kiếm bất động sản trong ô hợp lệ có khoảng cách nhỏ nhất tới (lat_i, lon_i)

Gọi p là số lượng vòng lặp tối đa để tìm kiếm bất động sản. Gọi M_i là số lượng bất động sản trong tập tìm kiếm sau vòng lặp thứ i . Ta có:

$$M_{i+1} = \frac{M_i}{4}, 0 \leq i \leq p-1, M_0 = M$$

Sau vòng lặp thứ k thì số lượng bất động sản trong mảng tìm kiếm là $M_k = \frac{M}{4^k}$

Điều kiện dừng của thuật toán tìm kiếm khi mỗi node chứa một lượng node $u \geq 1$ vừa đủ được cấu hình từ trước. Do đó ta có điều kiện sau:

$$M_p \geq u \geq 1 \iff \frac{M}{4^p} \geq 1 \iff p \leq \log_4 M$$

Do đó số lượng vòng lặp tối đa là $\log_4 M$, hay nói cách khác độ phức tạp của mỗi lần tìm kiếm sẽ là $O(\log_4 M)$ và độ phức tạp cho N câu lệnh query là $O(N \times \log_4 M)$.

So sánh với giải pháp thứ nhất, độ phức tạp của giải thuật đã giảm từ $O(N \times M)$ xuống còn $O(N \times \log_4 M)$. Điều này chứng tỏ sử dụng Quadtree là lựa chọn hợp lý và tối ưu cho hệ thống.

3.2 Framework và tính tự động của hệ thống trong lưu trữ và xây dựng mô hình

a, Feast Framework

Hiện nay có rất nhiều thư viện hỗ trợ xây dựng các mô hình học máy và học sâu vì vậy việc xây dựng một mô hình AI có thể chạy được ngay đã trở nên dễ dàng hơn. Tuy nhiên đối với những nhà khoa học dữ liệu, phân tích dữ liệu thì câu chuyện không chỉ dừng lại ở việc xây dựng mô hình mà còn quản lý mô hình, phân tích các features ảnh hưởng tới mô hình thế nào, lựa chọn features như thế nào trong pha sản phẩm. Hơn thế nữa các mô hình sau khi có dữ liệu mới thì mô hình sẽ thay đổi và hiển nhiên tập những features được dùng trong mô hình ở các phiên bản cũng sẽ khác nhau. Tuy nhiên như tôi thấy thì một quy trình MLOPs có quản lý các mô hình AI sau khi còn rất là khan hiếm ở các sản phẩm Việt Nam nói chung và các project nói riêng. Việc chỉ tập trung xây dựng mô hình tại một thời điểm và không có quy trình đánh giá các mô hình sau khi training xong với dữ liệu mới sẽ khiến cho các sản phẩm không có mức độ tin cậy và tính mới. Vì vậy ở đồ án này bên cạnh việc xây dựng giải thuật dự đoán giá bất động sản, tôi còn tập trung vào việc quản lý các mô hình sau khi được huấn luyện và đánh giá những mô hình đó.

Đồ án này giới thiệu một framework tương đối mới ở thị trường nói chung và ở thị trường Việt Nam nói riêng, đó là Feast Framework.

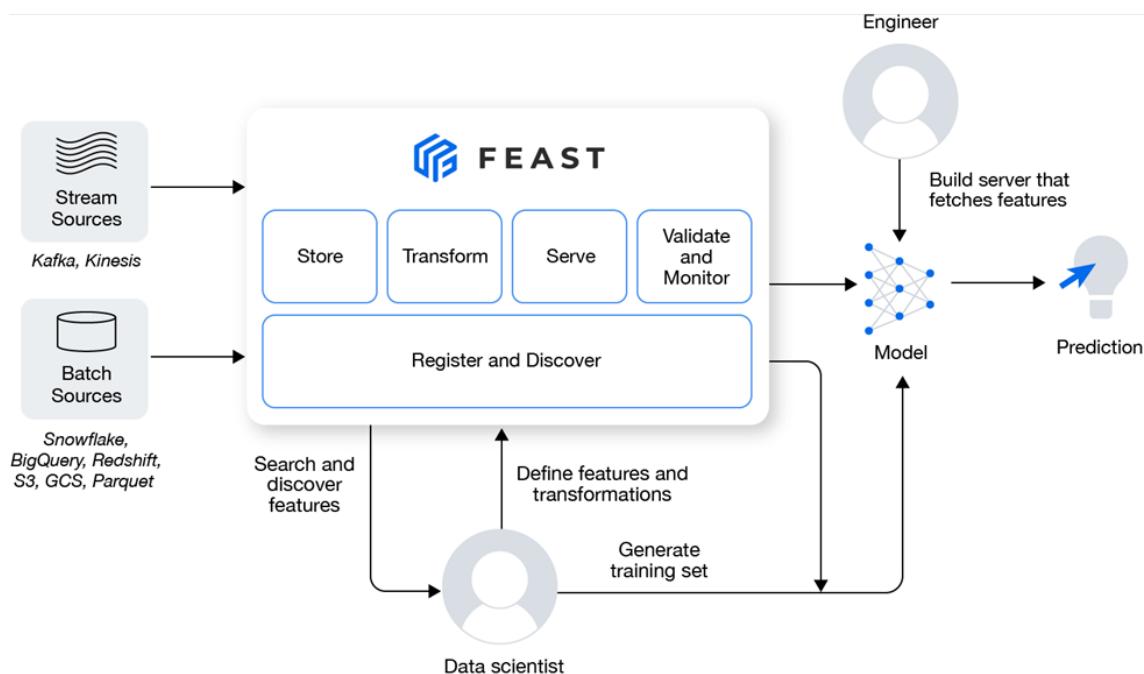
Hình vẽ 3.3 mô tả kiến trúc tổng quan của Feast. [9]

- Feast hỗ trợ nhiều kho dữ liệu đa dạng từ SQL(MySQL, PostgreSQL), noSQL(Apache Cassandra, Apache HBase), hệ thống lưu trữ đám mây (Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage), hệ thống phân phối dữ liệu (Apache Kafka, Apache Pulsar)
- Batch Sources có vai trò kết nối các nguồn dữ liệu khác nhau, bao gồm cơ sở dữ liệu, hệ thống lưu trữ đám mây. Bên cạnh batch sources chuyển đổi dữ liệu từ các nguồn thành định dạng phù hợp cho Feast Framework
- Stream Sources có vai trò giống với Batch Sources tuy nhiên thay vì chuyển đổi dữ liệu offline thì chuyển đổi dữ liệu realtime từ các nguồn như kafka, ...

- Store Component là nơi lưu trữ các dữ liệu sau khi được chuyển đổi từ các nguồn dữ liệu offline và realtime. Hỗ trợ truy vấn hiệu quả và API để cho datascience team có thể xây dựng mô hình hiệu quả hơn.
- Transform Component là transform feature hoặc engineer feature mới trước khi training mô hình học máy, học sâu
- Serve Component là module để cung cấp API lấy dữ liệu chứa feature gốc hoặc các feature mới dưới nhiều định dạng linh hoạt. Hơn thế nữa cung cấp cho người dùng một giao diện trực quan và dễ theo dõi hơn.

Với những ưu điểm trên thì Feast là một trong những framework thích hợp cho hệ thống BKPrice.

Hơn thế nữa trong mô hình dự đoán giá bất động sản, tôi đã thực hiện engineer các feature, do đó việc phân tích các features ảnh hưởng đến mô hình rất quan trọng

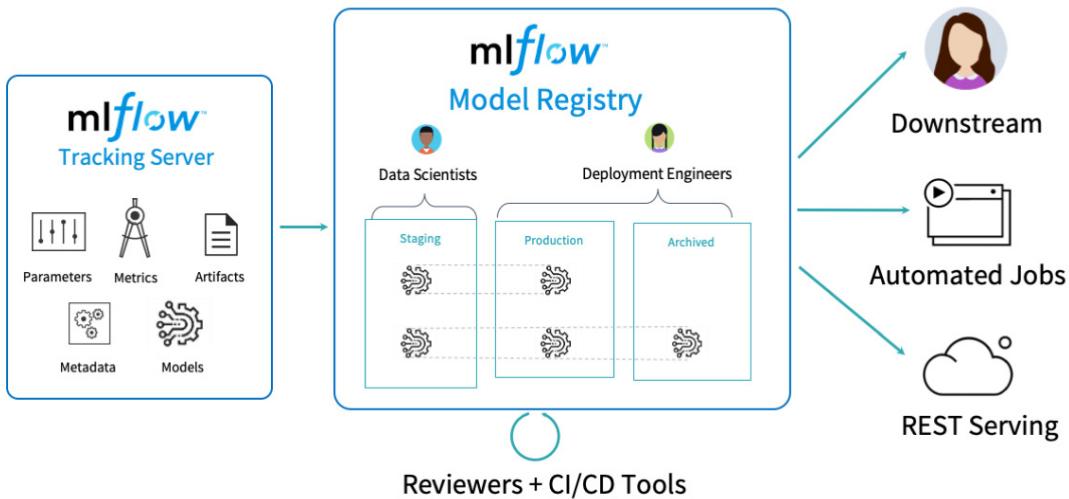


Hình 3.3: Feast Framework

b, MLflow Framework

MLFlow là một nền tảng mã nguồn mở nhằm đơn giản hóa vòng đời phát triển học máy (ML) [10]. Nó cung cấp một bộ công cụ và khung tổng thể toàn diện để quản lý và theo dõi quy trình phát triển ML từ đầu đến cuối, bao gồm thử nghiệm, giám sát, đánh giá và triển khai. MLFlow giúp các nhà khoa học dữ liệu và kỹ sư ML tập trung vào việc xây dựng và triển khai mô hình trong khi vẫn duy trì khả năng hiển thị, kiểm soát mô hình một cách khách quan hơn những cách quản lý mô

hình truyền thống trước đây.



Hình 3.4: Kiến trúc của MLflow

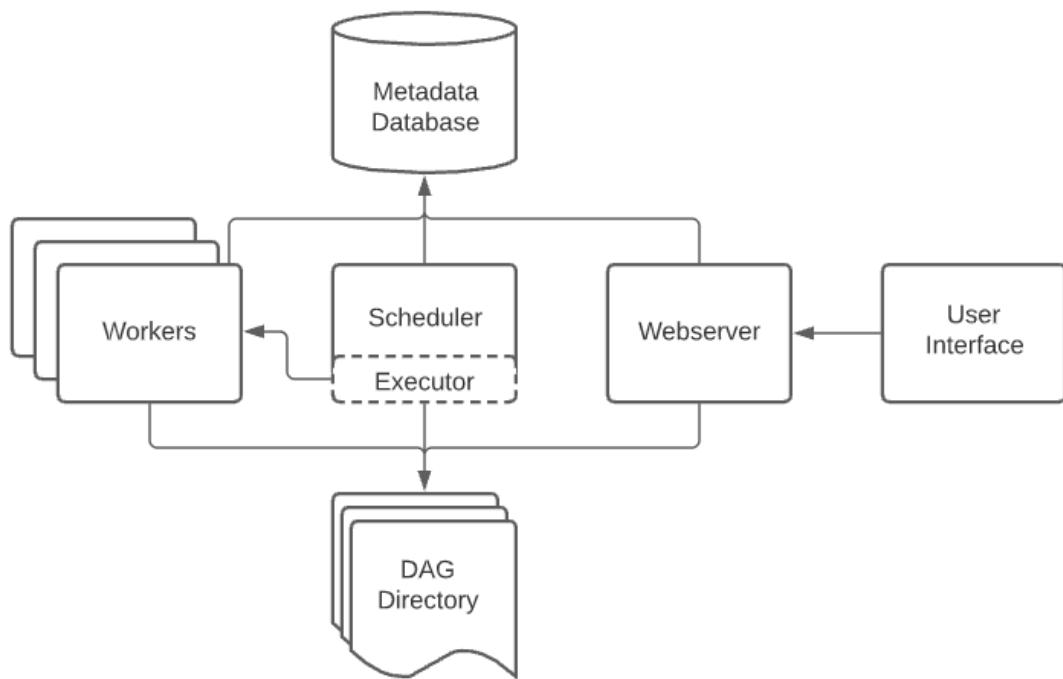
Thành phần chính của MLFlow: (i) MLFlow Tracking - Một API và giao diện người dùng để ghi dữ liệu về các thí nghiệm học máy và so sánh chúng bằng giao diện. (ii) MLFlow Projects - Định dạng đóng gói mã cho các lần chạy có thể tái tạo bằng Conda và Docker, vì vậy có thể dễ dàng chia sẻ mã ML với các bên liên quan. (iii) MLFlow Models - Định dạng đóng gói mô hình và công cụ cho phép dễ dàng triển khai cùng một mô hình (từ bất kỳ thư viện ML nào) để chấm điểm theo thời gian thực và hàng loạt trên các nền tảng như Docker, Apache Spark, Azure ML và AWS SageMaker. (iv) MLFlow Model Registry - Kho lưu trữ mô hình tập trung, bộ API và UI để quản lý cộng tác vòng đời đầy đủ của mô hình MLflow. Hình vẽ 3.4 mô tả đầy đủ các thành phần của MLflow.

Với những thành phần nền MLflow đã mang cho mình những ưu điểm vượt trội so với các công cụ quản lý mô hình trước đây. MLflow là một nền tảng mạnh mẽ và linh hoạt có thể giúp các nhà khoa học dữ liệu và kỹ sư ML đơn giản hóa vòng đời phát triển học máy. Do đó hệ thống BKPrice System đã dùng MLflow là nơi để quản lý vòng đời của các mô hình trí tuệ nhân tạo và giám sát các mô hình trí tuệ nhân tạo

c, Bộ lập lịch

Như tôi đã đề cập ở trên thì để đảm bảo được tính mới của hệ thống thì dữ liệu phải là yếu tố cần được chú ý đầu tiên. Để đảm bảo được dữ liệu cập nhật hằng ngày thì tôi sử dụng Airflow Framework [11].

Airflow Framework



Hình 3.5: Airflow architecture

Hình vẽ 3.5 mô tả kiến trúc tổng quan của Airflow Framework. Như chúng ta được thấy thì kiến trúc của airflow có các thành phần chính sau:

- DAG (Đồ thị Acyclic Directed): DAG là đơn vị cơ bản của airflow, đại diện cho một quy trình làm việc tự động. DAG bao gồm các tác vụ được kết nối với nhau theo thứ tự được cấu hình.
- Scheduler: Đây được gọi là bộ lập lịch trong Airflow. Nó có trách nhiệm theo dõi và kích hoạt chạy các tác vụ được lên lịch sẵn
- Executor: Đây là thành phần chịu trách nhiệm thực thi các tác vụ trong DAG. Executor như là bộ phân chia nhiệm vụ cho các workers. Khi các workers chạy xong các nhiệm vụ được giao thì sẽ báo cáo kết quả cho Executor. Khi đó Executor cập nhật trạng thái tác vụ trong DAG và thông báo cho Scheduler
- Webserver và User Interface: Airflow hỗ trợ người dùng giao diện để có thể quan sát trạng thái các tác vụ, thống kê tỷ lệ thành công, thất bại dễ dàng hơn.

Sử dụng airflow trong hệ thống BKPrice mang lại những lợi ích như sau:

- Tăng hiệu quả: Airflow giúp quá trình thu thập dữ liệu tự động theo lịch trình được cấu hình sẵn thay vì thủ công. Hơn thế nữa để huấn luyện mô hình một cách tự động thì lựa chọn airflow là một lựa chọn đúng. Do đó việc sử dụng airflow trong BKPrice giúp tiết kiệm thời gian và công sức

- Tính tin cậy: Do các DAG trong airflow được cấu hình một cách nhất quán vì vậy mà các tác vụ được thực hiện chính xác theo những gì mà hệ thống cấu hình
- Tăng khả năng mở rộng: BKPrice thu thập dữ liệu từ các nguồn dữ liệu bất động sản khác nhau vì vậy việc chạy song song các tác vụ là cần thiết. Và airflow cũng có thể giải quyết bài toán này.
- Dễ dàng theo dõi: Để tính tự động để có thể diễn ra mượt mà, quản lý tốt hơn. Thì việc sử dụng UI được cung cấp bởi airflow sẽ khiến cho việc xử lý các tác vụ khi thu thập dữ liệu bất động sản được giám sát tốt hơn.

Tóm lại, với những lợi ích trên thì chúng ta có thể thấy rằng việc áp dụng airflow để tăng tính tự động của hệ thống là một điều cần thiết

d, Blob Store

Như tôi đã đề cập thì BKPrice System sẽ thu thập dữ liệu từ các nguồn dữ liệu bất động sản. Việc các nguồn bất động sản có các định dạng khác nhau xảy ra rất thường xuyên. Hơn thế nữa ngoài dữ liệu là text thì dữ liệu thu thập có thể là HTML, Image. Vì vậy tôi chọn Blob Store làm nơi lưu trữ dữ liệu thô sau khi thu thập được từ các nguồn.

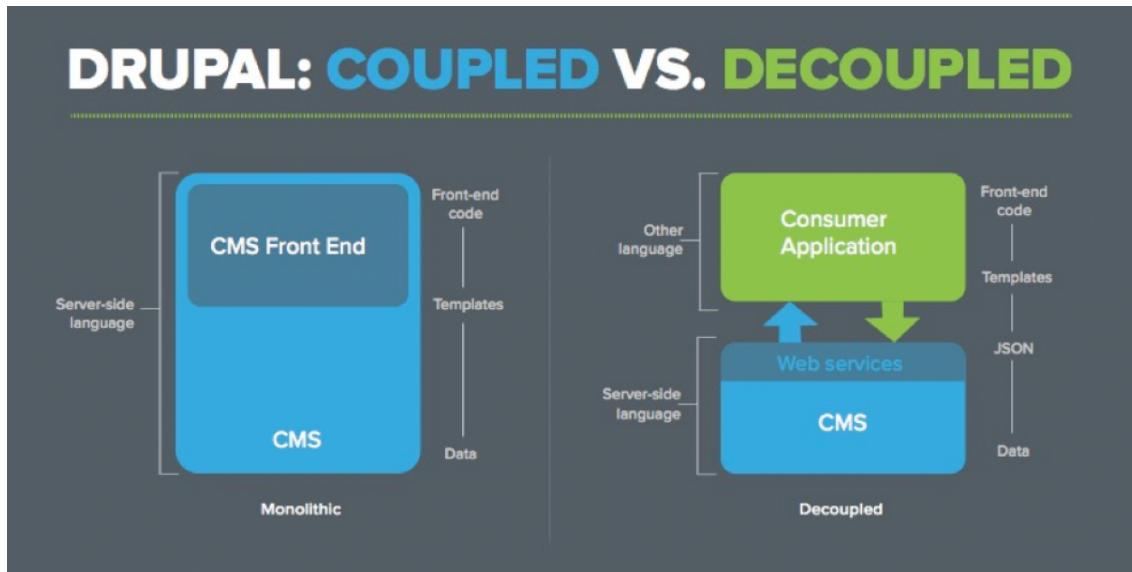
Blob Store là dịch vụ lưu trữ đám mây được thiết kế để lưu trữ dữ liệu phi cấu trúc: text, file, image, ... Và các thông tin này được lưu trữ dưới dạng file blob.

Dưới đây là một trong những lý do vì sao tôi lại chọn Blob Store làm nơi lưu trữ thông tin thu thập được cho BKPrice

- Lưu trữ dữ liệu dưới các format một cách linh hoạt
- Phục vụ nội dung: Như chúng ta được biết thì các thông tin thu thập được rất ít khi thay đổi (tĩnh). Với các hình ảnh có kích thước lớn, ... thì để có thể loading một cách mượt mà hơn thì blob store là lựa chọn tốt cho các service tìm kiếm thông tin bất động sản sau này
- Do thông tin bất động sản được thu thập hằng ngày nên kích thước tăng lên rất nhanh. Vì vậy Blobstore giúp BKPrice có thể mở rộng tốt hơn theo chiều ngang để đáp ứng nhu cầu lưu trữ.

e, Luồng phân phối dữ liệu thời gian thực

Một trong những điểm quan trọng khi xây dựng một hệ thống có tính khả mở đó chính là tách biệt các module trong hệ thống và khiến cho các module không được phục thuộc nhau qua nhiều. Những kiến trúc như vậy được gọi là decoupled architecture [12].



Hình 3.6: Sự khác nhau giữa coupled và decoupled architecture

Hình vẽ 3.6 mô tả được sự khác nhau giữa kiến trúc coupled và decoupled. Thay vì các module liên kết với nhau ở kiến trúc coupled thì có sự tách biệt rõ ràng giữa các module ở kiến trúc decoupled. Điểm yếu lớn nhất của các kiến trúc coupled đó chính là khi một module này có vấn đề gì thì sẽ ảnh hưởng tới module khác. Trong khi đó, đối với BKPrice System thì có rất nhiều quá trình mà output của module này sẽ là input của module kia. Vì thế trong BKPrice System thì luôn đảm bảo các service luôn được tách rời.

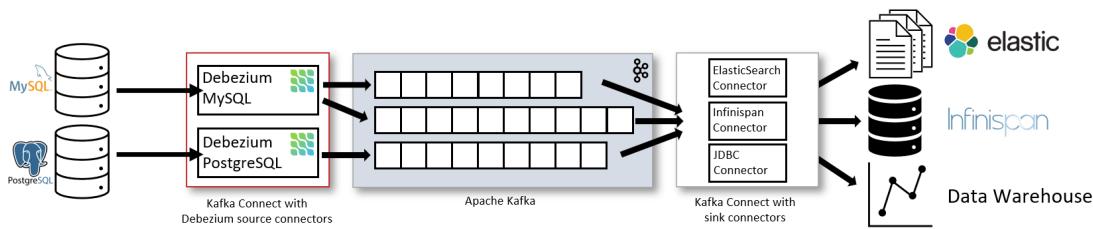
Do vậy tôi sẽ dùng Kafka Framework trong đồ án này [13]. Kafka Framework là một nền tảng xử lý luồng dữ liệu và phân phối những luồng dữ liệu. Điều này là cần thiết trong hệ thống BKPrice bởi vì từ luồng thu thập dữ liệu, đến làm sạch dữ liệu, đến xử lý dữ liệu, đến xây dựng mô hình,..., để đảm bảo tại các bước việc một service bị lỗi không ảnh hưởng đến luồng xử lý thì cần phải có một nơi để lưu trữ những thông tin đó. Lý do là khi mà một service gặp lỗi thì có thể có một nơi có thể lấy lại thông tin đang xử lý để và tiếp tục xử lý. Điều đó cũng đảm bảo được rằng hệ thống BKPrice là một hệ thống tin cậy và có cơ chế xử lý lỗi.

f, Data Capture Change

Khi dữ liệu mới thu nhập, xử lý và lưu trữ vào cơ sở dữ liệu. Để hệ thống có thể bắt được sự thay đổi dữ liệu, qua đó xây dựng mô hình và huấn luyện một cách tự động thì đồ án này tôi giới thiệu một Framework có tên là Debezium.

Debezium là một công cụ mã nguồn mở được sử dụng để bắt giữ thay đổi dữ liệu (CDC) từ các cơ sở dữ liệu quan hệ và truyền phát nó đến các hệ thống khác. Nó hoạt động bằng cách theo dõi các thay đổi được thực hiện đối với dữ liệu trong cơ sở dữ liệu và sau đó tạo ra các sự kiện mô tả những thay đổi đó. Các sự kiện này

sau đó có thể được tiêu thụ bởi các hệ thống khác để cập nhật dữ liệu của chúng hoặc thực hiện các hành động khác.



Hình 3.7: Debezium Architecture

Hình ảnh 3.y mô tả kiến trúc tổng quan của Debezium [14]:

- Connector Component: Gồm các plugin kết nối Debezium với các cơ sở dữ liệu cần capture data change
- Kafka Connect: là cầu nối vận chuyển realtime để kết nối Debezium với các hệ thống khác, chẳng hạn như Apache Kafka, Amazon Kinesis và Google Pub/Sub.

Điểm nổi bật của Debezium:

- Hỗ trợ nhiều cơ sở dữ liệu: Debezium hỗ trợ nhiều loại cơ sở dữ liệu quan hệ phổ biến, bao gồm MySQL, PostgreSQL, Oracle, SQL Server và MariaDB.
- Có thể mở rộng: Debezium có thể mở rộng cao để xử lý lượng lớn dữ liệu thay đổi.
- Dễ sử dụng: Debezium cung cấp các công cụ và API dễ sử dụng để triển khai và sử dụng.
- Linh hoạt: Debezium có thể được cấu hình để chuyển đổi và định dạng dữ liệu thay đổi theo nhiều cách khác nhau.
- Mã nguồn mở: Debezium là mã nguồn mở và miễn phí sử dụng.

Với những điểm nổi bật trên thì có nhiều ứng dụng Debezium [15] trong các sản phẩm thực tế:

- Cập nhật dữ liệu theo thời gian thực: Debezium có thể được sử dụng để cập nhật dữ liệu trong các hệ thống khác theo thời gian thực, chẳng hạn như kho dữ liệu, hệ thống phân tích và ứng dụng web.
- Đồng bộ hóa dữ liệu: Debezium có thể được sử dụng để đồng bộ hóa dữ liệu giữa nhiều cơ sở dữ liệu.
- Kiểm tra tính toàn vẹn dữ liệu: Debezium có thể được sử dụng để kiểm tra

tính toàn vẹn dữ liệu bằng cách theo dõi các thay đổi được thực hiện đối với dữ liệu trong cơ sở dữ liệu.

- Khám phá và giám sát dữ liệu: Debezium có thể được sử dụng để theo dõi các thay đổi được thực hiện đối với dữ liệu trong cơ sở dữ liệu cho mục đích kiểm toán.
- Phát triển ứng dụng theo thời gian thực: Debezium có thể được sử dụng để phát triển các ứng dụng theo thời gian thực dựa trên dữ liệu thay đổi.

Tóm lại, việc sử dụng Debezium là một điều cần thiết. Điều đó khiến cho hệ thống tự động và mượt mà hơn.

Tổng kết: Qua chương 3, tôi đã phân tích những lý thuyết toán học được sử dụng trong nghiên cứu. Bên cạnh là phân tích những framework được sử dụng và lý do lựa chọn những framework đó trong quá trình phát triển hệ thống BKPrice.

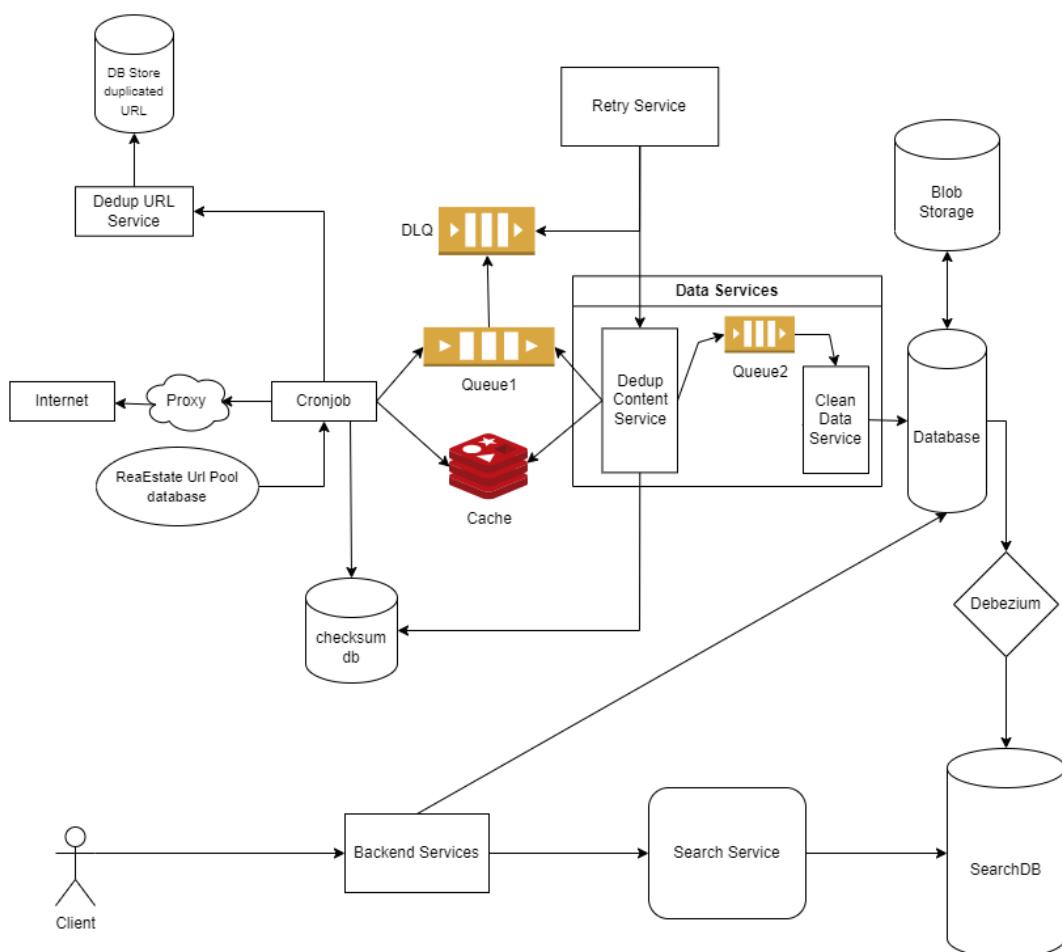
CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

Ở chương thứ 4 tôi đề cập một góc nhìn tổng quan hơn về những giải pháp được dùng trong hệ thống, bên cạnh đó nêu ra những nguyên nhân vì sao công nghệ đó được áp dụng. Chương này sẽ trình bày một cách tổng thể về hệ thống, các quy trình trong hệ thống từ mức tổng quan đến mức chi tiết nhất. Hơn thế nữa nêu rõ các giải thuật và tiến trình hoạt động của hệ thống. Từ đó giải quyết được các vấn đề mà đã đề cập ở các chương trên

4.1 Tổng quan giải pháp

Phần này mô tả các quy trình tổng quan của giải pháp. Mục tiêu của phần này là cho người đọc một cái nhìn tổng thể về giải pháp. Để cho dễ hiểu thì nên có một biểu đồ mô tả luồng hoạt động của giải pháp đề xuất.

4.2 Hệ thống thu thập dữ liệu tự động



Hình 4.1: BKPrice Data System

4.2.1 Data Components

Việc sử dụng mô hình AI trên những dữ liệu thu thập được và không cập nhật lại dữ liệu thường xuyên khiến cho mô hình trí tuệ nhân tạo trở nên lỗi thời so với dữ liệu mới. Điều này xảy ra rất thường xuyên:

- Dữ liệu được huấn luyện cho mô hình GPT3 là bộ dữ liệu khổng lồ bao gồm văn bản và mã từ internet, sách, bài báo khoa học, v.v. cho đến tháng 9/2020. Dữ liệu không được cập nhật mới nhất bởi vì chi phí huấn luyện mô hình đắt đỏ về nhân lực và tiền bạc
- Khảo sát ở phần trên cho thấy mô hình của biggee không được cập nhật thường xuyên khiến việc dự đoán giá bất động sản không thay đổi và không đáng tin cậy.

Do đó trong pha xử lý đầu tiên, BKPrice System tập trung vào việc xây dựng luồng thu thập và xử lý dữ liệu. Qua đó dữ liệu huấn luyện mô hình sẽ được cập nhật và trở thành nguồn cung cấp tin cậy cho mô hình AI ở pha tiếp theo.

Hình vẽ 4.1 mô tả pha đầu tiên của hệ thống - Thu thập và xử lý dữ liệu - Gọi chung là Data System. Ở phần này ta sẽ đi qua các thành phần và các service cơ bản của hệ thống dữ liệu.

- Realestate URL Pool database: Database lưu trữ danh sách url cần được thu thập thông tin. Đối với các nguồn dữ liệu bất động sản thì sẽ có từng script để lấy danh sách url riêng.
- Proxy: Việc thu thập dữ liệu ở mức độ nhiều và lớn sẽ gặp một vấn đề chính là tốc độ và privacy. Hầu hết các trang web lớn đều có cơ chế chặn IP đối với những IP truy cập cao bất ngờ vào hệ thống. Hơn thế nữa việc truy cập nhiều vào hệ thống sẽ khiến cho tốc độ truy cập các request về sau trở nên chậm hơn. Do đó tôi sử dụng proxy ở đây có vai trò cân bằng tải khi thu thập dữ liệu và ẩn dấu IP khi thu thập dữ liệu. Qua đó việc thu thập dữ liệu trở nên dễ dàng hơn.
- Cronjob: Thành phần này được xem như là một bộ lập lịch (Job Scheduler). Tôi sử dụng airflow framework đã được giới thiệu ở chương trước để phục vụ cho việc lên lịch thu thập dữ liệu.
- Queue: Ở trong BKPrice Data System, tôi sử dụng Kafka framework để xây dựng 3 queue có tên là: DLQ, Queue1, Queue2
- Database: Ở trong hệ thống có các database là DB Store duplicated URL, Realestate URL Pool database, Blob Storage, Checksum Database, Main DataBase,

Search Database (Elastic Search - Có thể dùng cho việc cung cấp tìm kiếm data cho các bên)

- Data Service: Các service con trong Data System bao gồm Dedup URL Service, Retry Service, Dedup Content Service, Clean Data Service, Search Service, Backend Service
- Cache: Tôi sử dụng Redis Framework

4.2.2 Data Pipeline

Ở phần trên đề cập tới các thành phần, các service ở bên trong BKPrice Data System. Đến phần này sẽ nói rõ hơn luồng dữ liệu được xử lý giữa các thành phần và service.

a, Thu thập dữ liệu

Module Cronjob có vai trò lập lịch để thu thập dữ liệu từ các nguồn bất động sản:

- batdongsan.com
- muaban.net
- meyland.com

Hình vẽ 4.2, 4.3, 4.4 mô tả dữ liệu gốc sau khi lấy được từ các nguồn dữ liệu: muaban.net, meyland.com, batdongsan.com.

Như chúng ta thấy thì mỗi nguồn dữ liệu sẽ có định dạng dữ liệu khác nhau, cách thức cấu trúc mỗi bản ghi cũng sẽ khác nhau. Vì vậy, trong pha thu thập dữ liệu sẽ có service để hậu xử lý và đưa các bản ghi từ các nguồn dữ liệu về một định dạng thống nhất.

Dưới đây tôi sẽ trình bày cách thức hệ thống dữ liệu thu thập từ 3 nguồn dữ liệu nêu trên.

- Đối với nguồn trang batdongsan.com, việc lấy dữ liệu bằng cách FETCH API hoàn toàn không khả thi. Do đó, tôi quyết định thu thập dữ liệu bằng cách sử dụng web driver tương tác trực tiếp và thu thập nguồn HTML. Hình vẽ 4.4 mô tả mẫu một bản ghi sau khi thu thập được từ batdongsan.com
- Đối với nguồn trang meyland.com, chúng ta có thể lấy dữ liệu bằng cách FETCH API. Tuy nhiên để tránh thu thập dữ liệu từ bot, hệ thống meyland đã chèn tự động một khóa ngẫu nhiên vào endpoint. Vì vậy tôi quyết định sẽ lấy tự động khóa ngẫu nhiên này và thu thập dữ liệu bằng cách FETCH API. Hình vẽ 4.3 mô tả mẫu một bản ghi sau khi thu thập từ meyland.com

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

```

    {
        "id": 69331418,
        "user_id": 3343311,
        "city_id": 15,
        "district_id": 193,
        "category_id": 33,
        "subcategory_id": 169,
        "category_ids": [
            33,
            169,
            43,
            2522
        ],
        "property_type": 43,
        "property_subtype": 2522,
        "brand_id": 0,
        "title": "Nhà ở trung tâm thành phố , gần chợ siêu thị",
        "covers": [
            "https://cloud.muaban.net/images/thumb-glist/2024/06/22/096/a9a6bbd8746f4e56805abd59af638496.jpg"
        ],
        "total_images": 13,
        "price": 1800000000,
        "price_display": "1 tỷ 800 triệu",
        "url": "/bat-dong-san/ban-nha-o-hem-ngo/nha-o-trung-tam-thanh-pho-gan-cho-sieu-thi-id69331418",
        "service_id": 1,
        "publish_at": "2024-06-23T00:01:01.862+07:00",
        "publish_display": "Hôm nay",
        "location": "Phường Hải Châu I, Quận Hải Châu",
        "is_company": false,
        "attributes": [
            {
                "value": "20 m2"
            },
            {
                "value": "2 PN"
            },
            {
                "value": "2 WC"
            }
        ],
        "sort": 2406232000101,
        "job_sort": 0
    },
}

```

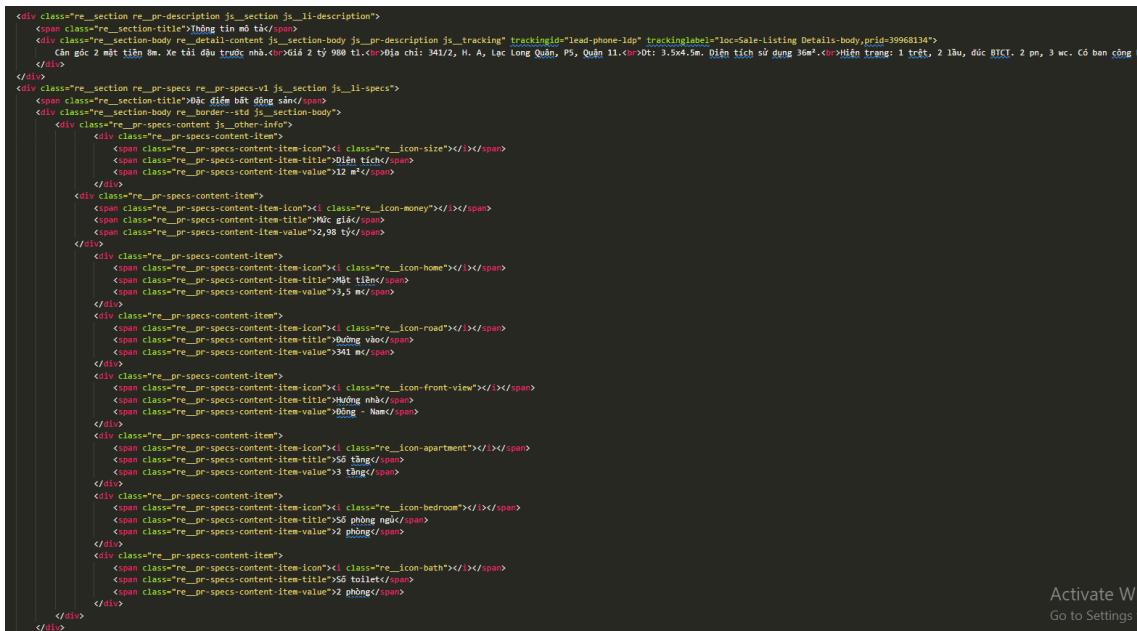
Hình 4.2: Dữ liệu từ trang muaban.net

```

    {
        "_id": "667ff7d9df59237c24c0fb8f",
        "title": "Bán đất 132m2 đồng thu thiem ,P.bình trung đông,Quận 2",
        "need": "can_ban",
        "creatorType": 0,
        "slug": "ban-dat-132m2-dong-thu-thiem-p-binh-trung-dong-quan-2-1719662573457",
        "subscriptionPriority": 1,
        "area": 132,
        "bedroom": null,
        "bathroom": null,
        "facade": null,
        "direction": [-], // 1 item
        "publishedDate": "2024-06-29T12:02:53.000Z",
        "images": [-], // 3 items
        "videos": [],
        "locations": [
            {
                "cityName": "Hồ Chí Minh",
                "districtName": "Thủ Đức",
                "wardName": null,
                "streetName": null,
                "address": "Bình Trung Đông, Phường Bình Trung Đông (Quận 2 cũ), Thành phố Thủ Đức, Tp Hồ Chí Minh",
                "projectName": null
            }
        ],
        "unitPriceLabel": "90,91 Tr/m2",
        "priceLabel": "12 Tỷ",
        "category": [-], // 2 items
        "typeOfHouse": [-], // 1 item
        "content": "<p>khu Đồng Thủ Thiêm <br /> Bán nhanh 1 nền nhà phố Khu Đồng Thủ Thiêm liền p.BÌTĐ (Q2) TP. THỦ ĐỨC <br /> -Diện tích: 132m2 ( 6 x 22) <br /> -Hướng: Đông Nam <br /> • Mã Nền : DDX <br /> -Mặt tiền Đường Trước đất 12m có lề dg thông ra đỗ xuân hợp nằm trong khu dân cư VIP toàn biệt thự xung quanh an ninh sạch sẽ . <br /> • Xây Dựng: 1 Hầm + 1 trệt + 2 lầu + Sân thượng. <br /> • Vị trí cực kì thuận tiện di lại . <br /> • Hiện trạng: Đất Trống khách mua và lên chủ đầu tư xây ngay. <br /> • Pháp lý: Sở hữu công ra tên cá nhân. <br /> Bán giao 2006 <br /> Khu dân cư Đồng Thủ Thiêm <br /> Chủ đầu tư: <br /> CÔNG TY CP ĐẦU TƯ THỦ THIỆM <br /> </p>",
        "address": "Thủ Đức, Hồ Chí Minh",
    }
}

```

Hình 4.3: Dữ liệu từ trang meeyland.com



```

<div class="re_section re_pr-description js_section js_li-description">
    <span class="re_section-title">Thông tin mă tâp</span>
    <div class="re_section-body re_detail-content js_section-body js_pr-description js_tracking" trackingid="lead-phone-ldp" trackinglabel="loc=Sale-Listing Details-body,prid=39968134">
        Căn góc 2 mặt tiền 8m. Xe tải đậu trước nhà.<br>Giá 2 tỷ 998 tr.Địa chỉ: 341/2, H. A, Lạc Long Quân, P5, Quận 11.<br>Dt: 3.5x4.5m. Diện tích sử dụng 36m2.<br>Hiện trạng: 1 trệt, 2 lầu, dúc BTCT. 2 pn, 3 wc. Có ban công b
    </div>
</div>
<div class="re_section re_pr-specs js_section js_li-specs">
    <span class="re_section-title">Đặc điểm bất động sản</span>
    <div class="re_section-body re_pr-specs-content js_section-body">
        <div class="re_pr-specs-content-item">
            <span class="re_pr-specs-content-item-icon"><i class="re_icon-size"></i></span>
            <span class="re_pr-specs-content-item-title">Diện tích</span>
            <span class="re_pr-specs-content-item-value">12 m2</span>
        </div>
        <div class="re_pr-specs-content-item">
            <span class="re_pr-specs-content-item-icon"><i class="re_icon-money"></i></span>
            <span class="re_pr-specs-content-item-title">Mức giá/k&gt;</span>
            <span class="re_pr-specs-content-item-value">2,98 tỷ</span>
        </div>
        <div class="re_pr-specs-content-item">
            <span class="re_pr-specs-content-item-icon"><i class="re_icon-home"></i></span>
            <span class="re_pr-specs-content-item-title">Mật độ</span>
            <span class="re_pr-specs-content-item-value">3,5 m2/m</span>
        </div>
        <div class="re_pr-specs-content-item">
            <span class="re_pr-specs-content-item-icon"><i class="re_icon-road"></i></span>
            <span class="re_pr-specs-content-item-title">Hướng view</span>
            <span class="re_pr-specs-content-item-value">Đông - Nam</span>
        </div>
        <div class="re_pr-specs-content-item">
            <span class="re_pr-specs-content-item-icon"><i class="re_icon-front-view"></i></span>
            <span class="re_pr-specs-content-item-title">Hướng nh&gt;</span>
            <span class="re_pr-specs-content-item-value">Đông - Nam</span>
        </div>
        <div class="re_pr-specs-content-item">
            <span class="re_pr-specs-content-item-icon"><i class="re_icon-apartment"></i></span>
            <span class="re_pr-specs-content-item-title">Số tầng</span>
            <span class="re_pr-specs-content-item-value">3 tầng</span>
        </div>
        <div class="re_pr-specs-content-item">
            <span class="re_pr-specs-content-item-icon"><i class="re_icon-bedroom"></i></span>
            <span class="re_pr-specs-content-item-title">Số phòng ngủ</span>
            <span class="re_pr-specs-content-item-value">2 phòng</span>
        </div>
        <div class="re_pr-specs-content-item">
            <span class="re_pr-specs-content-item-icon"><i class="re_icon-bath"></i></span>
            <span class="re_pr-specs-content-item-title">Số toilet</span>
            <span class="re_pr-specs-content-item-value">2 phòng</span>
        </div>
    </div>
</div>

```

Activate Wi
Go to Settings

Hình 4.4: Dữ liệu từ trang batdongsan.com

- Đối với nguồn dữ liệu muaban.net, việc lấy dữ liệu khá dễ dàng bằng cách FETCH trực tiếp API. Hình vẽ 4.2 mô tả mẫu một bản ghi sau khi thu thập từ meeyland.

Như chúng ta được biết thì việc trùng lắp bất động sản ở các nguồn xảy ra khá thường xuyên sở dĩ là vì các nhà môi giới không chỉ đăng tin lên một trang mà còn nhiều trang khác. Hơn thế nữa việc trùng lắp bất động sản ở cùng một nguồn cũng chính là một vấn đề hay gặp.

Để giải quyết vấn đề này trong quá trình thu thập dữ liệu thì tôi sử dụng giải pháp sau:

- DB Store duplicate URL: Cơ sở dữ liệu lưu trữ những url đã được thu thập từ các nguồn tin và không có url nào là trùng lắp. Bên cạnh đó Dedup URL Service là dịch dùng để kiểm tra tính trùng lắp URL trong quá trình thu thập. Nếu url không trùng lắp thì lưu trữ vào cơ sở dữ liệu và tiến trình thu thập dữ liệu
- Sử dụng checksumdb để lưu trữ mã hash là content hoặc title của bài đăng bất động sản. Checksumdb được thiết kế như cơ sở dữ liệu 1 cột và cột dữ liệu đó là unique. Vì vậy xác suất cao là không thể tồn tại 2 bài đăng có cùng nội dung được lưu trữ trong cơ sở dữ liệu. Dedup Content Service có nhiệm vụ kiểm tra nếu có trùng lắp thì sẽ không xử lý dữ liệu của bài đăng bất động sản trùng lắp đó.

b, Xử lý lỗi

Tuy nhiên việc lưu dữ liệu vào queue với khối lượng lớn thông tin các bất động sản như vậy sẽ khiến cho queue khó kiểm soát hơn. Do đó để vừa đảm bảo được tính tự động và tính ổn định của hệ thống thì BKPrice System sử dụng thêm redis cache. Kafka queue chỉ nhận nhiệm vụ lưu trữ và phân phối message, Dedup Content Service sẽ consume message đó và lấy dữ liệu từ cache và xử lý.

Bên cạnh đó có một vấn đề lớn hơn đó chính là với dữ liệu được thu thập từ nhiều nguồn thì việc các định dạng dữ liệu sẽ rất đa dạng Do đó không tránh khỏi Dedup Content Service gặp lỗi logic. Vì vậy để kịp thời phân tích những lỗi như vậy thì tôi có giải pháp như sau:

- DLQ: Kafka queue lưu trữ những message được xử lý lỗi ở Dedup Content Service. Do đó dễ dàng trong việc phân tích message lỗi đó và điều chỉnh lại Dedup Content Service.
- Retry Service: Sau khi phân tích thì retry service sẽ đảm bảo việc gọi lại Dedup Content Service là thực thi lại các message lỗi.

4.2.3 Xử lý và lưu trữ dữ liệu

Do định dạng dữ liệu khác nhau giữa các nguồn dữ liệu vì vậy việc làm sạch và xử lý dữ liệu giữa các nguồn sẽ được thực hiện trong Clean Data Service.

Dưới đây tôi sẽ trình bày một vài điểm lưu ý trong bước làm sạch và xử lý dữ liệu thô sau khi thu thập.

a, Xử lý và tích hợp dữ liệu

Trước khi quyết định dữ liệu được làm sạch như thế nào thì việc quyết định những thông tin nào được sử dụng nên được xem xét đầu tiên. Với dữ liệu được thu thập thì tôi quyết định trích xuất những thông tin sau:

- Số tầng của bất động sản
- Số phòng tắm và số phòng ngủ
- Loại của bất động sản: Biệt thự hay nhà riêng, ...
- Vị trí của ngôi nhà: Nhà mặt tiền, nhà mặt ngõ 1 ô tô tránh, 2 ô tô tránh, ...
- Độ rộng của ngõ trước nhà
- Địa chỉ của ngôi nhà
- Thông tin mô tả ngôi nhà
- Thông tin latitude và longitude

```
{
    "CITY": "Hà Nội",
    "DISTRICT": "Hai Bà Trưng",
    "STREET": "Đê Trần Khát Chân",
    "URI_REQ": "de-tran-khat-chan-bach-dang-hai ба trung ha noi",
    "LNG": "105.863419234662",
    "LAT": "21.0083065541043",
    "WARD": "Bách Đằng"
},
{
    "CITY": "Hà Nội",
    "DISTRICT": "Hai Bà Trưng",
    "STREET": "Đầm Trầu",
    "URI_REQ": "dam-trau-bach-dang-hai ба trung ha noi",
    "LNG": "105.8655045",
    "LAT": "21.0120352",
    "WARD": "Bách Đằng"
},
{
    "CITY": "Hà Nội",
    "DISTRICT": "Hai Bà Trưng",
    "STREET": "Giải Phóng",
    "URI_REQ": "giai-phong-bach-khoa-hai ба trung ha noi",
    "LNG": "105.8414191",
    "LAT": "21.0052678",
    "WARD": "Bách Khoa",
    "NAME_OSM": "Đường Giải Phóng, Phường Bách Khoa, Quận Hai Bà Trưng, Hà Nội, 10999, Việt Nam",
    "ID_OSM": 601455488
},
{
    "CITY": "Hà Nội",
    "DISTRICT": "Hai Bà Trưng",
    "STREET": "Hàm Kim Liên",
    "URI_REQ": "ham-kim-lien-bach-khoa-hai ба trung ha noi",
    "LNG": "105.8428374",
    "LAT": "21.0076916",
    "WARD": "Bách Khoa",
    "NAME_OSM": "Hàm Kim Liên, Phường Bách Khoa, Quận Hai Bà Trưng, Hà Nội, 10999, Việt Nam",
    "ID_OSM": 163843403
}
}
```

Hình 4.5: Thông tin đường phố Việt Nam

- Kích thước của bất động sản
- Giá trị của bất động sản.
- Thông tin thời gian đăng bán bất động sản

Như chúng ta thấy định dạng dữ liệu của 3 nguồn bất động sản trên chính là: HTML và JSON Object. Đối với định dạng HTML thì tôi dùng BeautifulSoup để trích xuất các trường thông tin trên thành JSON Object.

Tuy nhiên có một khó khăn ở đây chính là thông tin lat, lon và địa chỉ của từng nguồn sẽ khác nhau. Và có nguồn có, có nguồn không có. Vì vậy để xử lý vấn đề này thì tôi sử dụng một nguồn thông tin danh sách đường phố tại Việt Nam. Hình vẽ 4.5 mô tả các thông tin về đường phố Việt Nam mà chúng tôi sử dụng để matching bản ghi thu thập được với đường phố Việt Nam: thông tin quận (DISTRICT), thông tin phường (WARD), thông tin đường (STREET), thông tin vị trí địa lý (LAT, LNG)

Hình vẽ 4.6 mô tả định dạng format cuối cùng của dữ liệu. Định dạng dữ liệu này chính là định dạng dữ liệu được lưu vào cơ sở dữ liệu.

Sau khi dữ liệu được lưu trữ vào cơ sở dữ liệu, debezium sẽ trigger sự kiện tiếp

```

{
  "houseInfo": {
    "value": {
      "numberOffLoops": 1,
      "numberBedRooms": 0
    }
  },
  "propertyBasicInfo": {
    "landType": {
      "value": "residentialLand"
    },
    "accessibility": {
      "value": "notInTheAlley"
    },
    "frontRoadWidth": {
      "value": 0
    }
  },
  "address": {
    "value": {
      "addressDetails": "",
      "street": "Linh Nam",
      "ward": "Linh Nam",
      "district": "Huang Mai",
      "city": "Ha Noi",
      "country": "Viet Nam"
    }
  },
  "description": {
    "value": "<p>LINH NAM BÁN 46M ĐẤT, MT 4,5M - ĐẤT LÔ GÓC - 2 MẶT NGÕ THÔNG Ô TÔ TRÁNH DỪ"
  },
  "geolocation": {
    "value": {
      "latitude": {
        "value": "20.983566"
      },
      "longitude": {
        "value": "105.875573"
      }
    }
  },
  "typeOfRealEstate": {
    "value": "privateLand"
  },
  "facade": {
    "value": "twoSideOpen"
  },
  "houseDirection": {
    "value": "tay"
  },
  "landSize": {
    "value": 46
  },
  "price": {
    "value": 6.25
  },
  "unitPrice": {
    "value": "billion"
  }
},
  "crawlInfo": {
    "id": "1059102087",
    "source": "meeland",
    "time": "2024-06-22T08:11:00"
  }
}

```

Hình 4.6: Định dạng cuối cùng của dữ liệu

tục huấn luyện mô hình với dữ liệu mới. Ở trong nghiên cứu này, tôi cấu hình với 1000 record mới sẽ thực hiện lại quá trình huấn luyện mô hình dự đoán giá.

4.3 Hệ thống dự đoán giá bất động sản tin cậy

Ở phần trên đã mô tả được quy trình thu thập dữ liệu, xử lý dữ liệu và xử lý lỗi khi cần thiết. Ở trong phần này sẽ phân tích các giải thuật định giá đã có và đề xuất 1 giải thuật định giá phù hợp hơn. Điều đó ảnh hưởng đến độ tin cậy của hệ thống định giá bất động sản.

4.3.1 Giải thuật định giá bất động sản

a, Previous studies

- Nghiên cứu [16] sử dụng Mô hình XGBoost để dự đoán giá nhà ở Karachi City, Pakistan. Mô hình sử dụng các thông số cơ bản của ngôi nhà để dự đoán

	Name	Type	Description
1	Property_id	Numerical	Different types of properties i.e. House and flat
2	Location_id	Numerical	Locations or areas where the property situated
3	Property_type	Categorical	House type, Flat, Portion etc
4	Price	Numerical	House price (Prediction outcome)
5	Location	Categorical	House location
6	City	Categorical	City located
7	Province	Categorical	Where the city (Karachi) is located
8	Latitude	Categorical	House latitude
9	Longitude	Categorical	House longitude
10	Baths	Numerical	Number of bathrooms
11	Area	Categorical	House Area
12	Bedrooms	Numerical	Number of Bedrooms
13	Area Size	Numerical	House area
14	Area Category	Categorical	House area category

Hình 4.7: Danh sách các đặc trưng sử dụng trong mô hình

Các thuộc tính trên được apply thẳng vào mô hình để training. Mô hình sau khi training sẽ gặp một trong những vấn đề sau:

- Mô hình không có tri thức về mối liên hệ giữa các ngôi nhà trong vùng với nhau. Do đó vấn đề gặp phải đó là biến động của predictions trong vùng sẽ cao
- Mô hình không có tri thức về sự thay đổi theo thời gian khiến cho mô hình outofdate so với dữ liệu bất động sản mới
- Tác giả sử dụng đơn mô hình do đó bên cạnh variance của predictions

trong vùng cao mà variance predictions cho từng sample cao. Tôi sẽ phân tích và đưa ra số liệu ở chương thực nghiệm

- Nghiên cứu [17] sử dụng mô hình Stacked Generalization Regression để dự đoán giá nhà ở Beijing, Trung Quốc. Mô hình sử dụng các thông số cơ bản của ngôi nhà để dự đoán. Tuy nhiên giải pháp này tối ưu được một vài nhược điểm của giải pháp trên:
 - Có thêm thông tin về thời gian nhưng chưa tạo feature từ thông tin thời gian
 - Mô hình staking tổng quát hóa hơn giải pháp trước và đã giảm được variance trong dự đoán so với phương pháp trên

Attribute Name	Data Type	Description
Lng	float64	Longitude of the house
Lat	float64	Latitude of the house
district	int64	District (District 1- District 13)
distance	float64	Distance to the center of Beijing
age	int64	Age of the house
square	float64	Area of the house
communityAverage	float64	Average housing price of the community
followers	int64	Number of followers
tradeTime	int64	Trade Time (2002-2018)
livingRoom	int64	Number of bedrooms
floorType	object	Floor height relative to the building
floorHeight	int64	Floor height
buildingType	int64	Building Type
renovationCondition	int64	Renovation Condition
buildingStructure	int64	Building Structure
ladderRatio	float64	Ratio between population and number of elevators of the floor
elevator	int64	Whether the house has any elevator
fiveYearsProperty	int64	Whether the house is a five-year property
subway	int64	Whether the house is near any subways

Hình 4.8: Danh sách các đặc trưng sử dụng trong mô hình

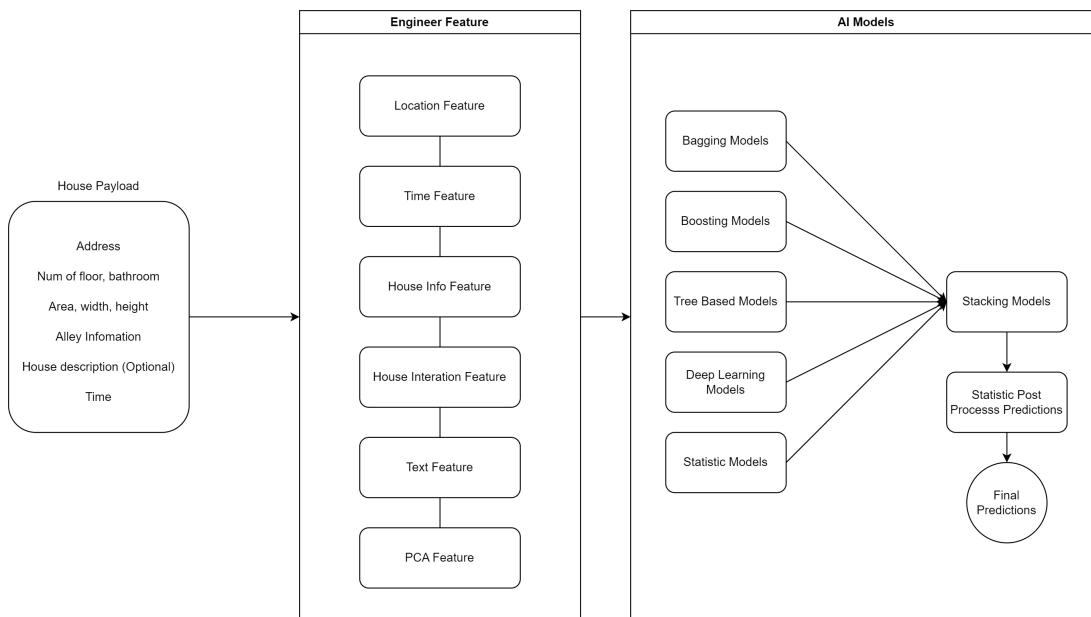
Tuy nhiên giải pháp này vẫn có những nhược điểm chưa giải quyết được đó là: Mô hình vẫn chưa có thêm thông tin tri thức về mối liên hệ các bất động sản trong vùng, mô hình vẫn chưa phân tích các thông tin về cơ sở vật chất trường học xung quanh bất động sản ảnh hưởng đến ngôi nhà như thế nào.

- Ở nghiên cứu [18], tác giả đã sử dụng mô hình XGBRegressor để dự đoán giá nhà với 14 thuộc tính. Nhược điểm của giải pháp này cũng nằm ở tính tổng quát của mô hình dự đoán khi chỉ dùng một mô hình. Điều này làm cho kết quả dự đoán ở các phân vùng dữ liệu có độ biến động cao. Hơn nữa giải pháp vẫn thiếu đi một số thông tin như thông tin quan như thông tin tương tác giữa các bất động sản lân cận, thiếu các thông tin tiện ích công ảnh hưởng đến giá trị bất động sản.
- Ở nghiên cứu [19], tác giả sử dụng Stacking Model Regression, điểm này đã phần nào giải quyết được bài toán độ biến động cao của mô hình dự đoán, tuy

nhiên giải pháp vẫn thiếu đi những thông tin quan trọng ảnh hưởng đến giá trị bất động sản.

- Ở nghiên cứu [20], tác giả sử dụng mô hình LSTM phân tích được tầm ảnh hưởng của thời gian lên giá trị bất động sản. Tuy nhiên như đối với dữ liệu dạng bảng thì các mô hình truyền thống vẫn là lựa chọn tốt hơn và mang tính tổng quát hơn. Nhất là các mô hình ensemble từ các mô hình truyền thống. Bên cạnh đó, giải pháp vẫn chưa giải quyết được những vấn đề tồn đọng ở phía trên.
- Ở nghiên cứu [21], tác giả có sự phân tích tương quanh giữa các thuộc tính. Tuy nhiên thiếu đi sự tổng quát và những thông tin ảnh hưởng đến giá vẫn chưa được giải quyết triệt để ở nghiên cứu này.
- Ở giải pháp [22], tác giả sử dụng 12 thuộc tính để dự đoán giá bất động sản. Điểm mạnh của giải pháp chính là có sự so sánh kết quả giữa các mô hình với nhau và có thống kê bảng biểu. Tuy nhiên giải pháp vẫn thiết đi những thông tin cần thiết để dự đoán giá nhà.

b, Đề xuất giải thuật dự đoán giá



Hình 4.9: BKPrice Prediction Algorithm

Hình vẽ 4.9 mô tả kiến trúc của giải thuật dự đoán giá nhà trong BKPrice System.

Ở mục trước thì chúng ta đã phân tích được các ưu điểm và nhược điểm của các giải pháp trên. Do đó trong nghiên cứu này tôi sẽ đề xuất một cách tiếp cận mới

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

Name	Description
University - School	Trường học
Fuel	Cây xăng
Cafe	Quán cafe
Parking	Bãi đỗ xe
Fast food	Tiệm đồ ăn nhanh
Marketplace	Chợ
Restaurant	Nhà hàng
Hospital	Bệnh viện
Kindergarten	Nhà mẫu giáo
Community centre	Trung tâm công cộng
Police	Khu vực cảnh sát
Place of worship	Nhà thờ
Bank - ATM	Ngân hàng

Bảng 4.1: Danh sách tiện tích công

cho bài toán dự đoán giá vừa kề thừa được những điểm mạnh, vừa giải quyết được những điểm yếu của các lời giải trên.

Đầu tiên, tôi muốn khai phá triệt để những thông tin liên quan đến vị trí của bất động sản. Ở đây module location feature (facility feature) sẽ đảm nhiệm vai trò này.

Các thông tin liên quan đến các tiện ích công xung quanh bất động sản: nhà hàng, trường học, ... ảnh hưởng trực tiếp đến giá bất động sản. Điều này là hiển nhiên sở dĩ càng có nhiều tiện ích xung quanh thì ngôi nhà càng được ưa chuộng hơn. Do đó mà giá trị của bất động sản sẽ có xu hướng cao hơn.

Ở trong nghiên cứu này, tôi sử dụng OpenStreetMap để trích xuất thông tin về tiện ích công xung quanh bất động sản dựa vào latitude và longitude. Củ thể các thuộc tính được sử dụng trong giải thuật dự đoán giá đó là: số lượng tiện ích công xung quanh bán kính 500m, 1000m và 2000m.

Bảng 4.1 mô tả danh sách các tiện ích công được sử dụng trong nghiên cứu.

Trong quá trình nghiên cứu tôi thấy rằng việc bất động sản gần những cung đường nổi tiếng hay trung tâm thương, ... ảnh hưởng đến giá bất động sản. Hình vẽ ... hiện thi danh sách các địa điểm nổi tiếng ở Hà Nội và Hồ Chí Minh. Từ đó tạo thêm những thuộc tính mới dựa vào khoảng cách tới các địa điểm này. Do ở mức độ thử nghiệm nên việc thu thập danh sách địa điểm được thực hiện bằng tay. Để có thể có nhiều thuộc tính hơn và tin cậy hơn, tương lai hệ thống sẽ thu thập tự động danh sách các địa điểm này.

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

lat	lon	name	address
10.773358	106.701287	Vincom Đồng Khởi	72 Lê Thánh Tôn, Bến Nghé, Quận 1, Hồ Chí Minh
21.030585	105.802927	Công viên Thủ Lệ	Công viên Thủ Lệ, Đường Bưởi, Ngọc Khánh, Ba Đình, Hà Nội
10.795529	106.716628	Công viên Vinhomes Central Park	208 Nguyễn Hữu Cánh, Phường 22, Quận Bình Thạnh, Hồ Chí Minh
10.809043	106.671617	Công viên Gia Định	Đường Hoàng Minh Giám, Phường 9, Phú Nhuận, Hồ Chí Minh
21.051217	105.835552	Hồ Tây 5	Hồ Tây, Tây Hồ, Hà Nội
21.011068	105.846908	Vincom Bà Triệu	191 Phố Bà Triệu, phường Lê Đại Hành, quận Hai Bà Trưng, Hà Nội
21.010941	105.839811	Công viên thống nhất	Công viên Thống Nhất, Đường Lê Duẩn, Lê Đại Hành, Đống Đa, Hà Nội
10.727873	106.717026	SC VivoCity Shopping Center	1058 Nguyễn Văn Linh, KP. 1, P. Tân Phong, Quận 7, TP. HCM
10.772185	106.698227	Công viên trên mây tại Taka Shimaya	92-94 Nam Kỳ Khởi Nghĩa, Phường Bến Nghé, Quận 1, Hồ Chí Minh
21.015025	105.775093	Tops Market The Garden	Tops Market The Garden, Mễ Trì, Từ Liêm, Hà Nội
20.964183	105.852110	Công viên yên sở	Công viên Yên Sở, Quốc lộ 1A, Gamuda Central, Hoàng Mai, Hà Nội
10.858315	106.582624	Công viên cá Koi Rin Rin Park	87/8P Xuân Thới Thượng 6, Ấp Xuân Thới Đông, Huyện Hóc Môn, Hồ Chí Minh
21.003251	105.801975	Trung tâm thương mại Hà Nội Center Point	27 Đường Lê Văn Lương, Nhân Chính, Thanh Xuân, Hà Nội
10.788040	106.700999	Thảo Cầm Viên	2 Nguyễn Bình Khiêm, Phường Bến Nghé, Quận 1, Hồ Chí Minh
21.029358	105.852400	Hồ Gươm	Hồ Gươm, Phố Lê Thái Tổ, Hàng Trống, Hoàn Kiếm, Hà Nội
21.016627	105.782185	Trung tâm thương mại Hà Nội Keangnam	72 Phạm Hùng, Keangnam, Mễ Trì, Từ Liêm, Hà Nội
21.002143	105.812609	Trung tâm thương mại Royal City Hà Nội	72A Đường Nguyễn Trãi, Khu đô thị Royal City, phường Thượng Đình, quận Đống Đa, Hà Nội
21.027274	105.896800	Aeon Mall Long Biên	Aeon Mall Long Biên, Đường Cổ Linh, p. Long Biên, Long Biên, Hà Nội

Hình 4.10: Danh sách các địa điểm nổi tiếng

Trong nghiên cứu này tôi cũng đã phân tích được nhận định trên. Khi xây dựng mô hình dự đoán giá bất động sản và đánh giá độ quan trọng của các thuộc tính.

Tiếp theo, trong nghiên cứu này tôi phân tích mức độ ảnh hưởng các bất động sản lân cận lên giá của bất động sản.

Một trong những cách tốt nhất để phân tích các thuộc tính liên quan đến các bất động sản lân cận đó chính là xác định top k bất động sản gần nhất dựa vào cặp giá trị (latitude, longitude). Khoảng cách giữa 2 cặp (latitude1, longitude1) và (latitude2, longitude2) đó chính là khoảng cách đường chim bay. Để tối ưu hóa trong việc tìm kiếm các bất động sản lân cận thì tôi sử dụng quadtree (Cấu trúc dữ liệu đã được giới thiệu ở chương trước).

Bài toán: Có n bất động sản trong bộ dataset được đại diện bởi tập n cặp điểm: $(lat_1, lon_1), (lat_2, lon_2), \dots (lat_n, lon_n)$.

Mục tiêu: Tìm tập k điểm gần nhất của (lat, lon)

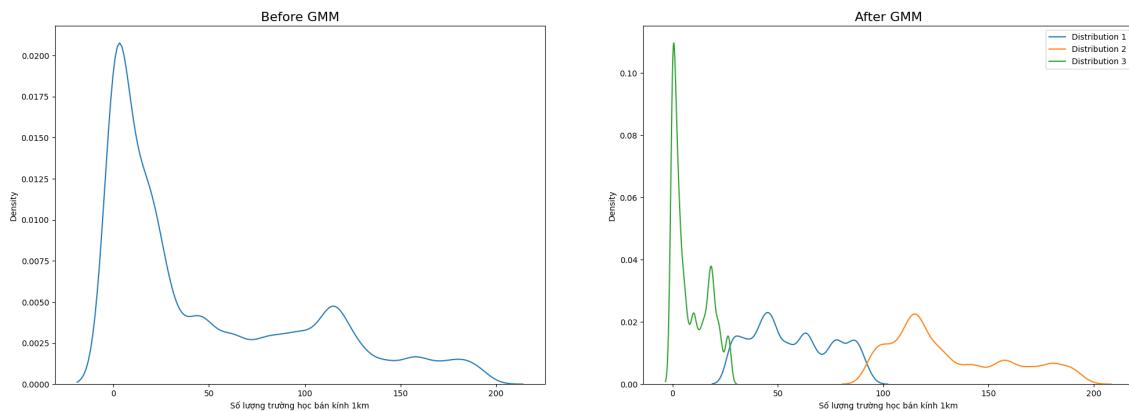
- Xây dựng quadtree với độ phức tạp $O(n\log n)$
- Tìm kiếm k điểm gần nhất với độ phức tạp $k * O(\log n)$

Gọi (lat_i, lon_i) là điểm thứ i trong k điểm gần nhất. Từ đó ta có danh sách các thuộc tính mới: $(district_i, ward_i, street_i, distance_i)$ là thông tin đường, phường, quận, khoảng cách chim bay của điểm thứ i ($1 \leq i \leq k$)

Bên cạnh đó các thuộc tính sau cũng sẽ được thêm trong quá trình huấn luyện mô hình:

- Thông tin quận: Dân số quận, diện tích quận, mật độ dân số, số lượng phường,

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT



Hình 4.11: Mô hình hóa mật độ của thuộc tính *num_of_school_in_1000m_radius* trước và sau khi khớp Gaussian Mixture Model

trung tâm của quận dựa trên lat, lon.

- Khoảng cách chim bay của bất động sản đến trung tâm của quận.
- Tỷ lệ diện tích của bất động sản với tổng diện tích của quận

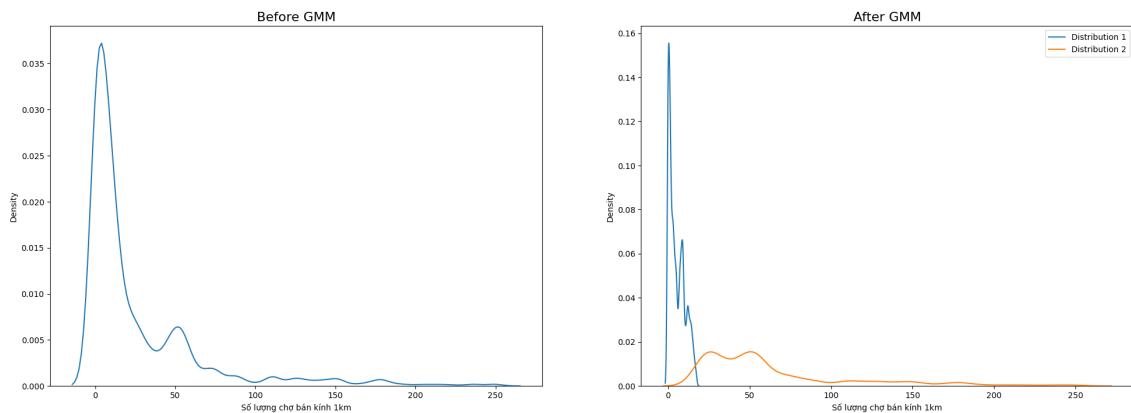
Sau khi thực hiện tạo thêm nhiều thuộc tính để training mô hình thì tôi đã thực hiện bước khai phá dữ liệu và từ đó có một vài insight sau:

- Một vài thuộc tính đang tuân theo phân bố gaussian mixture (Tổ hợp các phân bố gaussian). Do đó, việc kiểm định thuộc tính của những mẫu thử mới thuộc vùng phân bố nào của tập training dataset đóng vai trò quan trọng trong việc định giá chính xác hơn. Tôi đặt ra giả thuyết rằng một số thuộc tuân theo phân phối gaussian mixture. Điểm khác biệt của mô hình hỗn hợp Gaussian so với mô hình đơn Gaussian đó chính là mỗi điểm dữ liệu thay vì được phân loại vào 1 lớp (1 phân bố) thì được chia ra làm n phân bố khác nhau. Số lượng phân bố Gaussian phản ánh mức độ tổng quát của mô hình và ảnh hưởng tới quá trình học của mô hình trên những thuộc tính ấy.

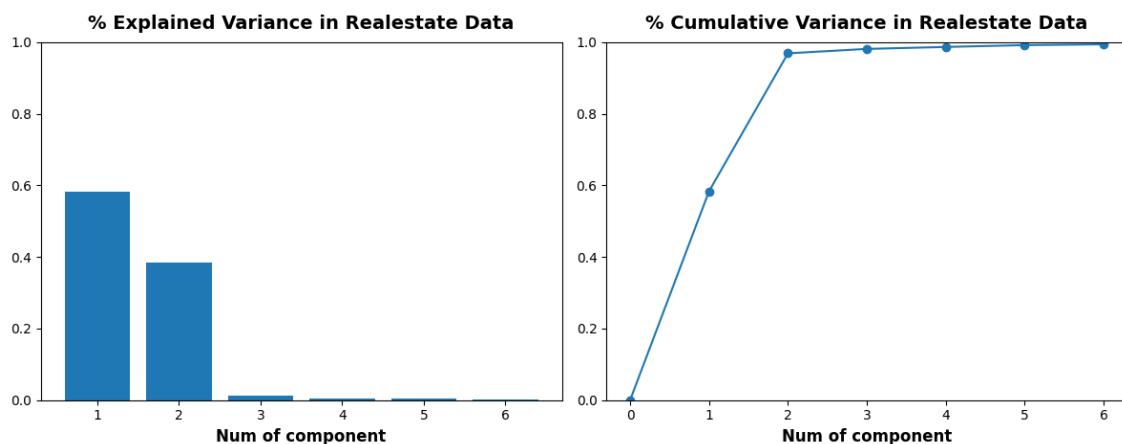
Hình vẽ 4.11 cho thấy thuộc tính số lượng trường học bán kính 1km đang mô hình tốt với tổ hợp 3 mô hình gaussian. Tương tự thuộc tính số lượng chợ bánh kính 1km cũng mô hình tốt với tổ hợp 2 mô hình gaussian. Tương tự hình vẽ 4.12 thuộc tính đang mô hình tốt với tổ hợp 2 mô hình gaussian.

Do đó ta thêm thuộc tính mới sau:

- *gmm_2_component_x*: Categorical Feature - [0, 1] (0 nghĩa là điểm dữ liệu có thuộc tính x thuộc phân bố thứ 1 trong 2 phân bố gaussian)
- *gmm_3_component_x*: Categorical Feature - [0, 1, 2] (2 có nghĩa là điểm dữ liệu có thuộc tính x thuộc phân bố thứ 3 trong 3 phân bố gaussian)
- Để làm nổi bật mối tương quan giữa các thuộc tính và giảm nhiễu giữa thì tôi



Hình 4.12: Mô hình hóa mật độ của thuộc tính *num_of_marketplace_in_1000m_radius* trước và sau khi khớp Gaussian Mixture Model



Hình 4.13: Biểu đồ đơn và tích lũy phương sai của các chiều dữ liệu

đã sử dụng kỹ thuật PCA đã được giới thiệu ở chương trước để giảm chiều dữ liệu và tạo ra thuộc tính giúp mô hình học tốt hơn. Tuy nhiên một trong những vấn đề lớn khi sử dụng PCA đó chính là giảm đến bao nhiêu chiều là đủ. Tôi thử nghiệm với số lượng chiều là 6. Hình 4.13 mô tả biểu đồ đơn và tích lũy phương sai cho 6 chiều dữ liệu. Ta nhận thấy rằng 2 chiều dữ liệu 1 và 2 có tỷ lệ phương sai cao nhất (58% và 39%). Điều này có nghĩa là có 97% thông tin từ tập training set được bảo toàn ở 2 chiều dữ liệu đầu tiên. Do đó tôi chọn số lượng chiều là 2 để tạo thêm 2 thuộc tính mới là *PCA1* và *PCA2* tương ứng cho giá trị của 2 chiều dữ liệu

Ở chương tiếp theo ta sẽ đến với thực nghiệm khi thêm các thuộc tính này trong quá trình huấn luyện mô hình.

Bên cạnh việc xây dựng thuộc tính, thì lựa chọn mô hình cũng đóng vai trò quan trọng trong việc định giá. Để cân bằng giữa độ hiểu quả thời gian xây dựng và tính giải thích được của mô hình thì trong nghiên cứu này tôi sử dụng tập *Tree-based models* cùng với tập *Boosting models*, mô hình *Multilayer perceptron* đơn giản,

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

Model Name	Model Type	Training Device
CatboostRegressor	Boosting Model	GPU
MLPRegressor	Multilayer perceptron Model	GPU
LGBMRegressor	Boosting Model	GPU
XGBRegressor	Boosting Model	GPU
AdaBoostRegressor	Boosting Model	CPU
GradientBoostingRegressor	Boosting Model	CPU
ExtraTreesRegressor	Tree-based Model	CPU
RandomForestRegressor	Tree-based Model	CPU
Lasso	Regression Model	CPU
Ridge	Regression Model	CPU
Linear	Regression Model	CPU
KNeighborsRegressor	Neighbor-based Model	CPU

Bảng 4.2: Danh sách mô hình cơ sở trong BKPrice Prediction System

các mô hình dựa trên láng giềng (Neighbor-based models) và các mô hình hồi quy (Regression models)

Hệ thống định giá xây dựng một mô hình *stacking/ensemble* có tên là *BKPrice Model*. Mô hình này đóng vai trò tổng hợp thông tin từ những mô hình trên, hậu xử lý kết quả dự đoán bằng phương pháp thống kê để đảm bảo tính tin cậy của hệ thống. Bảng 4.2 mô tả danh sách các mô hình cơ sở được sử dụng trong hệ thống.

Như chúng ta được biết thì quá trình lựa chọn tham số huấn luyện mô hình là một điều quan trọng và được ưu tiên hàng đầu. Do đó bên cạnh việc huấn luyện mô hình tự động thì việc lựa chọn tham số cũng được diễn ra tự động.

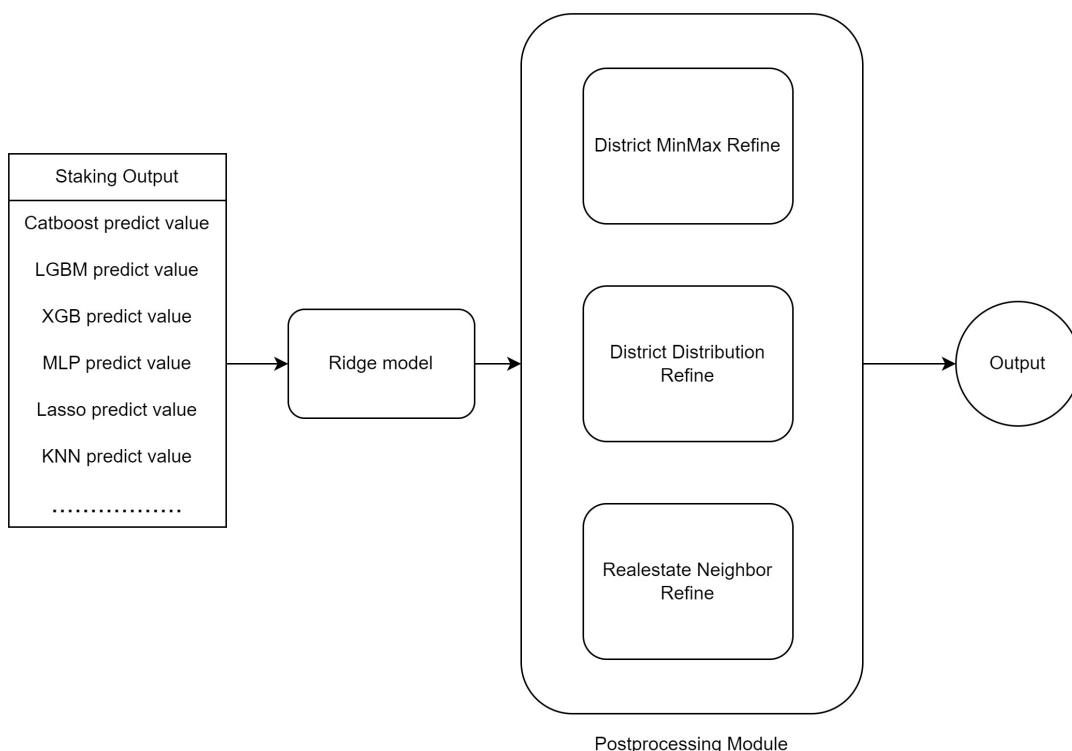
Để lựa chọn tham số tối ưu cho mô hình học máy thì tôi sử dụng class *GridSearchCV* của thư viện *scikit-learn*. Việc sử dụng GridSearchCV giải quyết các vấn đề sau của BKPrice System:

- Hiệu quả trong việc tìm kiếm tham số tối ưu
- Có khả năng mở rộng và đáp ứng được nhiều mô hình học máy
- Tự động hóa quá trình tìm kiếm tham số tối ưu.

Trong nghiên cứu này tôi lựa chọn các tham số: *n_estimators / iterations* , *learning_rate*, *max_depth*

Hình vẽ 4.14 mô tả luồng hậu xử lý kết quả dự đoán. Tôi gọi là *BKPrice Model*. Mô hình này đóng vai trò tích hợp thông tin dự đoán của các mô hình cơ sở. Để thực hiện nhiệm vụ trên thì tôi xây dựng một tập training dataset *D* như sau:

- Các thuộc tính của tập D chính là giá trị dự đoán của các mô hình cơ sở *B*



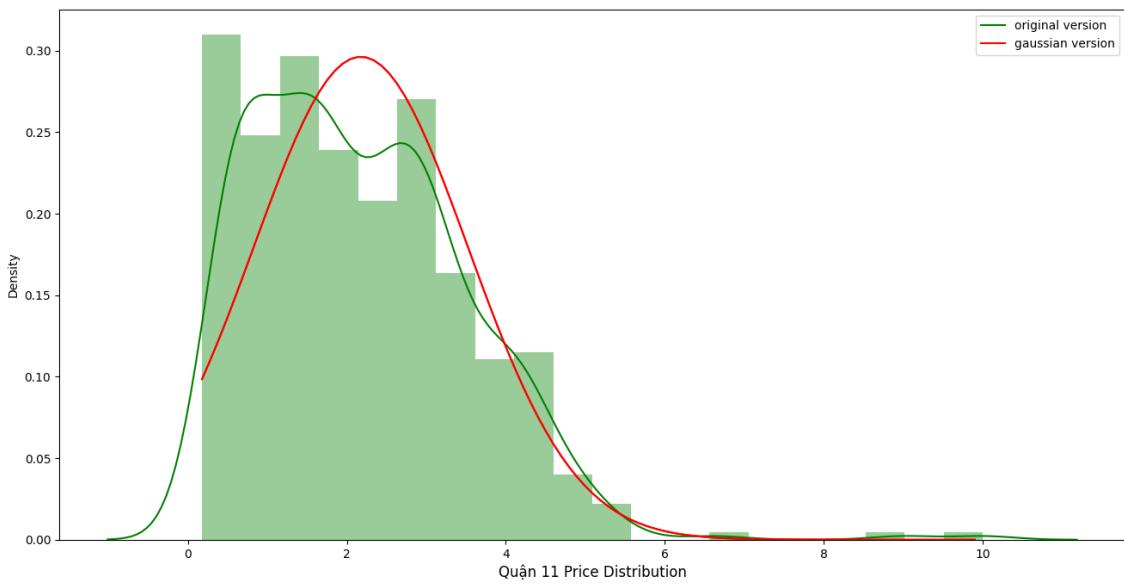
Hình 4.14: BKPrice Model

- Xây dựng các thuộc tính thống kê liên quan đến giá trị dự đoán của mô hình cơ sở B : mean - giá trị trung bình, std - giá trị độ lệch chuẩn

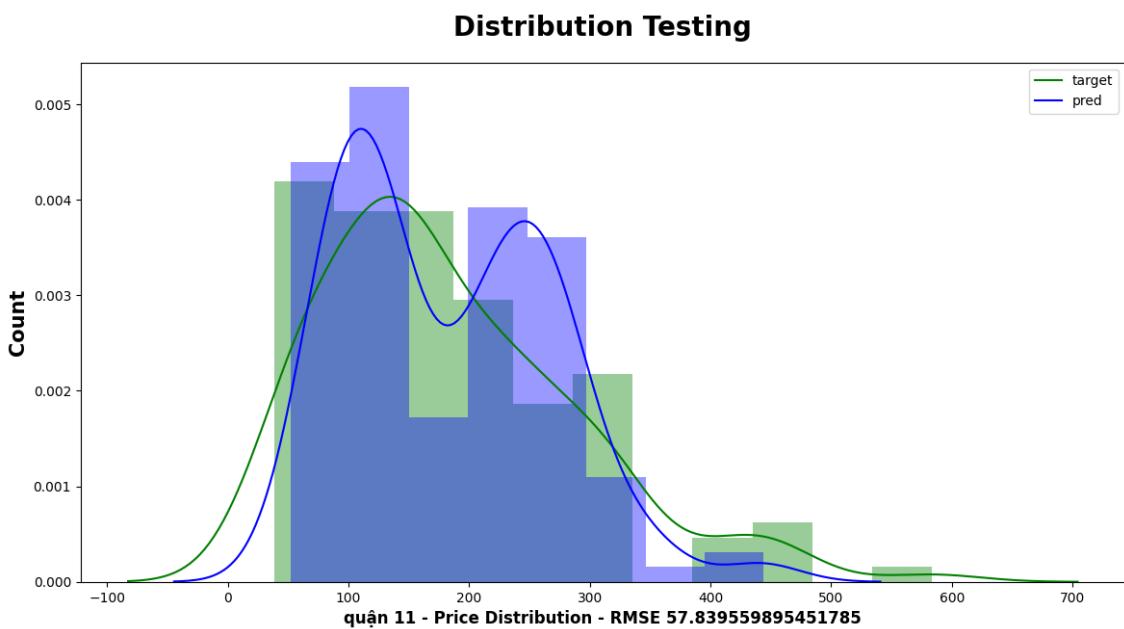
Trong nghiên cứu này tôi lựa chọn mô hình hồi quy Ridge để huấn luyện và đưa ra kết quả dự đoán. Để đảm bảo mức độ tin cậy trong kết quả dự đoán thì BKPrice Model hậu xử lý kết quả như sau:

- District MinMax Refine: đảm bảo kết quả dự đoán trong quận không vượt ra ngoài khoảng giá trị bé nhất và giá trị lớn nhất mà dữ liệu ghi nhận được
- District Distribution Refine: đảm bảo kết quả dự đoán không nằm ngoài phân bố giá của quận. Ở đây tôi tiếp tục dùng phân bố Gaussian.

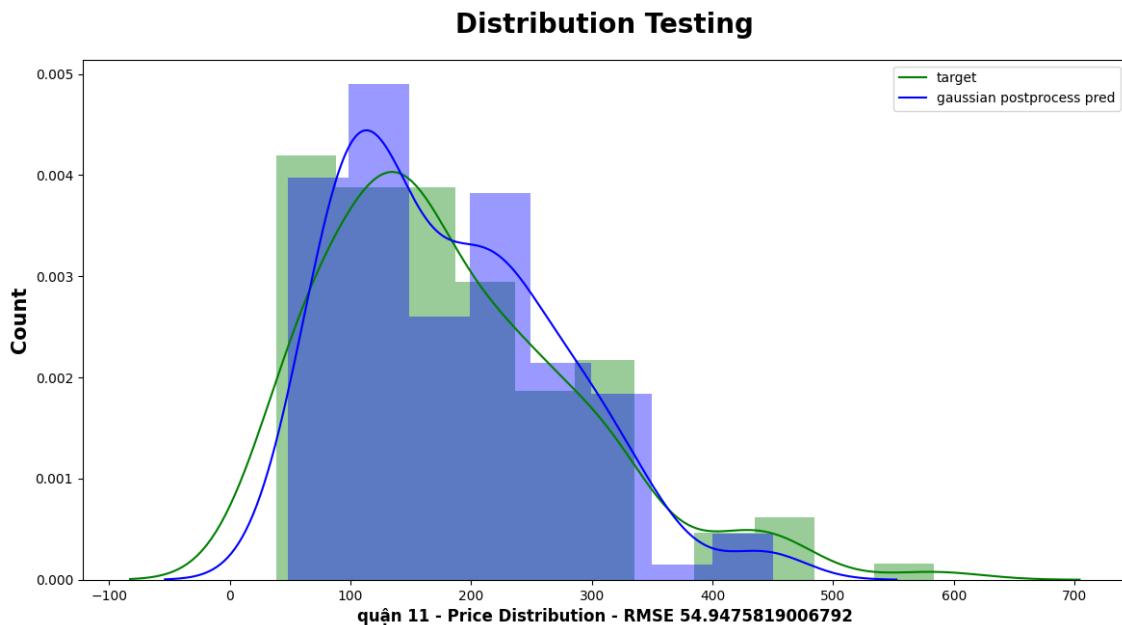
Hình vẽ 4.15 cho thấy giá của khu vực Quận 11, TP.HCM tuân theo phân bố Gaussian. Do đó ý tưởng mà tôi muốn đề cập là giá trị dự đoán được mô hình đưa ra phải khớp theo phân bố Gaussian này. Ở hình vẽ 4.16 ta thấy rằng một vài điểm phân bố giá dự đoán của mô hình chưa thực sự khớp với thực tế. Ở trong nghiên cứu này, đối với mỗi quận sẽ ứng với một hàm phân bố xác suất khác nhau GMM (Ở chương trước tôi đã đề cập cách tìm ra hàm phân bố xác suất bằng cách tối ưu hóa từng bước tính kỳ vọng Expectation và bước tính tối đa hóa Maximization). Từ đó với những giá trị dự đoán thì hệ thống sẽ khớp giá trị này với phân bố giá của từng quận. Hình vẽ 4.17 mô tả phân bố giá thực



Hình 4.15: Giá khu vực quận 11 tuân theo phân bố gaussian



Hình 4.16: Phân bố giá dự đoán chưa khớp với phân bố giá thực tế



Hình 4.17: Phân bố giá dự đoán đã gần khớp với phân bố giá thực tế

tế và giá dự đoán được hậu xử lý bằng Gaussian đã khớp nhau hơn.

- Realestate Neighbor Refine: đảm bảo kết quả dự đoán không vượt ra ngoài khoảng giá trị lớn nhất và bé nhất đối với top k bất động sản hàng xóm. Ở đây top k bất động sản hàng xóm nghĩa là top k bất động sản gần nhất và tương tự về khía cạnh loại bất động sản, ...

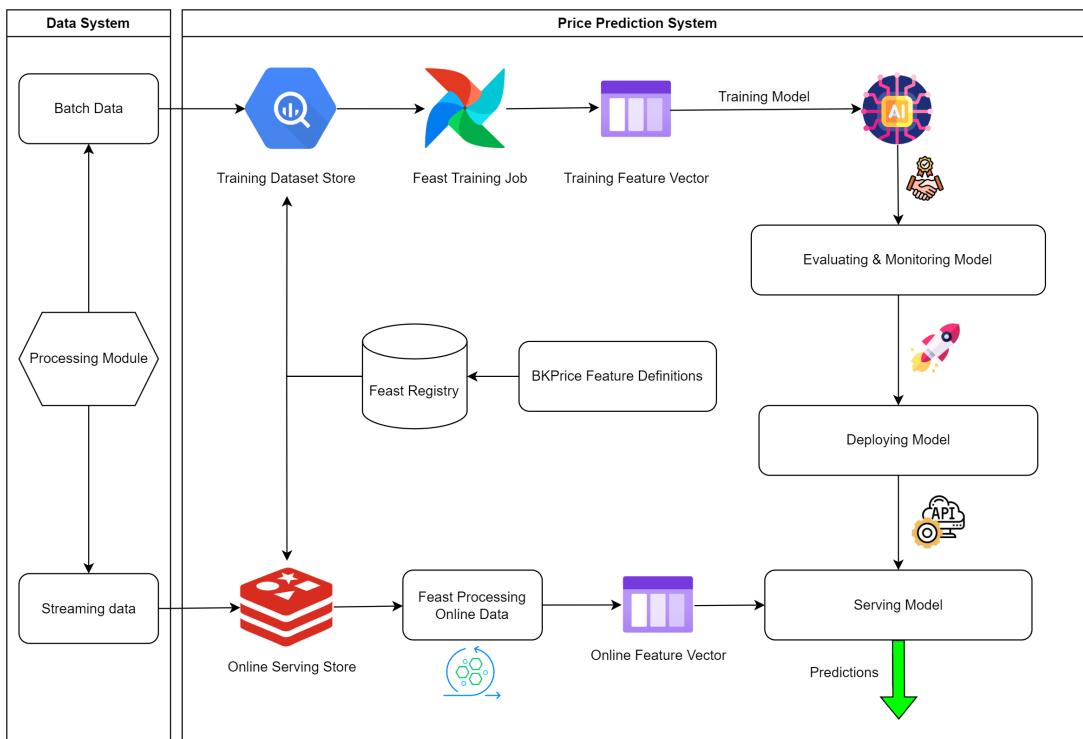
4.3.2 Hệ thống huấn luyện và triển khai mô hình tự động

Các mô hình trí tuệ nhân tạo được xây dựng và quyết định triển khai còn mang tính chủ quan, vô hình chung việc không kiểm soát độ hiểu quả và tin cậy của mô hình một cách tự động ảnh hưởng trực tiếp đến sản phẩm và người dùng đầu cuối. Do đó ở trong nghiên cứu này, tôi đề xuất một quy trình huấn luyện mô hình, đánh giá mô hình và quyết định lựa chọn mô hình cho quá trình inference một cách tự động, tin cậy và khách quan hơn.

Hình vẽ 4.18 mô tả quy trình huấn luyện mô hình một cách tự động trong hệ thống BKPrice.

Trong pha này, hệ thống có các thành phần quan trọng sau đây:

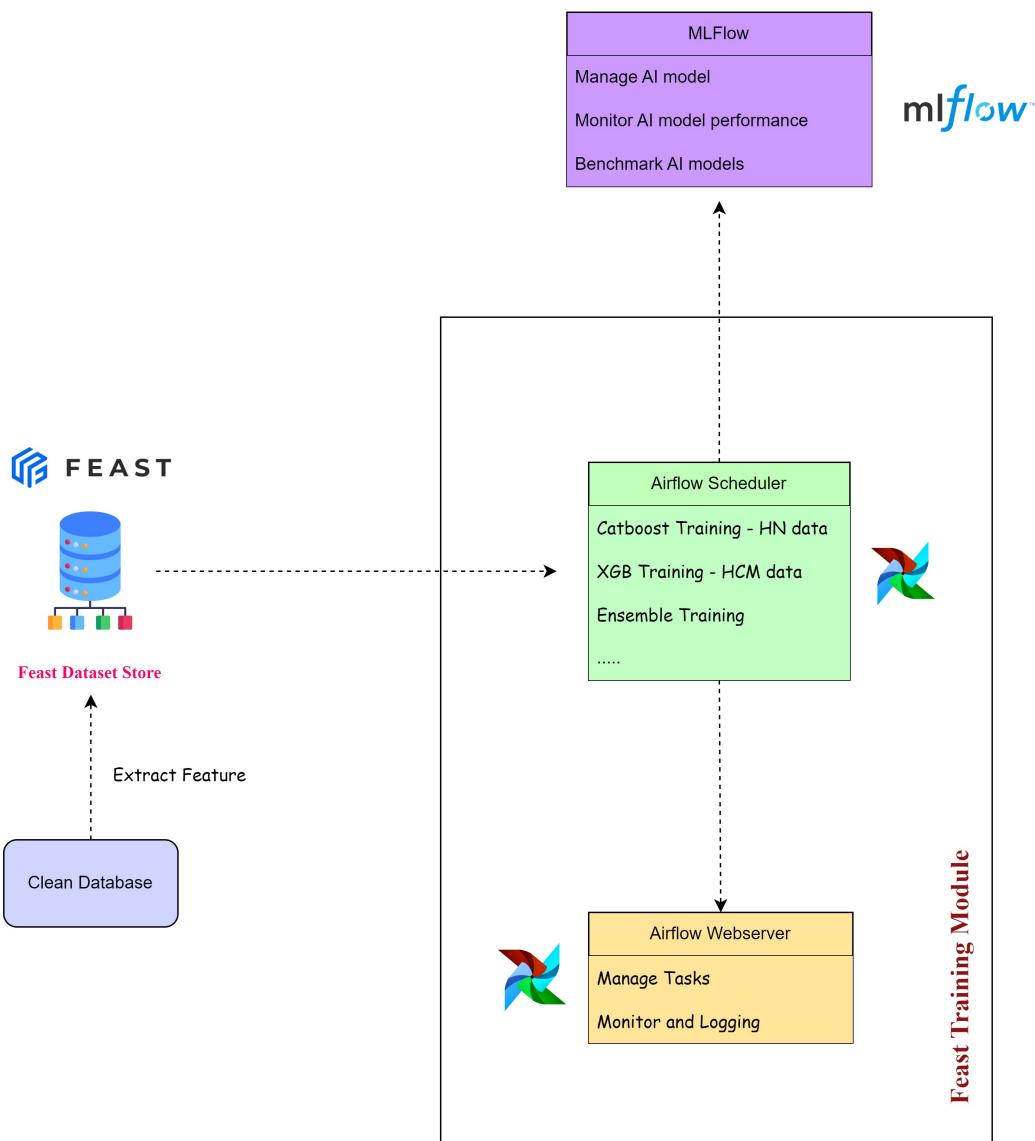
- Dữ liệu được thu thập tự động và được tiền xử lý bởi *Processing Module*. Dữ liệu sau bước tiền xử lý có tên gọi là *Batch Data*.
- *Training Dataset Store* là nơi lưu trữ danh sách data view để phục vụ cho quá trình huấn luyện mô hình tự động.
- Module *BKPrice Feature Definitions* đóng vai trò định nghĩa danh sách thuộc



Hình 4.18: BKPrice Prediction System

tính để xây dựng data view. Mỗi data view tương ứng với một tập dataset. *Feast registry* sẽ đóng vai trò đọc thông tin được định nghĩa trong file định nghĩa thuộc tính và tạo dataset trong *Traing Dataset Store*

- *Feast Training Job* đóng vai trò lên lịch để training mô hình. Trong nghiên cứu này tôi cấu hình với 1000 record mới được cập nhật vào *Batch Data* thì sẽ trigger một sự kiện huấn luyện mô hình dự đoán giá.
- *Training Feature Vector* là bộ vector đặc trưng được trích xuất theo những thuộc tính được định nghĩa trong *BKPrice Feature Definitions*
- Module tiếp theo đóng vai trò huấn luyện danh sách mô hình đơn và mô hình *stacking*. Quá trình huấn luyện *BKPrice Module* tích hợp từ những mô hình trên cũng được diễn ra một cách tự động.
- Module *Evaluating - Monitoring* đóng vai trò đánh giá và giám sát các mô hình sau khi được huấn luyện xong. Quá trình đánh giá mô hình và giám sát mô hình bằng MLFlow được diễn ra tự động. Các checkpoint của mô hình được lưu trữ đối với mỗi lần chạy. Do đó dễ dàng hơn trong việc lựa chọn phiên bản tốt nhất cho các pha sau của hệ thống.
- *Deploying Model* và *Serving Model* là pha cuối cùng của hệ thống đảm bảo quá trình triển khai và cung cấp dịch vụ AI dưới dạng API.



Hình 4.19: Tiếp nhận và huấn luyện mô hình tự động

Hình 4.18 cũng mô tả quá trình xử lý yêu cầu dự đoán giá bất động sản. *Streaming Data* là dữ liệu (đã được xử lý bằng Processing Module) theo batch được gửi yêu cầu định giá từ người dùng. Việc xây dựng *Feature Vector* tương tự như quá trình huấn luyện mô hình và chính là đầu vào cho module *Serving Model* để đưa ra kết quả định giá.

a, Tính tự động khi xây dựng dữ liệu huấn luyện

Hình vẽ 4.19 mô tả cách *Feast Training Module* tiếp nhận thông tin và huấn luyện mô hình tự động. *Feast Training Job* sẽ tiếp nhận thông tin từ *Feast Dataset Store*. Như đã đề cập ở phần trước, sau khi dữ liệu được thu thập, quá trình làm sạch sẽ được diễn ra tự động và lưu trữ vào *clean database*. Việc trích xuất thông tin: thông tin tiện ích công, thông tin khoảng cách, thông tin từ các bất động sản

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

lân cận, ... cũng sẽ được diễn ra tự động và cập nhật vào *Feast Dataset Store*. Đây là thành phần cho phép chúng ta quản lý tập huấn luyện mô hình và danh sách các đặc trưng tương ứng của dữ liệu.

The screenshot displays two main sections of the Feast Dataset Store interface:

Feature Views (Left):

Name	# of Features
df_hcm_feature_view_v3	188
df_hcm_feature_view_v0	16
df_hn_feature_view_v5	187
target_hn_feature_view_v2	1
df_hn_feature_view_v2	185

Data Sources (Right):

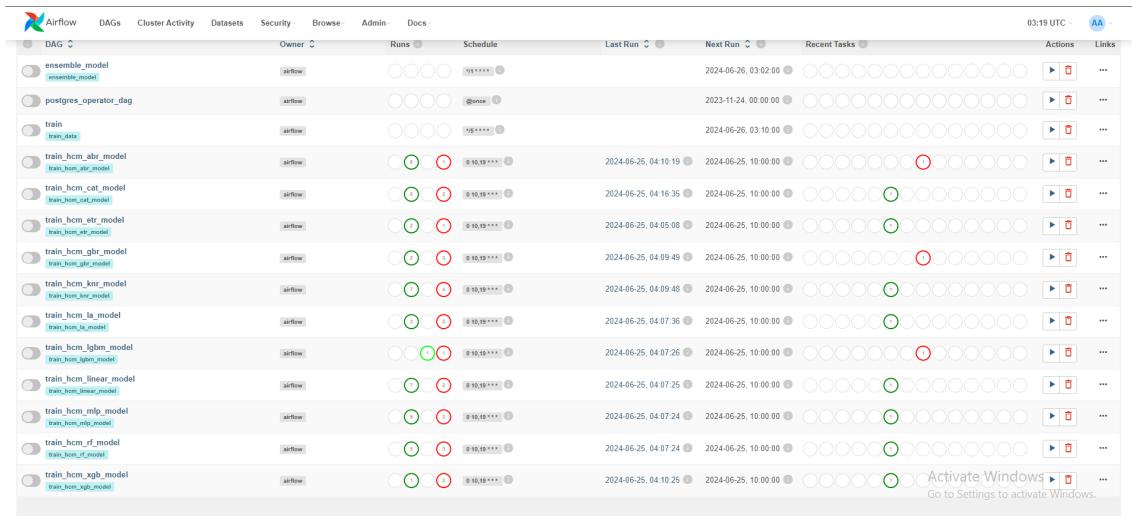
Name	Type
/home/long/long/datn-feast/data/target_df_hn_v1.parquet	BATCH_FILE
/mnt/long/long/datn-feast/data/update_data/demo1/target_df_hcm_v2.parquet	BATCH_FILE
/home/long/long/datn-feast/data/data_df_hn_v5.parquet	BATCH_FILE
/mnt/long/long/datn-feast/data/update_data/demo1/data_df_hn_v1.parquet	BATCH_FILE
/home/long/long/datn-feast/data/data_df_hn_v4.parquet	BATCH_FILE

Below these tables, there is a detailed list of feature columns and their types:

Column Name	Type	Column Name	Type
nearest_2_street	INT32	lon	FLOAT
houseDirection	INT32	frontWidth	FLOAT
nearest_1_ward	INT32	lat	FLOAT
nearest_6_district	INT32	w	FLOAT
nearest_3_district	INT32	numberOfBathRooms	FLOAT
nearest_4_ward	INT32	landSize	FLOAT
nearest_8_district	INT32	numberOfBedRooms	FLOAT
gmm_2_component_num_of_cafe_in_1000m_radius	INT32	num_of_marketplace_in_1000m_radius	FLOAT

Hình 4.20: Màn hình quản lý tập huấn luyện và các đặc trưng tương ứng

Hình vẽ 4.20 thể hiện thông tin về nguồn dữ liệu được lưu trữ ở đâu, có những *feature view* nào trong hệ thống và các thuộc tính của từng *feature view* là gì. Chúng ta quản lý tập danh sách huấn luyện mô hình và tập thuộc tính trở nên hiểu quả hơn. Điều này mang ý nghĩa quan trọng hơn trong việc quản lý được tính tự động trong khâu chuẩn bị dữ liệu training và giám sát được tập dữ liệu nào đang có ảnh hưởng tốt tới kết quả của bài toán dự đoán giá bất động sản ở Hà Nội và Hồ Chí Minh.



Hình 4.21: Giao diện chính quản lý các tác vụ huấn luyện mô hình

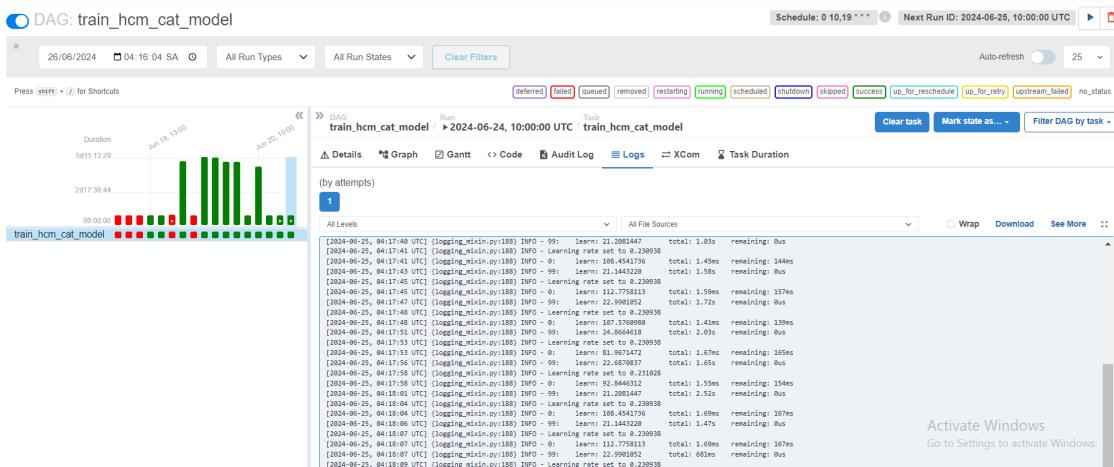
b, Tính tự động khi huấn luyện mô hình

Sau khi tiếp nhận thông tin từ Feast Dataset Store, hệ thống sẽ thực hiện quá trình huấn luyện mô hình với các tác vụ đã được định nghĩa và lên lịch sẵn. Feast Training Module đóng vai trò như bộ điều phối viên trong hệ thống. Tại đây sẽ thực thi luồng lên lịch chính của hệ thống, trong đó quan trọng nhất đó chính là (i) lên lịch để huấn luyện mô hình AI, (ii) thông báo thành công và lỗi các tác vụ huấn luyện, (iii) lưu trữ log trong mỗi lần chạy tác vụ.

Feast Training Module có 2 thành phần chính để đảm bảo những chức năng trên: (i) bộ lập lịch airflow scheduler, (ii) airflow webserver. Bộ lập lịch airflow scheduler thực hiện huấn luyện mô hình AI với dữ liệu đầu vào được lưu trữ ở Feast Store. Ở phía giao diện thì chúng ta có airflow UI và airflow webserver đảm nhận quản lý tác vụ và giám sát quá trình huấn luyện mô hình. Sự tách biệt rõ ràng của hệ thống có thể thấy ở việc mỗi thành phần chỉ thực hiện nhiệm vụ nhất định và gọi đến những tài nguyên nhất định. Việc các *DAG* huấn luyện mô hình tương tác trực tiếp với cơ sở dữ liệu đã được làm sạch, và trực tiếp trích xuất đặc trưng để thực hiện quá trình huấn luyện làm quá tải chức năng ban đầu của airflow scheduler đó chính là lập lịch. Vì vậy việc trích xuất đặc trưng sẽ được diễn ra tự động theo lô và lưu trữ ở *Feast Store*. Các tác vụ huấn luyện chỉ cần tương tác với *Feast Store* và thực hiện quá trình huấn luyện một cách độc lập.

Hình 4.21 mô tả giao diện chính quản lý các tác vụ huấn luyện mô hình. Chúng ta có thể dễ dàng thấy được có bao nhiêu loại tác vụ đã được lên lịch trong *scheduler* và mỗi tác vụ đang có trạng thái như thế nào: (i) Xanh đậm - thành công, (ii) Xanh lá cây - đang chạy, (iii) đỏ - thất bại. Đặc biệt, ở giai đoạn ổn định, số lượng tác vụ xanh và đỏ sẽ phản ánh được độ ổn định của mô hình.

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT



Hình 4.22: Giao diện quản lý các tác vụ trong mỗi DAG

Các tác vụ trong mỗi *DAG* cũng được quản lý trạng thái thực hiện đến bước nào, thành công hay thất bại, thời gian xử lý tác vụ. Điều này rất quan trọng trong quá trình tự động hóa của BKPrice System. Hình ảnh 4.22 mô tả giao diện chính của trang quản lý các tác vụ trong *DAG*. Bên cạnh đó phía giao diện cho phép người dùng: (i) Kết thúc các tác vụ chạy không tin cậy và kiểm tra dễ dàng hơn, (ii) cho phép khởi động một tác vụ, (iii) cho phép tiếp tục chạy các tác vụ đã được tạm dừng.

Qua các yếu tố trên chúng ta thấy được tính tự động trong hệ thống BKPrice, hơn thế nữa mọi quy trình tự động được giám sát rõ ràng ở phía giao diện và xử lý một cách kịp thời để tránh những sai sót không mong muốn.

c, Tính tự động khi giám sát và đánh giá mô hình AI

Tính tự động từ quá trình thu thập dữ liệu tới huấn luyện mô hình đôi lúc sẽ xảy ra những vấn đề không mong muốn như:

- Dữ liệu outlier hoặc nhiễu chưa được kiểm soát kỹ trước khi huấn luyện mô hình. Do đó kết quả mô hình có thể tệ đi bất thường.
- So sánh các phiên bản huấn luyện mô hình trở nên khó khăn nếu làm thủ công do số lượng lớn các tác vụ
- Chưa có cái nhìn tổng quan về các mô hình: biểu đồ, bảng biểu thay vì những con số khô khan.

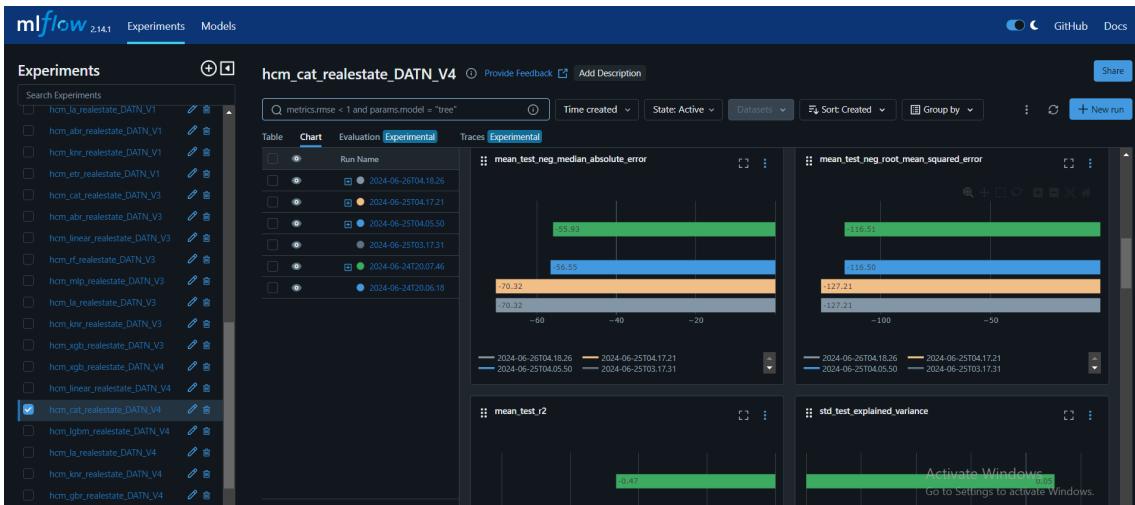
Để giải quyết những vấn đề trên thì tôi đã lựa chọn các độ đo và biểu đồ sau để giám sát các mô hình AI sau khi được huấn luyện:

- Biểu đồ độ tương quan giữa các biến thuộc tính và biến dự đoán trên bộ dữ liệu training.

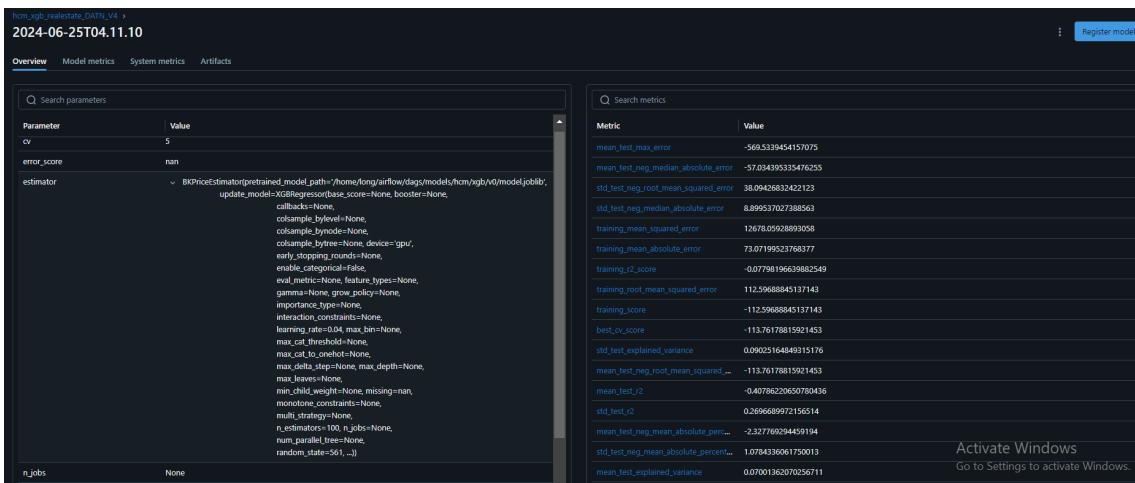
- Biểu đồ thể hiện độ quan trọng của từng thuộc tính đối với từng mô hình sau khi được huấn luyện
- *explained_variance*: Thể hiện tỷ lệ phương sai của dữ liệu được giải thích bởi mô hình sau khi huấn luyện. Nói một cách đơn giản hơn:
 - Giá trị *explained_variance* cao có nghĩa là mô hình có thể giải thích nhiều biến động trong dữ liệu dẫn đến dự đoán chính xác hơn
 - Giá trị *explained_variance* thấp có nghĩa là mô hình không thể giải thích tốt biến động trong dữ liệu dẫn đến kết quả dự đoán kém đi tính chính xác hơn
- *neg_mean_absolute_percentage_error*: Thể hiện tỷ lệ lỗi tuyệt đối của mô hình:
 - Giá trị *neg_mean_absolute_percentage_error* cao cho thấy mô hình dự đoán kém chính xác hơn
 - *neg_mean_absolute_percentage_error* thấp cho thấy mô hình dự đoán chính xác hơn
- *neg_root_mean_squared_error*: Thể hiện tỷ lệ lỗi gốc bình phương trung bình của mô hình:
 - Giá trị *neg_root_mean_squared_error* cao cho thấy mô hình dự đoán kém chính xác hơn
 - *neg_root_mean_squared_error* thấp cho thấy mô hình dự đoán chính xác hơn
- *max_error*: Thể hiện lỗi tuyệt đối lớn nhất mà mô hình mắc phải. Độ đo này giúp ta tìm được outlier của mô hình:
 - Giá trị *max_error* cao cho thấy mô hình có thể mắc sai sót lớn, dẫn đến dự đoán kém chính xác cho một số trường hợp trong dữ liệu.
 - *max_error* thấp cho thấy mô hình ít có khả năng mắc sai sót lớn, dẫn đến dự đoán chính xác hơn cho hầu hết các trường hợp trong dữ liệu.

Hình vẽ 4.23 mô tả các thời điểm huấn luyện mô hình. Mỗi thời điểm này sẽ tương ứng với một thời điểm lên lịch chạy của *airflow scheduler*. Ở trong MLflow mỗi tác vụ huấn luyện mô hình được gọi là một *experiment*. Mỗi experiment bao gồm nhiều thời điểm huấn luyện mô hình khác nhau. Điều này giúp hệ thống có thể giám sát được các thời điểm huấn luyện mô hình. Bên cạnh đó hệ thống còn giám sát được mô hình đang được huấn luyện với dữ liệu có kích thước và thuộc

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT



Hình 4.23: Màn hình quản lý thời các thời điểm huấn luyện mô hình



Hình 4.24: Thông tin của một thời điểm huấn luyện mô hình XGB trên bộ dữ liệu thành phố Hồ Chí Minh

tính như thế nào, trạng thái huấn luyện và thời gian huấn luyện xong mô hình. Điều này đảm bảo được tính giám sát được của quy trình MLOps tự động trong hệ thống BKPrice.

Bên cạnh đó việc trích xuất ra các thông số giám sát tính hiệu quả của mô hình cũng được trích xuất ra một cách tự động. Danh sách biểu đồ đánh giá khách quan được hiểu quả của các mô hình sau các lần chạy. Từ đó dễ dàng hơn trong việc đưa ra quyết định nên sử dụng mô hình nào.

Hình vẽ 4.24 thể hiện thông tin của một thời điểm huấn luyện mô hình XGB trên bộ dữ liệu thành phố Hồ Chí Minh. Chúng ta có thể giám sát được thử nghiệm (experiment) với mô hình nào, trạng thái thành công hay thất bại, các độ đo sau khi huấn luyện mô hình như thế nào.

Bài toán dự đoán giá là bài toán hồi quy vì thế hệ thống BKPrice đánh giá được

mức độ lỗi trung bình và mức độ lỗi lớn nhất của mô hình chiếm một vai trò quan trọng trong việc giám và đánh giá độ hiểu quả của dịch vụ AI. Bên cạnh đó hệ thống phân tích tự động được mức độ tương quan và độ quan trọng của từng thuộc tính. Từ đó điều chỉnh mô hình dễ dàng khách quan hơn.

d, Tính tự động trong việc đánh giá độ tin cậy của hệ thống

Quá trình tự động luôn mang đến nhiều rủi ro vì ít sự can thiệp bởi con người. Do đó để đánh giá được độ tin cậy của hệ thống thì tôi dùng các độ đo sau:

- Tỷ lệ phục hồi thành công dịch vụ xử lý dữ liệu bất động sản: Số lượng bản ghi được phục hồi thành công / số lượng bản ghi cần được phục hồi.
- Độ trễ của yêu cầu định giá bất động sản: Tính toán độ trễ của 1 yêu cầu định giá (Với ngữ cảnh hệ thống có đang có 1000 yêu cầu đồng thời)
- Tính tin cậy trong kết quả dự đoán của mô hình sau khi huấn luyện: Số lượng điểm dự đoán nằm trong phân phối / tổng số lượng điểm.

Ở mục thực nghiệm tôi sẽ đề cập cù thể hơn về tính tin cậy của hệ thống.

e, Tính tự động khi tích hợp mô hình AI

Ở phía trên chúng ta đã cùng tìm hiểu cách thành phần chính của hệ thống BKPrice Prediction System, vai trò và tương tác giữa các thành phần như thế nào.

Ở mục này, tôi muốn đề cập tới tính tự động khi tích hợp mô hình AI vào quá trình phát triển phần mềm. Cụ thể làm thế nào để đảm bảo quá trình cập nhật mô hình được diễn ra một cách tự động không ảnh hưởng tới phần mềm mà người dùng cuối sử dụng. Tính tự động của hệ thống đến từ:

- Dữ liệu huấn luyện mô hình được thu thập tự động
- Dữ liệu sau khi thu thập được xử lý tự động và xử lý lỗi tự động
- Xây dựng training dataset được diễn ra tự động
- Đưa ra quyết định lúc nào sẽ huấn luyện mô hình một cách tự động
- Việc tự động các bước trên được diễn ra ở phía background và trong suốt quá trình sử dụng của người dùng, do đó không ảnh hưởng tới trải nghiệm người dùng

Các phần trên đã giải thích được rõ tính tự động trong việc triển khai dịch vụ AI từ chuẩn bị dữ liệu, làm sạch dữ liệu và xây dựng dữ liệu đến công đoạn huấn luyện mô hình và đánh giá mô hình tự động. Do đó tính tự động khi triển khai dịch vụ AI được đảm bảo khi mà tính tự động của những dịch vụ nhỏ hơn cũng được đảm bảo.

4.3.3 Triển khai quá trình MLOPs bằng mô hình ngôn ngữ lớn

Như đã đề cập ở các mục trên, chúng ta đã có cách nhìn tổng quan hơn về các thành phần của hệ thống. Tuy nhiên với số lượng thành phần của hệ thống lớn như vậy, việc tương tác và khởi động hệ thống sẽ trở nên khó khăn. Do đó ở mục này tôi thực hiện quá trình tích hợp chatbot để quá trình chạy luồng MLOps trở nên dễ dàng hơn.

Ý tưởng chính của chatbot ở đây chính là hỗ trợ phía người quản trị và người dùng có cách nhìn tốt hơn về quy trình MLOps trong hệ thống BKPrice. Khởi động từng thành phần, chạy từng quy trình, tắt các dịch vụ trở nên khó khăn hơn trong các hệ thống lớn. Do đó việc tương tác với hệ thống bằng mô hình ngôn ngữ lớn là một điểm lợi thế để giúp người dùng và người quản trị có thể làm quen hơn với hệ thống và thực hiện từng quy trình dễ dàng hơn.

Mô hình ngôn ngữ tự nhiên đã trở thành một trong những chủ đề được mọi người quan tâm hơn từ khi ChatGPT ra đời. Sở dĩ vì khả năng thông hiểu và tạo ra các văn bản tự nhiên như con người. Do đó những mô hình ngôn ngữ tự nhiên được sử dụng với nhiều mục đích khác nhau như: dịch thuật, phân loại văn bản, trả lời tự câu hỏi. Những khả năng mà mô hình ngôn ngữ lớn làm được chứng tỏ rằng sự thông hiểu ngôn ngữ con người. Vì vậy mà ở trong nghiên cứu này, tôi đã tận dụng khả năng này của mô hình ngôn ngữ để tối giản hóa một vài bước trong quy trình tự động hóa MLOps.

Trong nghiên cứu này, tôi sử dụng kỹ thuật Prompt, tạo ra đầu vào đặc biệt và điều hướng mô hình ngôn ngữ thực hiện theo những yêu cầu cụ thể của người dùng. Cụ thể hơn có 3 loại prompt được sử dụng trong nghiên cứu này: (i) classifier prompt, (ii) controller prompt, (iii) information prompt.

Mô hình ngôn ngữ lớn hỗ trợ người dùng tương tác tốt hơn với quy trình MLOps của BKPrice System, bên cạnh đó bao gồm những câu hỏi xung quanh về MLOps. Do đó classifier prompt thực hiện tác vụ phân loại: tương tác với hệ thống MLOps hoặc hỏi những câu hỏi đáp xoay quanh lĩnh vực MLOps. Đối với những câu hỏi đáp về lĩnh vực MLOps, BKPrice System tận dụng khả năng và kiến thức của mô hình ngôn ngữ lớn để trả lời cho người tương tác. Đối với những truy vấn tương tác với hệ thống MLOps thì sẽ được giải quyết bởi controller prompt. Đây là một prompt làm nổi bật tính tự động hóa gọi chức năng (function calling) của mô hình ngôn ngữ lớn. Controller Prompt là một Prompt điều hướng mô hình ngôn ngữ lớn gọi theo những chức năng được định nghĩa sẵn. Đối với quy trình MLOps trong BKPrice System, mô hình ngôn ngữ lớn đảm bảo tương tác với những chức năng sau: (i) Thu thập dữ liệu, (ii) Làm sạch dữ liệu, (iii) Lưu trữ vào cơ sở dữ liệu, (iv)

Trích xuất đặc trưng và xây dựng tập huấn luyện mô hình dự đoán giá và (v) Huấn luyện mô hình dự đoán giá bất động sản

Dưới đây, mô tả mẫu của 3 loại prompt được sử dụng trong hệ thống BKPrice:

(i) Classifier Prompt: Bạn là người phân loại, bạn nhận được truy vấn sau từ người dùng, sử dụng thông tin chi tiết và chọn loại truy vấn nào sau đây mà yêu cầu của người dùng rơi vào được đưa ra ở định dạng json với định dạng key:value sau: (key là loại hành động, value là ý nghĩa của hành động): "MLOPS": Truy vấn lệnh thực hiện các chức năng hệ thống như: thu thập dữ liệu, dọn dẹp dữ liệu, chèn dữ liệu, xây dựng tập dữ liệu, đào tạo mô hình AI và giám sát mô hình AI. Truy vấn của người dùng không bao gồm các từ nghi vấn, Không phải là câu hỏi trong truy vấn. Phải là câu bắt buộc bao gồm các tác vụ yêu cầu liên quan đến xử lý dữ liệu, dọn dẹp dữ liệu, lưu trữ, xây dựng dữ liệu để thực hiện đào tạo mô hình AI liên quan đến bất động sản. "KHÁC": Các loại câu hỏi khác bao gồm các câu hỏi chung liên quan đến đường ống MLOps. Các câu hỏi như: bạn có thể làm gì, bạn hoạt động như thế nào.

(ii) Information Prompt: Bạn là một chatbot được chuyên hỏi đáp về quy trình MLOPS. Bạn có thể sử dụng những hiểu biết mà tôi cung cấp dưới đây để trả lời câu hỏi của người dùng. Nếu không có thông tin trong những gì tôi cung cấp, vui lòng trả lời theo hiểu biết của bạn. Và hãy nhớ rằng bạn trả lời các câu hỏi về quy trình MLOPS và các giải thuật trong định giá bất động sản.

(iii) Controller Prompt: Bạn là người điều khiển, bạn nhận được truy vấn từ người dùng, sử dụng thông tin mà tôi cung cấp và chọn hành động chính tốt nhất từ danh sách những hàm số mà tôi liệt kê dưới đây.

Bảng dưới đây mô tả danh sách chức năng tương tác giữa mô hình ngôn ngữ lớn và quy trình MLOps của hệ thống BKPrice.

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

Function Name	Parameter	Description
_crawl_data	+ realestate_source	Tiến hành thu thập dữ liệu
_clean_data		Tiến hành làm sạch dữ liệu từ hàng đợi
_insert_data		Tiến hành đẩy dữ liệu từ hàng đợi vào database
_build_offline_batch_data		Trích xuất đặc trưng, xây dựng tập huấn luyện
_train_price_prediction_model	+ modelname + feature_set_version + city	Huấn luyện mô hình dự đoán giá

Bảng 4.3: Function Calling List

CHƯƠNG 5. ĐÁNH GIÁ VÀ THỰC NGHIỆM

Ở chương trước đã trình bày về kiến trúc tổng quan của hệ thống, phân rã hệ thống lớn thành hệ thống con và trình bày giải thuật định giá, cách tối ưu giải thuật định giá bất động sản. Bên cạnh đó tôi đề cập đến vai trò của mỗi khối trong hệ thống. Trong phần này tôi sẽ tiến hành đánh giá giải thuật định giá bất động sản, bên cạnh đó thử nghiệm độ tự động của quy trình MLOps trong điều kiện thực tế và đưa ra nhận xét, cải tiến.

5.1 Các tham số đánh giá

Hệ thống BKPrice tận dụng quá trình tự động MLOPs kết hợp cùng giải thuật dự đoán giá bất động sản để đưa ra kết quả cho người dùng. Bên cạnh quá trình tự động MLOps, để đáp ứng quá trình chạy và giám sát dễ dàng hơn, BKPrice sử dụng ngôn ngữ tự nhiên để tương tác với người quản trị và người dùng.

Do đó để đánh giá độ hiểu quả của hệ thống, tôi đã chia hệ thống thành hai giai đoạn, mỗi giai đoạn tương ứng với một tập tham số đánh giá khác nhau: (i) Giai đoạn đánh giá hiệu năng mô hình dự đoán, (ii) giai đoạn đánh giá quy trình MLOps tự động.

5.1.1 Tập tham số đánh giá hiệu năng mô hình dự đoán

Bài toán dự đoán giá bất động sản là bài toán hồi quy. Do đó tôi sử dụng độ đo *Root mean square error (RMSE)* để đánh giá độ chính xác của giải thuật. Bên cạnh đó tôi sử dụng các thông số khác để đánh giá như: *độ lỗi lớn nhất (max_error)*, *mức độ giải thích của mô hình dự đoán (explained_variance)*,..

Bên cạnh việc đánh giá độ chính xác của mô hình, tôi thực hiện đánh giá hiệu năng của ở hai khía cạnh: (i) Thời gian và tỷ lệ thành công của quá trình huấn luyện mô hình tự động, (ii) Thời gian và tỷ lệ thành công của quá trình thực hiện dự đoán từ phía người dùng.

Thời gian (độ trễ) của quá trình huấn luyện tự động và dự đoán là khoảng thời gian khi hệ thống nhận được yêu cầu để khi nó phản hồi tới người dùng. Khi thời gian xử lý càng ngắn, người dùng sẽ nhận phản hồi nhanh chóng hơn. Do đó trải nghiệm của người dùng sẽ được cải thiện đáng kể. Đối với hệ thống BKPrice độ trễ chính là thời gian huấn luyện và thời gian dự đoán của mô hình định giá bất động sản.

Tỷ lệ thành công trong hệ thống BKPrice được đo bằng tỷ lệ số lần thành công trên tổng số lần thực nghiệm. Tham số đánh giá tỷ lệ thành công thể hiện mức độ ổn định và đáp ứng của một hệ thống tự động.

Cấu hình	Thông số
GPU	GA102GL [A40] - NVIDIA Corporation
CPU	32 Core/Socket - 2 Thread/Core - AuthenticAMD

Bảng 5.1: Cấu hình thiết bị sử dụng

5.1.2 Tập tham số đánh giá quy trình MLOps

Để đảm bảo quy trình MLOps tin cậy thì việc đánh giá và giám sát quy trình là không thể thiếu. Tôi thực hiện đánh giá quy trình MLOps trên những khía cạnh: (i) Thời gian và tỷ lệ thành công của từng giai đoạn giao tiếp thu thập dữ liệu, làm sạch dữ liệu, trích xuất đặc trưng đến xây dựng tập dữ liệu huấn luyện, huấn luyện và đánh giá, triển khai mô hình dự đoán. (ii) Tổng thời gian chạy và tỷ lệ thành công của một quy trình MLOPs tự động (iii) Trạng thái phần cứng của hệ thống trong quá trình thử nghiệm CPU, GPU, RAM.

5.2 Phương pháp thí nghiệm

Phần này sẽ trình bày mô tả về các thí nghiệm để đánh giá hệ thống BKPrice. Thiết lập môi trường thực nghiệm, thiết bị và cấu hình của các thành phần trong hệ thống, thiết lập siêu tham số đảm bảo quá trình thí nghiệm hoạt động ổn định.

5.2.1 Cấu hình thiết bị sử dụng

Bảng 5.1 mô hình cấu hình GPU và CPU. Mục đích sử dụng GPU để huấn luyện những mô hình cấu hình GPU. Bên cạnh đó tận dụng GPU để xây dựng chatbot phục vụ quá trình giám sát luồng MLOPs tốt hơn. Tận dụng nhiều luồng (thread) của CPU để tăng tốc độ xử lý nghiệp vụ trong hệ thống BKPrice: thu thập dữ liệu, làm sạch, huấn luyện mô hình.

5.2.2 Môi trường lập trình thử nghiệm

Hệ thống BKPrice được hoạt động với nhiều mô-đun khác nhau và tương tác qua lại với nhau từ việc xử lý thu thập dữ liệu, đẩy vào hàng đợi. Dịch vụ làm sạch dữ liệu lấy dữ liệu từ hàng đợi và xử lý lưu trữ vào cơ sở dữ liệu. Quá trình trích xuất đặc trưng và xây dựng tập huấn luyện, thực hiện huấn luyện mô hình một cách tự động. Do đó cấu hình các thông số thử nghiệm sẽ được mô tả rõ trong bảng 5.2 sau đây:

Environment	Framework - Version	Port config information
BKPriceBot	Uvicorn App - 0.24.0	Server port: 2001
Queue	Kafka - 2.0.2	Using 3 broker ports: 9092, 9093, 9094
Scheduler	Airflow - 2.9.2	UI port: 8080
Crawlbot	Uvicorn App - 0.24.0	Server port: 8885
Chatbot	Gradio Chat - 4.37.1	UI port: 7860
Feature Management	Feast App - 0.37.1	UI port: 8890, server port: 8008
AI Model Management	MLFlow - 2.14.1	UI port: 5000, server port: 5000
Monitor	Cronitor App - 4.7.1	Public service: cronitor.app
FeastBot	Uvicorn App - 0.24.0	Server port: 8886

Bảng 5.2: Cấu hình môi trường trong hệ thống BKPrice

5.2.3 Cấu hình siêu tham số

Bảng 5.3 mô tả cấu hình siêu tham số được sử dụng trong hệ thống BKPrice.

Param	GridSearchCV	Model
weight	[-0.001, 0, 0.001]	BKPriceEstimator (pretrained_model, update_model)
n_estimators	[500, 1000]	AdaBoostRegressor ExtraTreesRegressor GradientBoostingRegressor LGBMRegressor RandomForestRegressor XGBRegressor
learning_rate	[0.01, 0.04, 0.08]	AdaBoostRegressor MLPRegressor LGBMRegressor CatBoostRegressor ExtraTreesRegressor GradientBoostingRegressor LGBMRegressor RandomForestRegressor XGBRegressor
n_neighbors	[100, 500, 1000]	KNeighborsRegressor
max_iter	[100, 500, 1000]	Lasso Ridge

Bảng 5.3: Cấu hình siêu tham số của mô hình AI

Bảng 5.4 mô tả cấu hình tập thuộc tính sử dụng trong quá trình huấn luyện và dự đoán giá bất động sản.

5.2.4 Cấu hình tập thuộc tính

Feature Set Version	Information	Description
Version 0 - 15 Features	+ 7 Categorical Features + 8 Numerical Features	+ Basic Realestate Feature
Version 1 - 183 Features	+ 35 Categorical Features + 148 Numerical Features	+ Basic Realestate Feature + Facility Feature + Neighbor Feature
Version 2 - 188 Features	+ 40 Categorical Features + 148 Numerical Features	+ Basic Realestate Feature + Facility Feature + Neighbor Feature + Gaussian Feature
Version 4 - 63 Features	+ 7 Categorical Features + 56 Numerical Features	+ Basic Realestate Feature + Facility Feature
Version 5 - 190 Features	+ 40 Categorical Features + 150 Numerical Features	+ Basic Realestate Feature + Facility Feature + Neighbor Feature + Gaussian Feature + PCA Feature

Bảng 5.4: Cấu hình phiên bản thuộc tính

Đối với tập thuộc tính *version 0*, các mô hình trí tuệ nhân tạo được đào tạo với những thông tin bất động sản cơ bản được thu thập từ các nguồn: loại hình bất động sản, vị trí của bất động sản, thông tin về vị trí, diện tích, số phòng tắm, phòng ngủ.

Tập thuộc tính *version 1*, mô hình sẽ được đào tạo với nhiều thông tin hơn. Bên cạnh những thông tin cơ bản như *version 0*, mô hình sẽ được đào tạo với những thông tin về tiện ích công và các thuộc tính về bất động sản lân cận.

Tập thuộc tính *version 2*, phát triển hơn so với *version 1* với những thuộc tính liên quan về phân loại phân bố dữ liệu sử dụng phương pháp *GMM* đã được đề cập ở chương trước.

Tập thuộc tính *version 4*, mô hình sẽ chỉ huấn luyện với những thông tin cơ bản của bất động sản kết hợp với tiện tích công.

Tập thuộc tính *version 5* phát triển từ *version 2*. Mô hình sẽ được huấn luyện với những thuộc tính chống nhiễu được tọa từ phương pháp giảm chiều dữ liệu *PCA*.

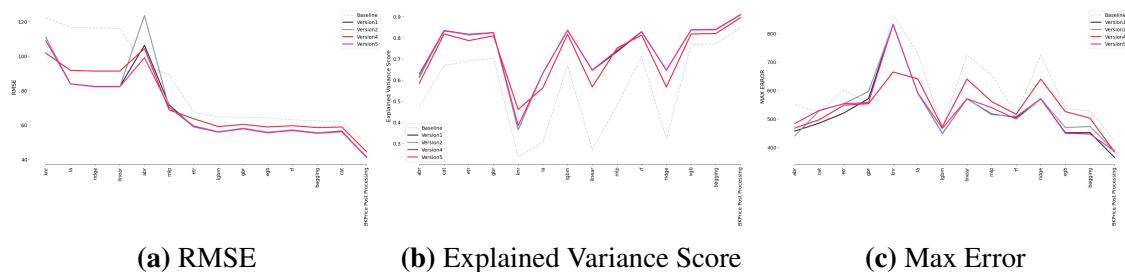
5.2.5 Tiến hành thí nghiệm

a, Mô hình dự đoán giá bất động sản

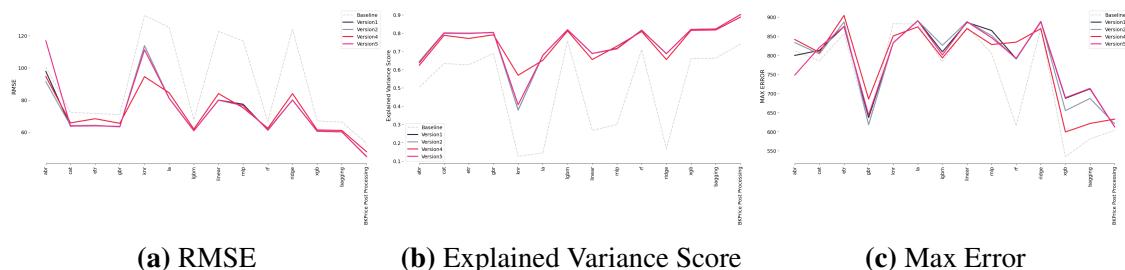
Ở mục này tôi sẽ đi tiến hành các thí nghiệm chất lượng mô hình dự đoán giá bất động sản dựa trên 5 bộ thuộc tính gồm: (i) 1 bộ thuộc tính cơ sở - version 0, (ii) 4 bộ thuộc tính còn lại được đề xuất.

Tập dữ liệu sử dụng trong quá trình đánh giá: (i) 35000 mẫu dữ liệu huấn luyện, 9000 mẫu dữ liệu đánh giá ở khu vực Hồ Chí Minh, (ii) 100000 mẫu dữ liệu huấn luyện, 9000 mẫu dữ liệu đánh giá ở khu vực Hà Nội.

Tập mô hình sử dụng trong quá trình đánh giá: (i) lớp mô hình học máy, học sâu cơ bản, các mô hình bagging cơ bản, (ii) mô hình học máy của BKPrice khi kết hợp các mô hình cơ bản và hậu xử lý mô hình.



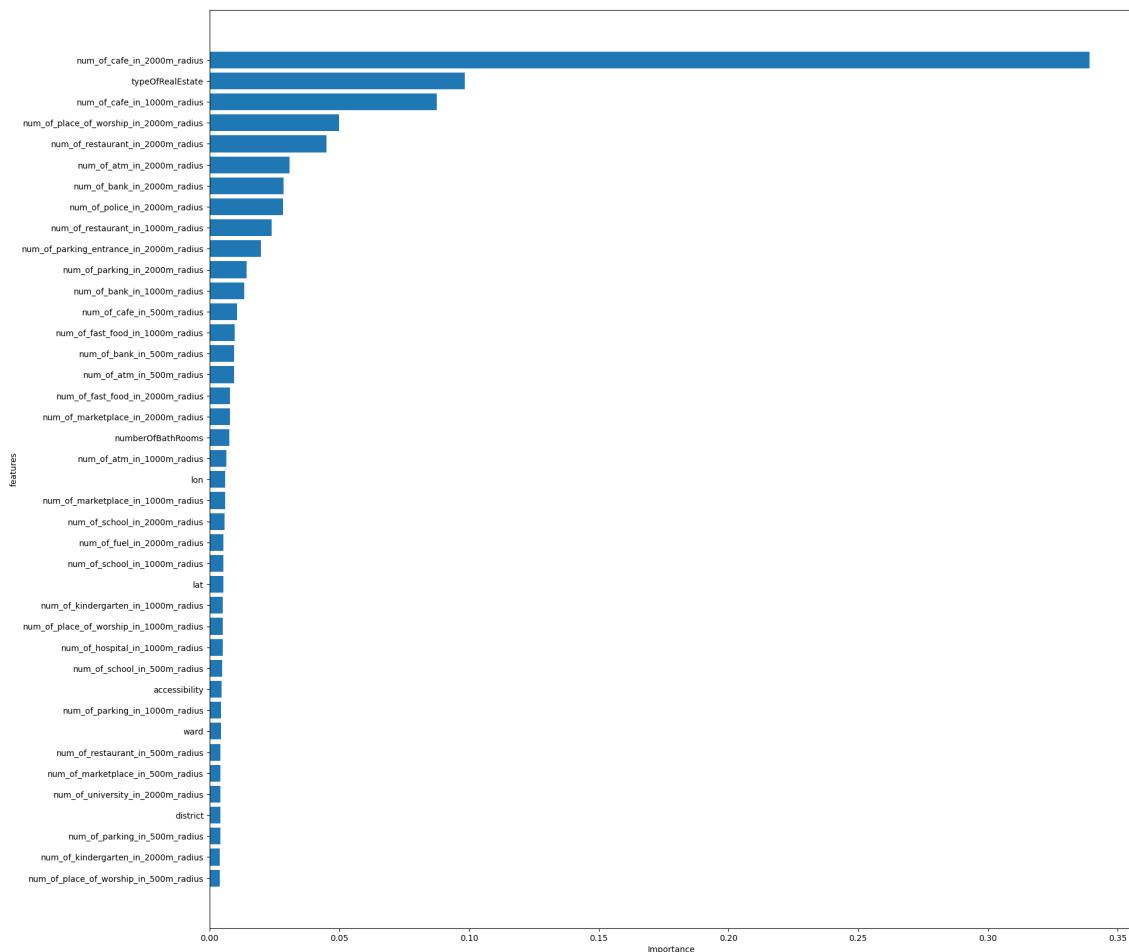
Hình 5.1: Benchmark Featureset and Model Performance on Ha Noi Dataset



Hình 5.2: Benchmark Featureset Version and Model Performance on Ho Chi Minh Dataset

Hình vẽ 5.1, 5.2 thể hiện chất lượng dự đoán trên khía cạnh độ đo lỗi bình phương RMSE, độ đo mức độ giải thích biến động của mô hình và độ lỗi tối đa ở 2 tập dữ liệu Hà Nội và Hồ Chí Minh khi so sánh tập các mô hình học máy cơ bản, bagging và mô hình BKPrice Post Processing (mô hình bagging được hậu xử lý)

Đường nét đứt là đường cơ sở - thể hiện độ lỗi của tập các mô hình khi chỉ dùng tập thuộc tính version 0 (tập thuộc tính gồm những thông tin cơ bản của bất động sản). Độ lỗi trên hầu hết các mô hình khi huấn luyện trên các tập thuộc tính version 1, 2, 4, 5 đều thấp hơn so với phiên bản thuộc tính truyền thống (version 0) mà các phương pháp trước kia sử dụng. Điều này nhấn mạnh được sự ảnh hưởng của thông



Hình 5.3: Độ quan trọng của thuộc tính

tin tiện ích công và các thông tin giữa những bất động sản lân cận lên kết quả định giá của mô hình. Điều này hợp lý với thực tế khi mà các chuyên gia bất động sản đã nhấn mạnh rằng các dịch vụ tiện ích và tiện ích công là một trong những yếu tố ảnh hưởng đến giá bất động sản [23], cơ sở hạ tầng trường học cũng ảnh hưởng đến giá bất động sản [24]. Ta có thể thấy rằng mức độ giải thích của tập mô hình khi chỉ sử dụng phiên bản thuộc tính 0 thấp hơn đáng kể so với mô hình khi sử dụng phiên bản thuộc tính 1, 2, 4, 5. Điều này càng nhấn mạnh hơn được khả năng mô hình hóa và tổng quát hóa trong dự đoán sẽ được cải thiện hơn nếu dữ liệu huấn luyện có thông tin về tiện ích công và thông tin về vị trí bất động sản.

4 đường màu đen, màu xám, màu đỏ và màu hồng thể hiện hiệu năng của tập mô hình khi sử dụng các tập thuộc tính đề xuất phiên bản 1, 2, 4, 5.

Hiệu năng của tập mô hình sử dụng tập thuộc tính phiên bản 4 (version) đã cải tiến chất lượng dự đoán đáng kể so với phương pháp cơ sở. Hình vẽ trên cũng chứng minh rằng model đang mô hình hóa tốt hơn tập dữ liệu khi sử dụng thêm tập thông tin về tiện ích công. Hình vẽ 5.3 mô tả mức độ quan trọng cao nhất của 40 thuộc tính trên mô hình XGB được huấn luyện với tập thuộc tính version 4.

Các thuộc tính số lượng quán cà phê, số lượng nhà hàng có mức độ quan trọng lớn trong quá trình mô hình XGB khớp với dữ liệu huấn luyện. Hình vẽ trên cũng đã nhấn mạnh và phần nào chứng minh được độ quan trọng của thông tin tiện ích công đến giá dự đoán bất động sản của mô hình.

Tiếp đến 3 đường tương ứng với tập thuộc tính version 1, 2, 5 cải tiến chất lượng mô hình so với mô hình sử dụng tập thuộc tính version 4. Mức giải thích độ biến động trên dữ liệu thử nghiệm của tập mô hình huấn luyện trên tập thuộc tính version 1, 2, 5 hầu hết đề cao hơn so với hiệu năng của cùng mô hình khi được huấn luyện trên bộ thuộc tính version 0 (cơ sở) và version 4. Do đó mô hình có khả năng học được sự biến động giá tốt hơn với dữ liệu thử nghiệm trong tương lai. Hình vẽ 5.1 và 5.2 chứng minh rằng mức độ ảnh hưởng của tập thuộc tính được xây dựng từ các bất động sản lân cận cùng với các thuộc tính PCA, GMM component lên giá bất động sản. Việc thêm thuộc tính PCA vào mô hình dự đoán là một cách thức gói gọn thông tin đa chiều của bất động sản về một số chiều quan trọng nhất. Do đó ta thấy ngay độ hiểu quả của tập thuộc tính version 2, 5 so với version 0, 1, 4. Độ lỗi thấp hơn và độ giải thích biến động dữ liệu cao hơn với những mô hình học máy nâng cao như: XGBRegressor, Bagging và Bagging kết hợp hậu xử lý dữ liệu BKPrice Postprocessing. Bên cạnh đó, một dãy chứng nữa về độ đo độ lỗi lớn nhất khi thực nghiệm giữa tập thuộc tính. Ta thấy rằng mức độ lỗi tối đa mà thuật toán có thể có trên tập thuộc tính version 1, 2, 4, 5 cũng đã thấp hơn so với phiên bản tập thuộc tính cơ sở version 0 trên tập dữ liệu Hà Nội. Tuy nhiên ta thấy rõ được sự trái ngược này trên tập dữ liệu Hồ Chí Minh khi mà độ lỗi tối đa có thể nhận được ở phiên bản ít thuộc tính lại thấp hơn so với các phiên bản đề xuất. Do đó ta thấy được sự đánh đổi giữa việc sai sót lớn nhất là bao nhiêu và mức độ trung bình lỗi bình phương trên toàn bộ dự đoán của mô hình. Hơn thế nữa ta thấy được điểm mạnh của giải thuật hậu xử lý kết hợp với mô hình Bagging qua 2 hình vẽ 5.1, 5.2 đó là giải thuật BKPrice Post Processing đã cân bằng tối ưu được mức độ lỗi tối đa (Max Error) bằng cách hậu xử lý bằng *Mix Max Scale*, *Neighbor Price Scale* và *Distribution Scale*. *Min Max Scale* và *Neighbor Price Scale* đảm bảo được kết quả không được vượt quá giá trị lớn nhất và bé nhất từng được ghi nhận trong vùng. *Distribution Scale* đảm bảo được giá trị bất động sản không nằm ra khỏi phân bố giá của vùng. Việc hậu xử lý điều chỉnh giá của mô hình là một cách thêm tri thức hậu nghiệm cho mô hình và xử lý nhiễu.

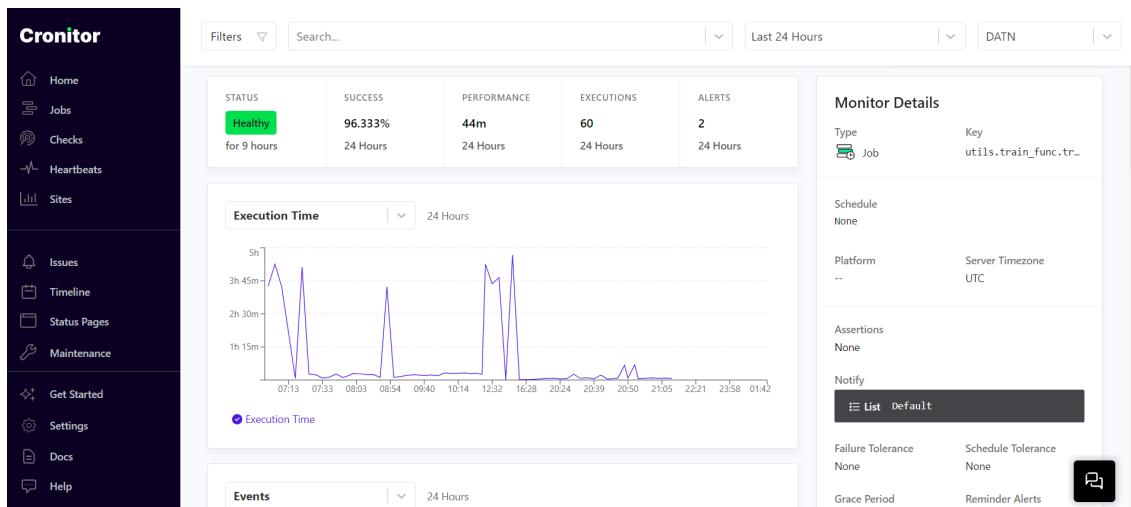
Bảng 5.5, 5.6 mô tả tỷ lệ lỗi RMSE của tất cả mô hình trên 2 tập thuộc tính version 1 và version 5. Kết quả cho thấy việc thử nghiệm tạo thuộc tính từ Gaussian Mixture và PCA hiểu quả trong việc giúp thuật toán mô hình hóa tốt tập dữ liệu bất động sản.

model_name	version 1	version 5
abr	106.3	99.0
cat	56.6	56.1
etr	59.1	59.0
gbr	58.0	57.8
knr	110.8	109.0
la	83.9	83.9
lgbm	56.1	55.9
linear	82.4	82.3
mlp	71.7	70.3
rf	57.2	56.9
ridge	82.4	82.3
xgb	55.8	55.5
bagging	55.5	55.3
BKPrice Post Processing	41.4	41.3

Bảng 5.5: Độ lỗi RMSE trên tập huấn luyện Hà Nội

model_name	Version 1	Version 5
abr	97.8	116.9
cat	63.7	63.8
etr	64.2	63.9
gbr	63.3	63.6
knr	113.8	111.2
la	81.1	81.1
lgbm	60.9	60.9
linear	80.0	80.0
mlp	77.4	76.5
rf	61.2	61.5
ridge	80.0	80.0
xgb	60.6	60.6
bagging	60.2	60.2
BKPrice Post Processing	44.9	44.7

Bảng 5.6: Độ lỗi RMSE trên tập huấn luyện Hồ Chí Minh



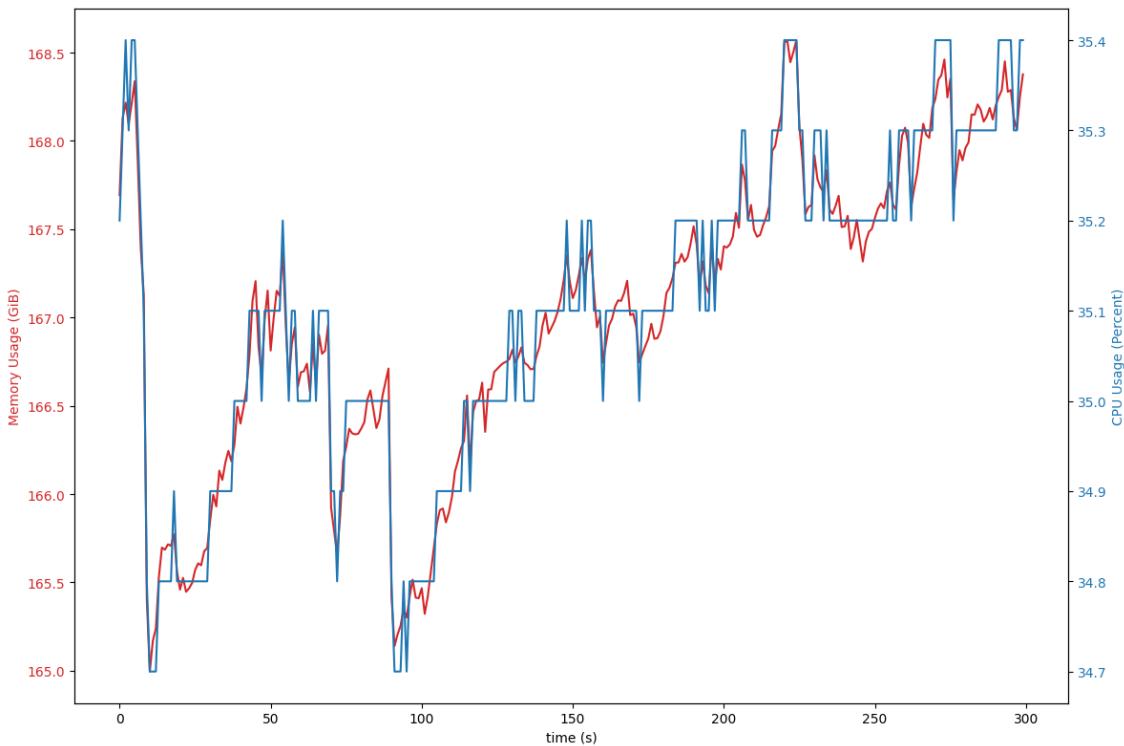
Hình 5.4: Thống kê trung bình thời gian huấn luyện mô hình và tỷ lệ thành công khi huấn luyện

b, Thực nghiệm mức độ ổn định của hệ thống

Hình vẽ 5.3 thống kê tỷ lệ lỗi và thời gian huấn luyện tập mô hình. Ta thấy được ngày tỷ lệ thành công trong các thực nghiệm huấn luyện mô hình tự động với dữ liệu mới đạt ngưỡng 96.333%. Điều này nói lên rằng mức độ tin cậy khi huấn luyện mô hình với dữ liệu mới. Sở dĩ với dữ liệu mới sẽ tồn tại những ngoại lệ trong cách xử lý và làm sạch dữ liệu và huấn luyện mô hình. Do đó, thống kê mức độ thành công của quá trình tự động đảm bảo độ tin cậy của hệ thống rất quan trọng.

Bên cạnh đó, hình 5.3 thống kê độ trễ huấn luyện mô hình tối đa có thể đạt được là bao nhiêu. Tôi đã thử nghiệm chạy tất cả tác vụ sau cùng một lúc: (i) các tác vụ liên quan đến thu thập, làm sạch và lưu trữ dữ liệu, (ii) các tác vụ liên quan đến trích xuất đặc trưng và tạo tập huấn luyện, (iii) các tác vụ training mô hình. Ta thấy rằng trường hợp tệ nhất, mất 4h43m để huấn luyện một mô hình. Tuy nhiên con số này không ảnh hưởng lớn sở dĩ quá trình cập nhật dữ liệu và huấn luyện mô hình chạy ở phía nền của hệ thống BKPrice. Điều này không ảnh hưởng đến trải nghiệm đầu cuối của người sử dụng. Độ trễ lớn nhất có ý nghĩa lớn đối với quá trình tự động hóa trong hệ thống BKPrice. Do đó từ hình vẽ 5.3 ta phần nào chứng minh được mức độ tin cậy của hệ thống BKPrice.

Hình vẽ 5.4 mô tả trong quá trình thực nghiệm thí nghiệm trên, dung lượng bộ nhớ sử dụng và mức độ sử dụng CPU theo từng giây. Chúng ta thấy ngay mức độ sử dụng CPU đạt ngưỡng an toàn ($30 \leq \text{CPU usage percent} \leq 50$), đảm bảo được khả năng chịu tải của hệ thống BKPrice khi thực hiện đa tác vụ cùng một lúc. Mức độ sử dụng dung lượng bộ nhớ ở ngưỡng an toàn. Sở dĩ dung lượng tăng lên từ giây thứ 100, quá trình lưu trữ vào cơ sở dữ liệu đồng thời lưu trữ tập dữ liệu huấn luyện được bắt đầu. Tuy nhiên trong tương lai việc lưu trữ dữ liệu làm sạch và dữ liệu



Hình 5.5: Trạng thái dung lượng bộ nhớ sử dụng và mức độ sử dụng CPU%

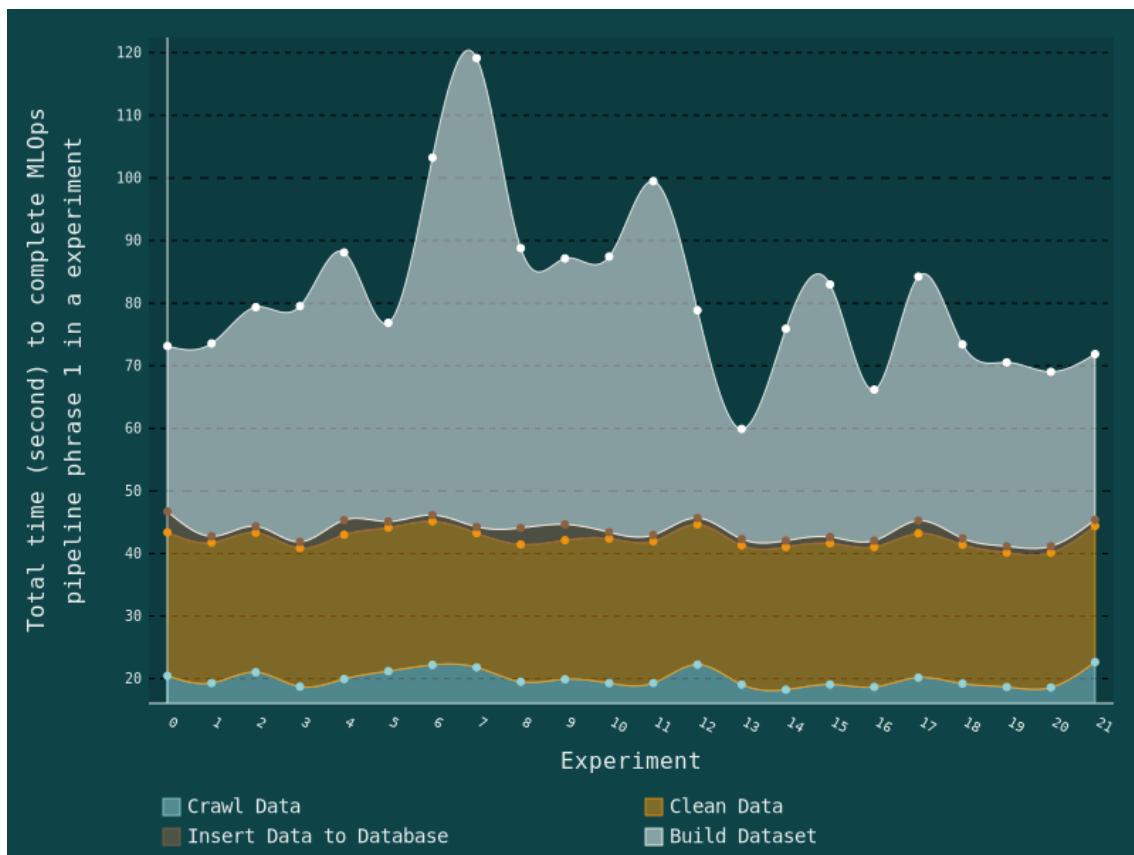
huấn luyện sẽ được lưu trữ trên cách dịch vụ lưu trữ cloud để đảm bảo làm việc và truy xuất dữ liệu lớn tốt hơn.

Tiếp theo tôi xin trình bày thực nghiệm đánh giá thời gian hoàn thành một quy trình MLOps: (i) thu thập 100 mẫu dữ liệu, (ii) làm sạch 100 mẫu dữ liệu, (iii) lưu trữ 100 mẫu dữ liệu vào cơ sở dữ liệu, (iv) Xây dựng tập huấn luyện, (v) Huấn luyện tất cả mô hình AI với tập huấn luyện được lấy từ 5000 mẫu dữ liệu dữ liệu từ tập huấn luyện đã có và 100 mẫu dữ liệu mới thu thập được), (vi) đánh giá mô hình trí tuệ nhân tạo.

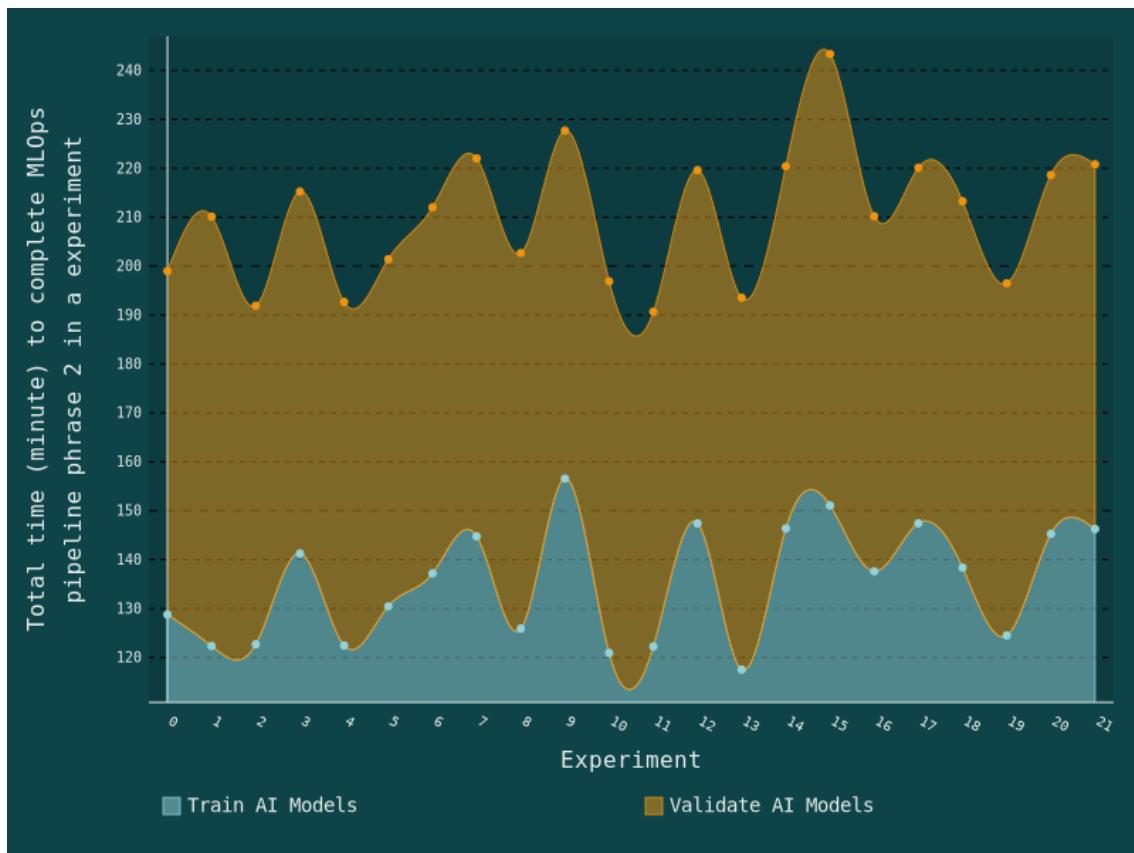
Hình vẽ 5.5 mô tả tổng thời gian để hoàn thành 4 bước đầu tiên của quy trình MLOps. Đối với thử nghiệm 100 mẫu dữ liệu thì thời gian xử lý cho 4 bước đầu tiên khoảng 1 đến 2 phút. Trong đó thời gian nhiều nhất được ghi nhận ở quá trình làm sạch và trích xuất đặc trưng, xây dựng tập huấn luyện.

Hình vẽ 5.6 mô tả tổng thời gian để hoàn thành 2 bước cuối cùng (v) và (vi) của quy trình MLOps. Ở đây tôi thực nghiệm xử lý tuần tự huấn luyện mô hình. Thời gian huấn luyện mô hình dự đoán giá trên tập dữ liệu 5100 mẫu từ 100 phút đến 150 phút. Thời gian hậu xử lý mô hình và thực hiện quá trình đánh giá, lưu thông tin và vẽ bảng biểu khoảng 70 phút đến 110 phút.

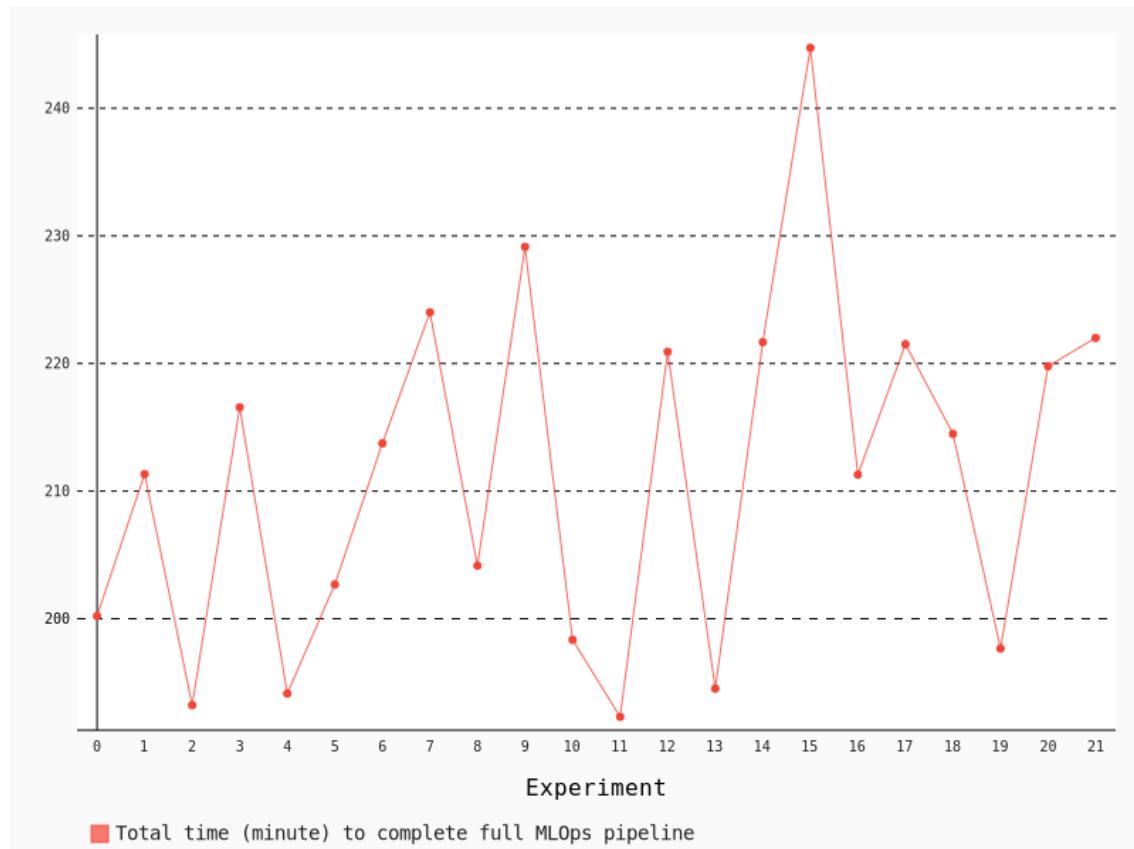
Hình vẽ 5.7 mô tả tổng thời gian để hoàn thành toàn bộ các bước (i) đến bước (vi). Thời gian trung bình là 211 phút, thời gian hoàn thành lâu nhất ghi nhận 244



Hình 5.6: Mô tả thời gian hoàn thành giai đoạn thu thập, làm sạch lưu trữ và xây dựng tập huấn luyện - Phrase 1



Hình 5.7: Mô tả thời gian hoàn thành giai đoạn huấn luyện và đánh giá mô hình - Phrase 2



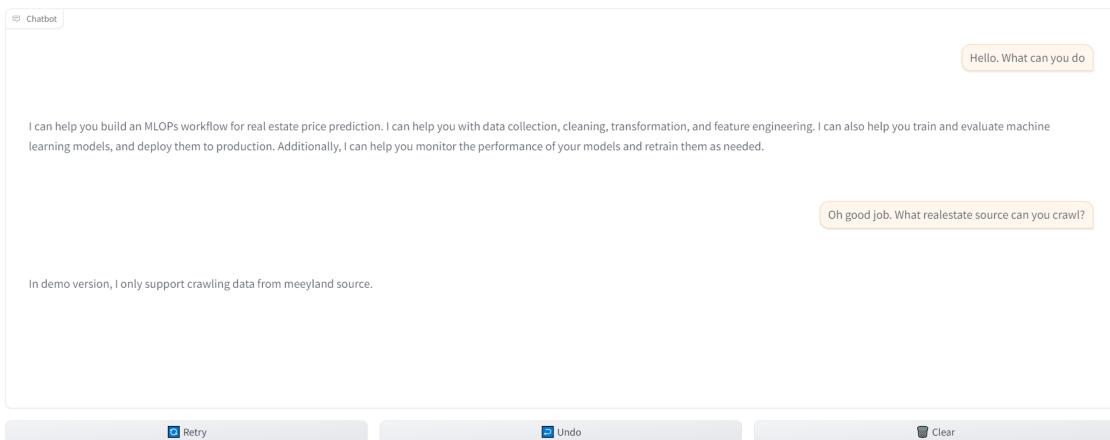
Hình 5.8: Mô tả tổng thời gian hoàn thành toàn bộ luồng MLOps

phút, thời gian hoàn thành quy trình nhanh nhất là 192 phút.

Với những thực nghiệm trên ta thấy được rằng hệ thống BKPrice có thể cập nhật tri thức mới cho các mô hình trí tuệ nhân tạo trong ngày, bắt kịp với tốc độ tạo mới dữ liệu của các nguồn tin bất động sản. Bên cạnh đó kết quả dự đoán cũng được đánh giá tốt hơn các phương pháp truyền thống khi sử dụng nhiều thông tin và thuộc tính ảnh hưởng đến mô hình hơn.

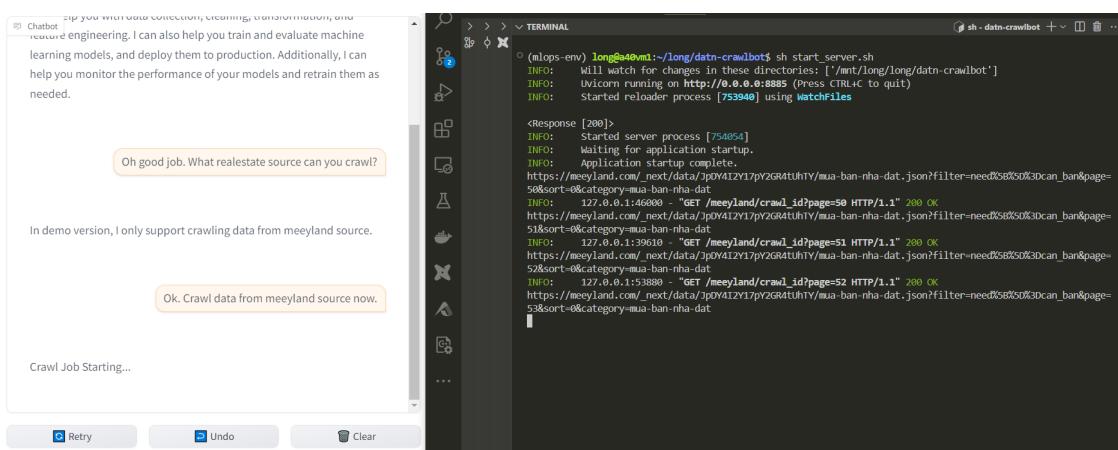
c, Thực nghiệm tương tác với quy trình MLOps bằng mô hình ngôn ngữ lớn

Ở mục này tôi tiến hành thực nghiệm khả năng tương tác giữa mô hình ngôn ngữ lớn với quy trình MLOps của hệ thống BKPrice. Tôi lựa chọn *gemini-pro* trở thành mô hình ngôn ngữ lớn tương tác với hệ thống MLOps bởi vì khả năng hiểu nhiều loại thông tin từ nhiều loại dữ liệu khác nhau. Điều này tạo nên một mô hình ngôn ngữ lớn hiểu rõ bài toán MLOps và tương tác tốt hơn với hệ thống BKPrice.



Hình 5.9: Chat Phrase 1

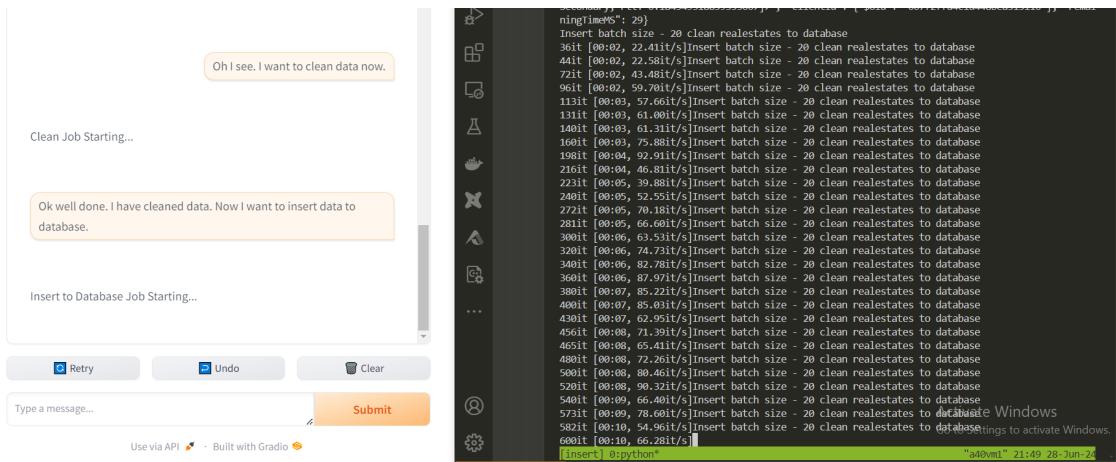
Hình vẽ 5.4 mô tả quá trình tương tác qua lại ban đầu của chatbot và hệ thống BKPrice. Người dùng hỏi thông tin về hệ thống để nắm rõ được hệ thống có thể làm được những gì.



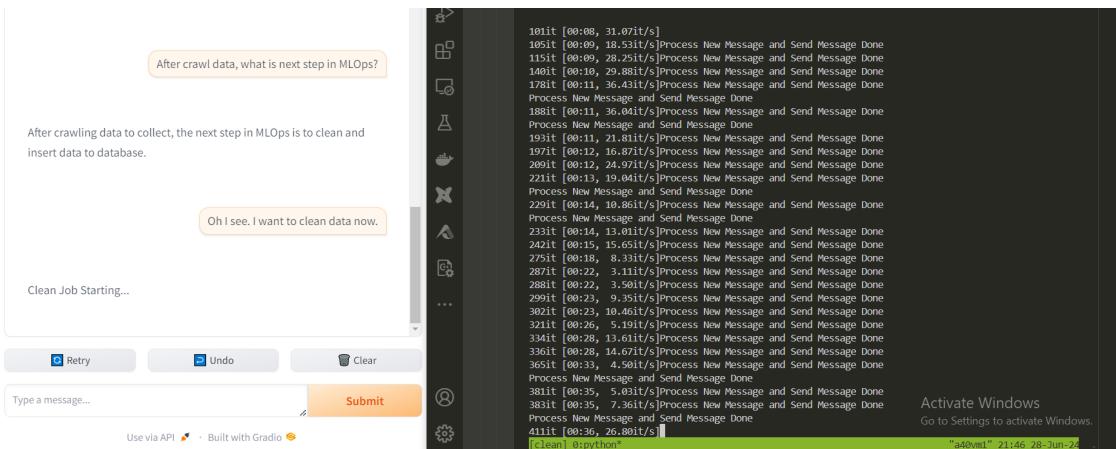
Hình 5.10: Chat Phrase 2

Tiếp theo, người dùng thực hiện yêu cầu thu thập dữ liệu từ nguồn bất động sản meeyland. Màn hình 5.5 mô tả quá trình yêu cầu thu thập từ người dùng đến quá trình phân loại yêu cầu và thực hiện quá trình thu thập dữ liệu trong hệ thống BKPrice (Crawl Server sẽ đảm nhiệm vai trò này xử lý yêu cầu của người dùng ở khía cạnh thu thập dữ liệu)

CHƯƠNG 5. ĐÁNH GIÁ VÀ THỰC NGHIỆM



Hình 5.12: Chat Phrase 4



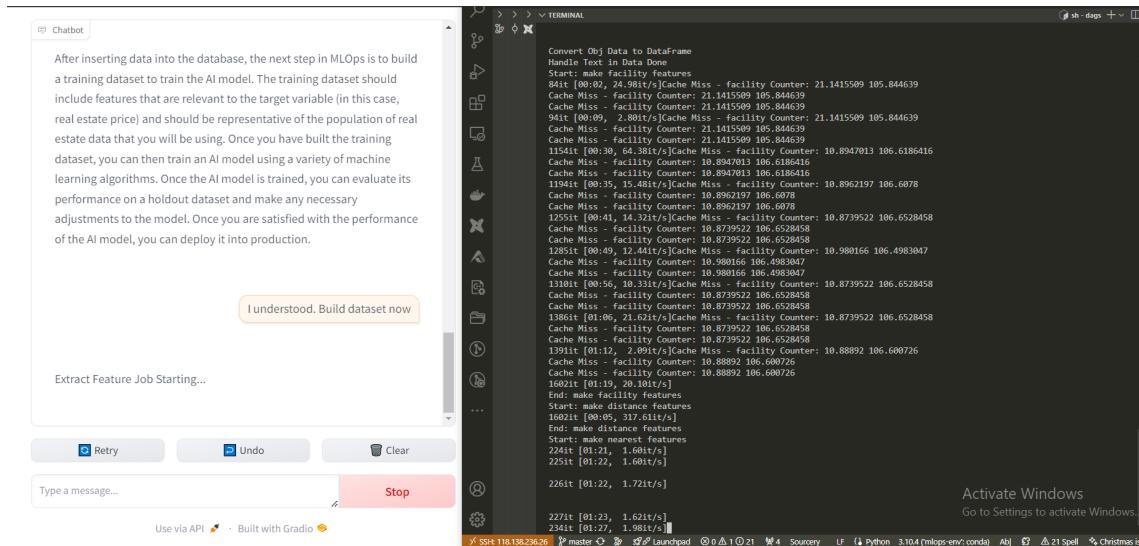
Hình 5.11: Chat Phrase 3

Người dùng có thể yêu cầu khởi động dịch vụ làm sạch và lưu trữ dữ liệu vào cơ sở dữ liệu trong quá trình thu thập dữ liệu. Sở dĩ hệ thống BKPrice bao gồm các mô-đun có độ kết dính nhau tốt nên hỗ trợ người dùng xử lý các tác vụ song song. Màn hình 5.6, 5.7 mô tả hệ thống BKPrice thực hiện quá trình lấy thông tin đã thu thập được (các gói thông điệp từ hàng đợi), tiến hành xử lý thông điệp (thông tin thô về bất động sản) và đẩy vào hàng đợi cho giai đoạn MLOps tiếp theo. Hàng đợi tiếp theo lưu trữ thông tin bất động sản đã được làm sạch. Dịch vụ lưu trữ dữ liệu sẽ đảm nhận nhiệm vụ lấy dữ từ hàng đợi, kiểm tra trùng lặp và thực hiện lưu trữ dữ liệu mới theo lô vào cơ sở dữ liệu.

Người dùng đã thực hiện một luồng cơ bản lấy dữ liệu, làm sạch và lưu trữ dữ liệu trong hệ thống BKPrice. Tiếp theo, người dùng có thể yêu cầu xây dựng tập huấn luyện cho việc xây dựng thuật toán dự đoán giá phía sau. Khi nhận được yêu cầu xây dựng tập huấn luyện, hệ thống BKPrice thực hiện quá trình trích xuất đặc trưng theo các phiên bản tập thuộc tính được cấu hình sẵn trong hệ thống, bao gồm 5 version: verison 0, 1, 2, 4, 5. Quá trình trích xuất đặc trưng bao gồm công đoạn:

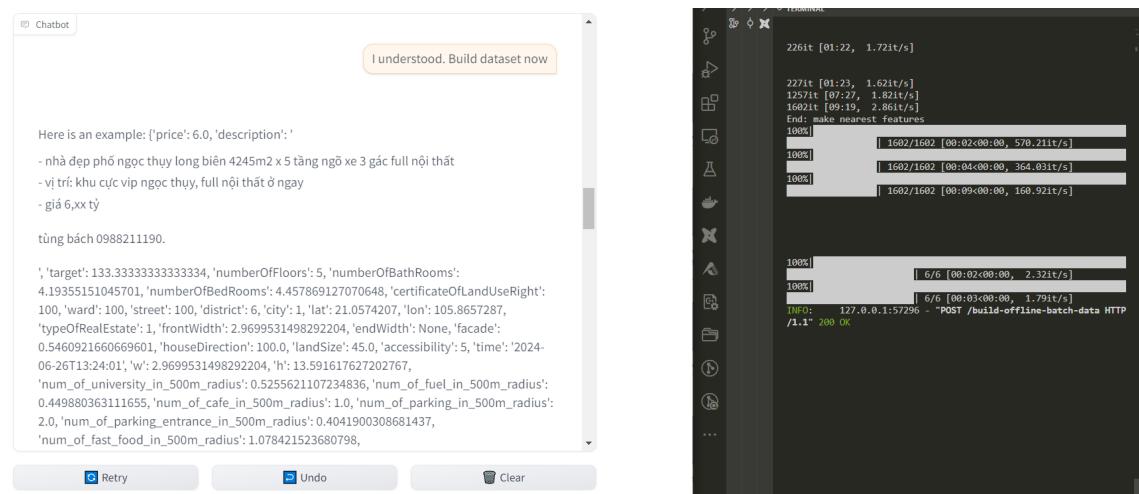
CHƯƠNG 5. ĐÁNH GIÁ VÀ THỰC NGHIỆM

lấy dữ liệu sạch từ cơ sở dữ liệu, tạo thuộc tính dựa vào thông tin tiện ích công, thông tin bất động sản lân cận, GMM và PCA.



Hình 5.13: Chat Phrase 5

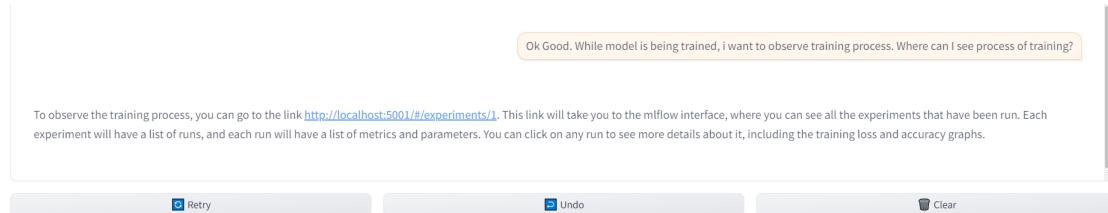
Quá trình xây dựng tập huấn luyện hoàn thành và trả về mẫu cho người dùng. Hình vẽ 5.8, 5.9 trình bày quá trình trích xuất đặc trưng và hình thành tập huấn luyện



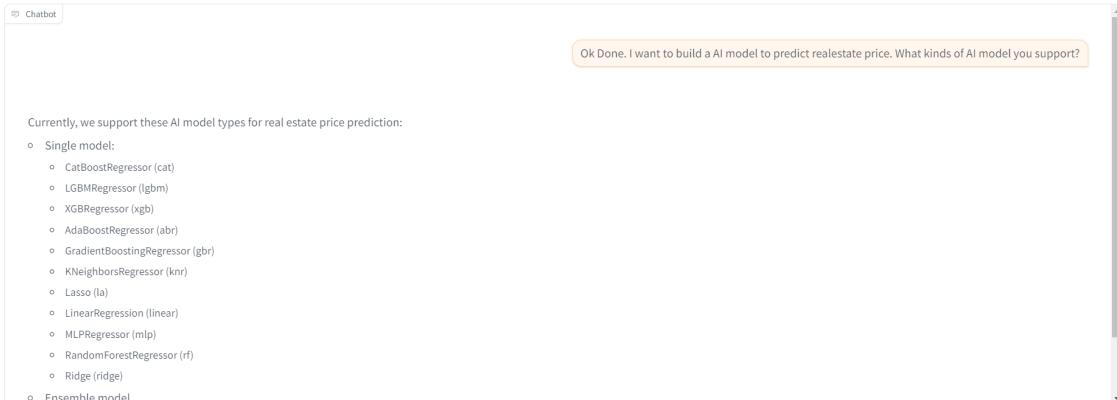
Hình 5.14: Chat Phrase 6

Hình vẽ 5.10, 5.11 mô tả quá trình người dùng hỏi đáp về những loại mô hình được hỗ trợ trong hệ thống BKPrice. Bên cạnh đó việc huấn luyện một mô hình dự đoán giá tổng BKPrice yêu cầu 3 tham số sau: (i) modelname - tên của mô hình, (ii) feature_set_version - phiên bản tập thuộc tính sử dụng để huấn luyện mô hình, (iii) city: dữ liệu thuộc thành phố nào (Hồ Chí Minh, Hà Nội)

CHƯƠNG 5. ĐÁNH GIÁ VÀ THỰC NGHIỆM

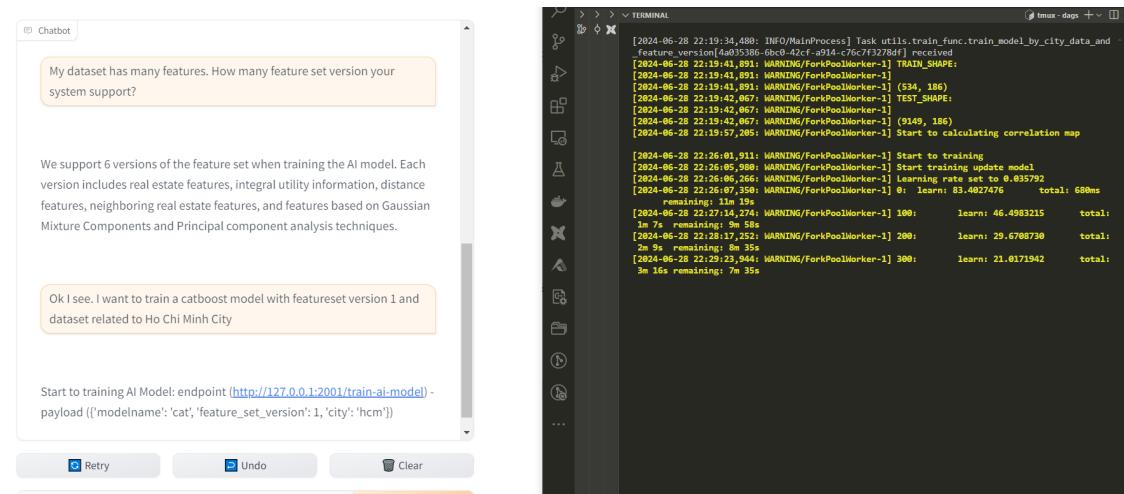


Hình 5.17: Chat Phrase 9



Hình 5.15: Chat Phrase 7

Controller Prompt có vai trò điều hướng mô hình ngôn ngữ lớn và trích xuất ra tập tham số theo yêu cầu. Hình vẽ 5.11 mô tả được thông tin tham số và hàm số được trích xuất ra và tương tác với hệ thống BKPrice. Yêu cầu huấn luyện mô hình đã được xử lý và thực hiện phía nền của hệ thống.



Hình 5.16: Chat Phrase 8

Hình vẽ 5.12 cung cấp thông tin về giao diện hệ thống BKPrice cung cấp để người dùng giám sát quá trình và kết quả huấn luyện mô hình dự đoán giá nhà.

Như chúng ta quan sát toàn bộ quá trình thực nghiệm tương tác giữa BKPrice

System và chatbot, ta thấy được sự tự động của hệ thống thông qua yêu cầu của người dùng. Một quy trình MLOps hoàn thiện được chạy chỉ bằng tương tác hỏi đáp thông qua chatbot. Nó sẽ trở thành 1 bước đệm tốt để biến quá trình quản lý MLOps trong hệ thống BKPrice bằng lập trình thành tương tác hỏi đáp. Khi đó trải nghiệm người dùng và tính thân thiện của hệ thống sẽ được cải thiện.

CHƯƠNG 6. GIẢI PHÁP VÀ ĐÓNG GÓP NỐI BẬT

Chương trước đã đánh giá giải thuật định giá bất động sản và luồng MLOps tự động. Qua đó đảm bảo được tính ổn định, tự động và tin cậy của hệ thống BKPrice. Chương này sẽ tập trung trình bày về những điểm được cho là điểm nhấn chính của hệ thống BKPrice.

6.0.1 Tính mới và tính sáng tạo

Bài toán định giá bất động sản và tính tự động trong các dịch vụ bất động sản đã không còn xa lạ gì trong thực tế. Do đó đã tồn tại rất nhiều giải pháp như giải pháp được cung cấp bởi Biggee. Đây là một giải pháp hiệu quả tuy nhiên nhiên giải pháp yêu cầu nhiều thông tin đặc thù để định giá bất động sản như tờ thửa. Bên cạnh giải pháp này, OneHousing cung cấp một giải pháp định giá bất động sản. Tuy nhiên việc định giá ở giải pháp này chưa đủ tin cậy vì giải pháp chỉ định giá dựa trên địa điểm và còn mang tính thủ công bởi sự xác thực từ bên thứ 3. Mọi ý tưởng đột phá đều bắt nguồn từ việc quan sát một cách kĩ lưỡng và hệ thống BKPrice cũng vậy. Từ việc nhận thấy vấn đề lớn trong bài toán định giá bất động sản là chưa đủ tin cậy trong kết quả đến việc dịch vụ trí tuệ nhân tạo chưa được cập nhật và xử lý tự động với dữ liệu mới, tôi đã khảo sát và nhận ra được những nhược điểm trên. Giải quyết những nhược điểm trên tạo ra một hệ thống BKPrice định giá bất động sản trở nên tin cậy hơn và được thay đổi cập nhật thường xuyên. Vì thế BKPrice là một ý tưởng mang tính mới và sáng tạo.

6.0.2 Tính module hóa và tái sử dụng

Trong lĩnh vực định giá bất động sản, hiện nay các giải pháp như OneHousing, Biggee thường chỉ được phát triển để giải quyết đơn tác vụ. Việc tái sử dụng đang còn mang nhiều khó khăn. Tuy nhiên phát triển một sản phẩm có tính tái sử dụng sẽ tiết kiệm được rất nhiều tài nguyên trong quá trình phát triển hiện tại và làm bàn đạp tốt cho tương lai.

Giải pháp cho vấn đề trên đó chính là thực hiện mô-đun hóa hệ thống ở mức cao. Việc mô-đun hóa hệ thống ở mức cao cho phép hệ thống tách ra dễ dàng hơn thành các thành phần nhỏ, dễ dàng hơn trong việc độc lập các mô-đun và các mô-đun chỉ giao tiếp với nhau qua các giao thức nhất định. Điều này đồng nghĩa với việc chuyên biệt hóa từng thành phần của hệ thống, cho phép tái sử dụng và thay thế một hoặc một số module trong hệ thống.

Hệ thống BKPrice được xây dựng với mức độ mô-đun hóa cao. Nhìn ở mức tổng quan, hệ thống BKPrice. Những khối này có mức độ tách rời nhau cao, đến mức

mỗi khối chỉ giao tiếp với nhau qua API. Điều này cho phép hệ thống dễ dàng thay thế được hoặc tái sử dụng lại ở các hệ thống khác. Mức độ kết dính thấp giữa các mô-đun được nhìn rõ qua các điểm sau đây: (i) Quá trình thu thập dữ liệu được diễn ra độc lập với quá trình xử lý dữ liệu. Đặc biệt quá trình xử lý lỗi cũng xảy ra độc lập với quá trình xử lý dữ liệu chính. (ii) Quá trình lưu trữ dữ liệu được diễn ra độc lập. (iii) Quá trình trích xuất đặc trưng, xây dựng tập huấn luyện được diễn ra độc lập với quá trình huấn luyện mô hình. (iii) Quá trình giám sát, đánh giá mô hình và triển khai mô hình cũng được diễn ra độc lập với nhau. Các khối trong BKPrice được gắn với nhau bằng cách gọi qua API hoặc gọi qua thư viện.

Tóm lại các thành được diễn ra một cách độc lập, nếu không phải là bài toán bất động sản thì BKPrice sẽ trở thành một luồng MLOPs pipeline hoàn chỉnh với dịch vụ trí tuệ nhân tạo bất kỳ nếu đảm bảo được các yếu tố sau: (i) Thay thế các nghiệm vụ thu thập dữ liệu bất động sản bằng các dịch vụ thu thập khác tương ứng, (ii) Thay thế nghiệm vụ huấn luyện mô hình, (iii) Điều chỉnh các độ đo đánh giá mô hình phù hợp.

Do đó BKPrice không chỉ dừng lại là một hệ thống dự đoán giá bất động sản cùng với luồng MLOPs tự động mà BKPrice sẽ trở thành một luồng MLOps tự động dễ dàng thay đổi để đáp ứng cho các miền bài toán khác nhau vì sự tách rời giữa các mô-đun và có thể kế thừa: hệ thống thu thập dữ liệu , cách giao tiếp với hệ thống làm sạch và lưu trữ dữ liệu, huấn luyện mô hình và đánh giá mô hình.

CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

7.1 Kết luận

Triển khai dịch vụ trí tuệ nhân tạo là một bài toán khó đặc bẩm bảo tính tự động và tin cậy của các dịch vụ AI là một trong những vấn đề lớn hiện nay. Tư duy thiết kế và tiếp cận bài toán trí tuệ nhân tạo truyền thống trở nên kém hiểu quả khi bài toán dự đoán giá đặc thù cần được cập nhật dữ liệu và mô hình theo thời gian. Vì vậy đặt ra một vấn đề rằng cần có một giải pháp mang tính tự động và tin cậy trong bài toán dự đoán giá bất động sản để giảm thiểu rủi ro và chi phí. Đã có nhiều giải pháp cho bài toán về khía cạnh giải thuật và cách tiếp cận bài toán trong đó phải kể đến giải pháp đến từ Biggee. Đây là một giải pháp hiểu quả tuy nhiên giải pháp yêu cầu nhiều thông tin đặc thù để định giá bất động sản như tờ thửa. Bên cạnh giải pháp này, OneHousing cung cấp một giải pháp định giá bất động sản. Tuy nhiên việc định giá ở giải pháp này chưa đủ tin cậy vì giải pháp chỉ định giá dựa trên địa điểm và còn mang tính thủ công bởi sự xác thực từ bên thứ 3. Do đó nghiên cứu đi đến một quyết định đột phá để giải quyết những nhược điểm này, một hệ thống mang tên BKPrice. Ý tưởng của hệ thống là xây dựng một quy trình dự đoán giá bất động sản đủ tin cậy về mặt đầu ra của mô hình, đánh giá mô hình và triển khai mô hình một cách tự động. Hệ thống cung cấp dịch vụ dự đoán giá bất động sản và một luồng MLOps tự động. Để đạt được độ tự động và độ tin cậy, hệ thống BKPrice thực hiện 2 nhiệm vụ chính sau: (i) Trích xuất đa đặc trưng, xây dựng dữ liệu và huấn luyện mô hình với giải thuật định giá tự động, (ii) Đánh giá mô hình và triển khai mô hình tự động. Ở trong nghiên cứu này, hệ thống BKPrice đã giải quyết triệt để hai nhiệm vụ trên:

- Hệ thống BKPrice đã có luồng thu thập dữ liệu, làm sạch dữ liệu một cách tự động. Từ những dữ liệu đó, hệ thống thực hiện trích xuất bộ đặc trưng và xây dựng tập dữ liệu huấn luyện cho các mô hình trí tuệ nhân tạo. Để đảm bảo được tính tin cậy của kết quả dự đoán, hệ thống đã thực hiện kết hợp các mô hình và hậu xử lý kết quả. Các bước xử lý luôn được tối ưu để đảm bảo hệ thống hoạt động tốt hơn, có khả năng triển khai và mở rộng sau này.
- Hệ thống BKPrice đã có cơ chế đánh giá những mô hình và triển khai mô hình tự động. Hệ thống đã giám sát mô hình huấn luyện dựa vào những độ đo từ đó nhận biết được mô hình huấn luyện tốt ở đâu, tệ ở đâu và điều chỉnh một cách hợp lý

Kết thúc lại, hệ thống BKPrice đã đáp ứng được những yêu tố được đề cập trong xuyên suốt nghiên cứu: (i) Tính mới, (ii) Tính tự động, (iii) Tính tin cậy, (iv) Tính

triển khai.

Qua những vấn đề và giải pháp được đề cập nói chung, hệ thống BKPrice nói riêng, nghiên cứu có những đóng góp nổi bật sau:

- Kế thừa những điểm mạnh và giải quyết những tồn đọng của giải pháp đã có.
- Đề xuất giải thuật dự đoán giá bất động sản tin cậy dựa vào đa mức thông tin
- Xây dựng luồng MLOps tự động cho bài toán bất động sản và dễ dàng sửa đổi để sẵn sàng cho các dịch vụ trí tuệ nhân tạo khác.
- Làm bước đệm tốt cho các hệ thống khác trong tương lai

Trong quá trình nghiên cứu, tôi đã học hỏi được nhiều kiến thức và kinh nghiệm. Bên cạnh (i) việc nâng cao khả năng tìm kiếm thông tin, khảo sát thị trường, tôi còn (ii) rèn luyện khả năng đặt ra câu hỏi cho chính sản phẩm mình ra đã đáp ứng thị trường bất động sản như thế nào và còn tồn đọng những gì. Hơn thế nữa, (iii) phân tích thiết kế hệ thống lớn nhiều thành phần, chức năng, (iv) cách vận dụng những kiến thức đã được học tại trường đại học Bách Khoa Hà Nội và kiến thức tìm tòi để tạo nên một nền tảng vững chắc để phát triển sản phẩm, (v) học cách đánh giá các sản phẩm lớn và áp dụng cho sản phẩm của mình, cuối cùng đó là (vi) sự tiếp thu ý kiến từ mọi người xung quanh để cải thiện sản phẩm hơn.

7.2 Hướng phát triển trong tương lai

Mặc dù đã giải quyết được đa số vấn đề tồn đọng của các giải pháp định giá bất động sản tuy nhiên hệ thống vẫn có thể tiếp tục phát triển theo các hướng sau:

- Cung cấp dịch vụ dự đoán bất động sản toàn bộ khu vực trên toàn đất nước Việt Nam và quốc tế. Việc đáp ứng không chỉ trong nước mà còn toàn cầu giúp sản phẩm có thể đi xa và nhận được nhiều lời đánh giá khách quan của bạn bè thế giới. Khi số lượng người dùng tăng lên đáng kể sẽ xuất hiện nhiều vấn đề cần tối ưu trong hệ thống.
- Phát triển các tính năng mới bên trong hệ thống BKPrice: gợi ý bất động sản bạn có thể thích cá nhân hóa theo từng người dùng, so sánh bất động sản và gợi ý danh sách bất động sản nên đầu tư.
- Cung cấp một luồng MLOps tự động cho tất cả bài toán trí tuệ nhân tạo không chỉ dừng lại ở định giá bất động sản. Ví dụ: các bài toán dự đoán giá chứng khoán, phân tích hành vi sử dụng mạng xã hội.
- Tích hợp luồng xử lý lớn. Điều này cần thiết đối với các bài toán đặc thù mang yếu tố thời gian thực như dự đoán giá trong thị trường tiền ảo chặng hạn. Dữ liệu trong lĩnh vực này tăng lên một cách nhanh chóng, vì vậy phát triển các

gói xử lý dữ liệu lớn sẽ là một hướng phát triển trong tương lai.

CHƯƠNG 8. PHỤ LỤC

8.0.1 Các trường hợp dự đoán giá bất động sản trong thực tế

The screenshot shows a house listing for a 150m² property on Tran Duy Hung street. The listing includes details like price (30ty), area (150 m²), and location (Cau Giay, Hanoi). A note indicates a 19.9% increase in value over the last year. To the right, a Postman API test case is shown for predicting real estate prices. The request URL is `(domain)/predict-realestate-batch`, and the JSON body contains the property details. The response shows a single result with ID 225.48833893826123.

Hình 8.1: Test Case 1

Hình ảnh 8.1 mô tả dự đoán giá nhà đường Trần Duy Hưng, Phường Trung Hòa, Cầu Giấy, Hà Nội của BKPrice System so sánh với giá trên trang batdongsan.com [25]

The screenshot shows a house listing for a 36m² property on Hang Bai street in Hoan Kiem district. The listing includes details like price (12ty), area (36 m²), and location (Hoan Kiem, Hanoi). A note indicates a 43.4% increase in value over the last year. To the right, a Postman API test case is shown for predicting real estate prices. The request URL is `(domain)/predict-realestate-batch`, and the JSON body contains the property details. The response shows a single result with ID 356.512096771762384.

Hình 8.2: Test Case 2

Hình ảnh 8.2 mô tả dự đoán giá nhà phố Hàng Bài, Phường Hang Bai, Hoàn Kiếm, Hà Nội của BKPrice System so sánh với giá trên trang batdongsan.com [26]

CHƯƠNG 8. PHỤ LỤC

The screenshot shows a house listing on the left and a Postman API request on the right.

House Listing Details:

- Mức giá: 5,79 tỷ
- Diện tích: 58 m²
- Phòng ngủ: 3 PN
- Địa chỉ: 343/ Nguyễn Trọng Tuyển, Phường 1, Quận Tân Bình
- Kết cấu: Nhà 1 trệt, 2 lầu.
- Giá còn thương lượng 5 tỷ 790 triệu đồng.
- Nhà trong hẻm rộng rãi, hiện nhà tôi đang cho thuê 15 triệu/tháng. Anh chị nào cần ở liền cứ báo với tôi.
- Nhà mới sửa lại nên ko cần phải sửa gì, vỏ lái ở ngay.
- Khu dân đông, có camera quan sát, an ninh. Tiện về kinh doanh mua bán. Gần bách hóa xanh, family mart, gần trường học...

Postman API Request (Body JSON):

```

{
    "id": 1,
    "landSize": 58,
    "city": "HCMC",
    "district": "tan binh",
    "ward": "Phuong 1",
    "street": "Nguyen Trong Tuyen",
    "prefixDistrict": "quận",
    "numberOffloors": 1,
    "numberOfBathRooms": 4,
    "numberOfLivingRooms": 3,
    "endWidth": 4,
    "frontWidth": 4,
    "frontRoadWidth": 5,
    "certificatedOrLandUseRight": true,
    "typeOfRealEstate": "privateProperty",
    "facade": "oneSideOpen",
    "houseDirection": "east",
    "facility_check_ok": true,
    "narrow_alley": 3,
    "version": "v5"
}
  
```

Hình 8.3: Test Case 3

Hình ảnh 8.3 mô tả dự đoán giá nhà đường Nguyễn Trọng Tuyển, Phường 1, Tân Bình, Hồ Chí Minh của BKPrice System so sánh với giá trên trang batdongsan.com [27]

The screenshot shows a house listing on the left and a Postman API request on the right.

House Listing Details:

- Mức giá: 17 tỷ
- Diện tích: 47 m²
- Đường vào: Mật tiễn 4.07 m
- Thông tin mô tả

Postman API Request (Body JSON):

```

{
    "id": 1,
    "landSize": 47,
    "city": "HCMC",
    "district": "Giai Da",
    "ward": "Phuong Mai",
    "street": "Giai Phong",
    "prefixDistrict": "quận",
    "numberOffloors": 5,
    "numberOfBathRooms": 6,
    "numberOfLivingRooms": 6,
    "endWidth": 4,
    "frontWidth": 4.07,
    "frontRoadWidth": 7,
    "certificatedOrLandUseRight": true,
    "typeOfRealEstate": "townhouse",
    "facade": "twoSideOpen",
    "houseDirection": "east",
    "facility_check_ok": true,
    "narrow_alley": 1,
    "version": "v5"
}
  
```

Hình 8.4: Test Case 4

Hình ảnh 8.4 mô tả dự đoán giá nhà đường Giải Phóng, Phường Phương Mai, Đồng Đa, Hà Nội của BKPrice System so sánh với giá trên trang batdongsan.com [28]

The screenshot shows a house listing on the left and its corresponding JSON representation on the right.

House Listing Details:

- Mức giá: 50 tỷ (~649.35 triệu/m²)
- Diện tích: 77 m²
- Phòng ngủ: 18 PN
- Số tầng: 7 tầng
- Giá tại khu vực này đã giảm trong vòng 1 năm qua: 7.4%
- Xem lịch sử giá >

JSON Representation:

```

11     ..... "body": {
12     ..... "landSize": 77,
13     ..... "city": "HCM",
14     ..... "district": "Quận 1",
15     ..... "street": "Ông Ông Lãnh",
16     ..... "street2": "Nguyễn Thái Học",
17     ..... "prefixDistrict": "quận",
18     ..... "numberofFloors": 7,
19     ..... "numberofBathRooms": 19,
20     ..... "numberofLivingRooms": 18,
21     ..... "endWidth": 5,
22     ..... "frontDepth": 9,
23     ..... "ironFrontDepth": 30,
24     ..... "certificateOfLandTitle": true,
25     ..... "typeOfRealEstates": "townhouse",
26     ..... "facade": "oneSideOpen",
27     ..... "houseOrientation": "east",
28     ..... "facility_check_ok": true,
29     ..... "narrow_alley": 3,
30     ..... "version": "v5"

```

Body: [{ 2: 629.7499875192859 }]

Hình 8.5: Test Case 5

Hình ảnh 8.5 mô tả dự đoán giá nhà đường Nguyễn Thái Học, Phường Cầu Ông Lãnh, Quận 1, Hồ Chí Minh của BKPrice System so sánh với giá trên trang batdongsan.com [29]

8.0.2 Quản lý và đánh giá mô hình

The screenshot shows the BKPrice System interface with two main sections: 'Model metrics' and 'System metrics'.

Model metrics:

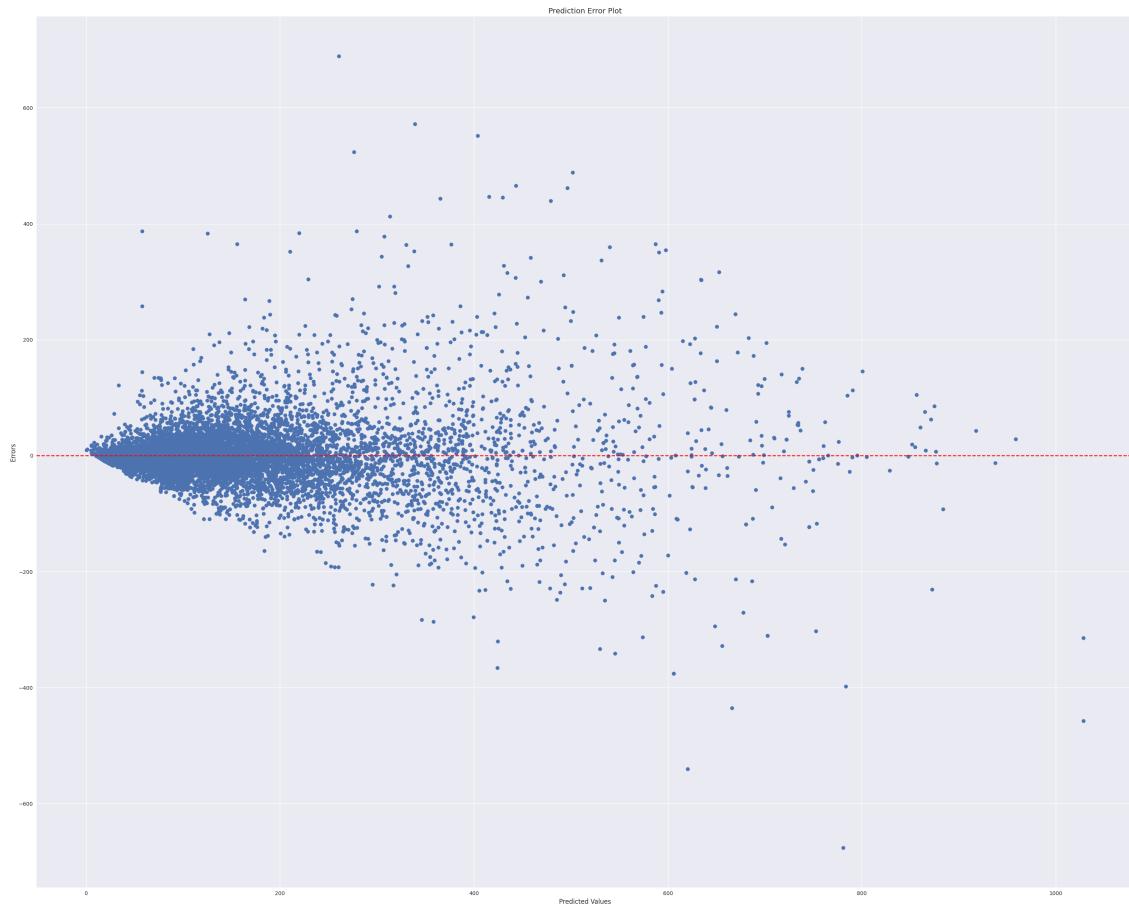
Parameter	Value
cv	5
error_score	nan
estimator	BKPriceEstimator(pretrained_model_path='/home/long/airflow/dags/models/hcm/xgb/v5/model.joblib', update_model=XCGBRegressor(base_score=None, booster=None, callbacks=None, colsample_bylevel=None, colsample_bynode=None, colsample_bytree=None, device=None, early_stopping_rounds=None, enable_categorical=False, eval_metric=None, feature_types=None, gamma=None, grow_policy=None, importance_type=None, interaction_constraints=None, learning_rate=0.04, max_bin=None, max_cat_threshold=None, max_cat_to_onehot=None, max_delta_step=None, max_depth=None, max_leaves=None, min_child_weight=None, missing=nan, monotone_constraints=None, multi_strategy=None, n_estimators=1000, n_jobs=None, num_parallel_tree=None, random_state=822, ...))
n_jobs	None
best_weight	-0.001
param_grid	{'weight': [-0.001, 0, 0.001]}

System metrics:

Metric	Value
mean_test_explained_variance	0.09347455771955002
mean_test_max_error	-466.0792941509377
mean_test_neg_root_mean_squared_error	-69.8432737819157
mean_test_r2	-0.20984045256971742
mean_test_neg_mean_absolute_percent	-3.857111679728928
training_mean_squared_error	6531.8169657111776
training_mean_absolute_error	40.491152193361856
training_r2_score	-0.01375576745004587
training_root_mean_squared_error	80.81965705019897
training_score	-80.81965705019897
best_cv_score	-69.8432737819157
std_test_r2	0.3430068622092332
std_test_explained_variance	0.2622444081522733
std_test_max_error	510.62460764640736
std_test_neg_root_mean_squared_error	40.55842208118176
std_test_neg_median_absolute_error	4.8677937079584375
std_test_neg_mean_absolute_percent	0.8468405048144864
mean_test_neg_median_absolute_error	-31.454972941943918

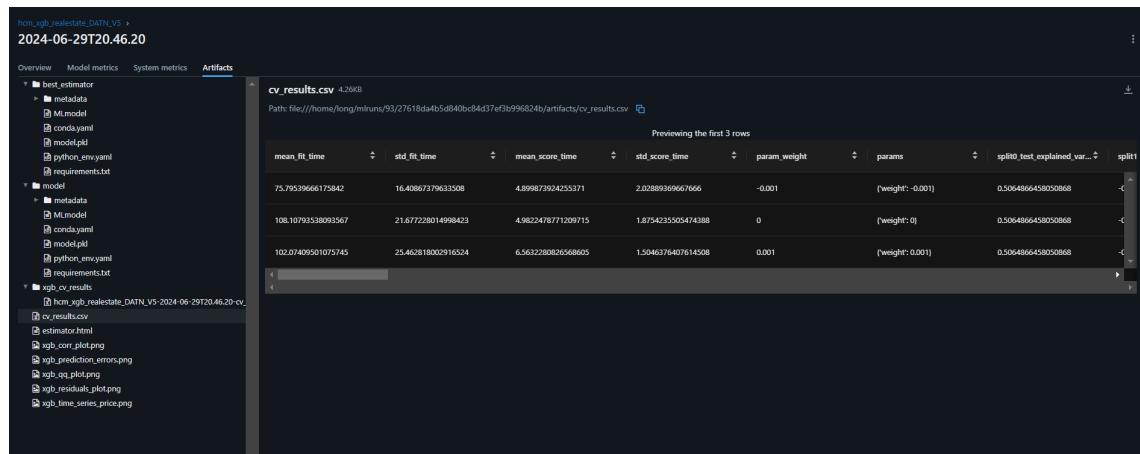
Hình 8.6: Thông tin huấn luyện và đánh giá mô hình

Hình ảnh 8.6 mô tả thông tin huấn luyện mô hình và đánh giá mô hình dự đoán giá bất động sản. Bên cạnh đó kèm thông tin tham số tốt nhất trong quá trình tuning mô hình.



Hình 8.7: Biểu đồ độ lỗi dự đoán trích xuất tự động

Hình ảnh 8.7 mô tả độ lỗi dự đoán trích xuất tự động sau khi quá trình huấn luyện mô hình kết thúc.



Hình 8.8: Thông tin mô hình sau khi huấn luyện kết thúc

Hình ảnh 8.8 mô tả thông kê các độ đo sau khi huấn luyện mô hình: thời gian huấn luyện trung bình, kết quả đánh giá mô hình. Bên cạnh đó hình ảnh cho thấy các thông tin checkpoint mô hình cũng được lưu trữ phiên bản tốt nhất cho lần chạy này cùng với danh sách các bảng biểu khác như: (i) Đo độ tương quan của tập thuộc

tính, (ii) bảng thông tin kết quả đánh giá mô hình

8.0.3 Cơ sở dữ liệu

The screenshot shows a MongoDB interface with the following details:

- Left Sidebar:** Shows 'My Queries', 'Performance', 'Databases' (with 'realestate' selected), and a 'Search' bar.
- Top Bar:** Shows 'Documents' (396.2K), 'Aggregations', 'Schema', 'Indexes' (1), and 'Validation'.
- Search Bar:** 'Type a query: { field: 'value' } or Generate query +'
- Action Buttons:** 'ADD DATA', 'EXPORT DATA', 'UPDATE', 'DELETE'.
- Document Preview Area:** Displays five documents from the 'realestate_url_pool' collection. Each document has the following structure:


```
_id: ObjectId('66721d48a483ab60e9f38f31')
crawl_at: 2024-06-18T23:50:32.500+00:00
url: "https://batdongsan.com.vn/ban-dat-duong-ba-diem-4-xa-ba-diem/n-gap-lo-"
source: "batdongsan"
```

```
_id: ObjectId('66721d48a483ab60e9f38f33')
crawl_at: 2024-06-18T23:50:32.500+00:00
url: "https://batdongsan.com.vn/ban-dat-duong-khuat-van-buc-xa-tan-kien/ban-"
source: "batdongsan"
```

```
_id: ObjectId('66721d48a483ab60e9f38f34')
crawl_at: 2024-06-18T23:50:32.500+00:00
url: "https://batdongsan.com.vn/ban-nha-rieng-duong-ngo-chi-quoc-phuong-binh-"
source: "batdongsan"
```

```
_id: ObjectId('66721d48a483ab60e9f38f35')
crawl_at: 2024-06-18T23:50:32.500+00:00
url: "https://batdongsan.com.vn/ban-dat-duong-lien-xa-xa-tan-tien-16/100m-fu-"
source: "batdongsan"
```

```
_id: ObjectId('66721d48a483ab60e9f38f37')
crawl_at: 2024-06-18T23:50:32.500+00:00
url: "https://batdongsan.com.vn/ban-nha-rieng-duong-phan-xich-long-phuong-3-"
```
- Bottom Right:** 'Activate Windows' and 'Go to Settings to activate Windows.'

Hình 8.9: Cơ sở dữ liệu bất động sản

Hình ảnh 8.9 mô tả kho lưu trữ danh sách url được thu thập để thực hiện quá trình crawl dữ liệu.

The screenshot shows a MongoDB interface with the following details:

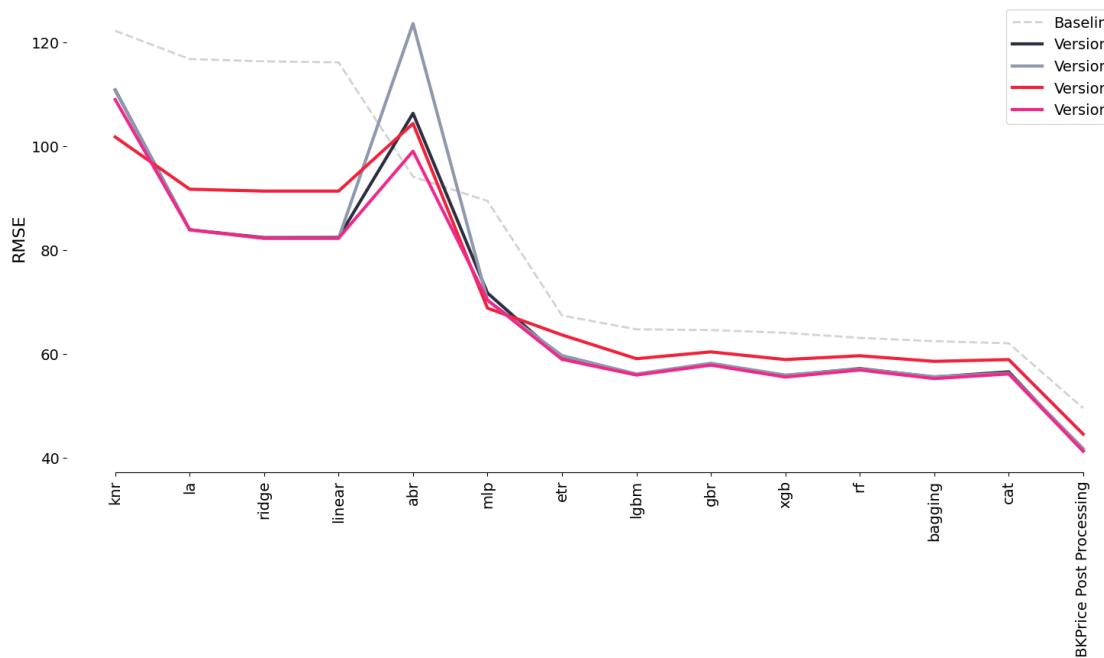
- Top Bar:** 'ADD DATA', 'EXPORT DATA', 'UPDATE', 'DELETE'.
- Document Preview Area:** Displays a single document from the 'realestate' collection. The document structure is expanded to show nested objects:


```
propertyBasicInfo : Object
  landType : Object
    comment : Array (empty)
    status : "UNSELECTED"
    value : "residentialLand"
  accessibility : Object
  distanceToNearestRoad : Object
  frontRoadWidth : Object
  address : Object
    comment : Array (empty)
    status : "UNSELECTED"
  value : Object
    addressDetails : ""
    street : "Lĩnh Nam"
    ward : "Lĩnh Nam"
    district : "Hoàng Mai"
    city : "Hà Nội"
    country : "Việt Nam"
  description : Object
  geolocation : Object
  typeOfRealEstate : Object
  frontWidth : Object
  endWidth : Object
  facade : Object
  houseDirection : Object
  landSize : Object
  contact : Object
  price : Object
```
- Bottom Right:** '1 - 20 of 40429'

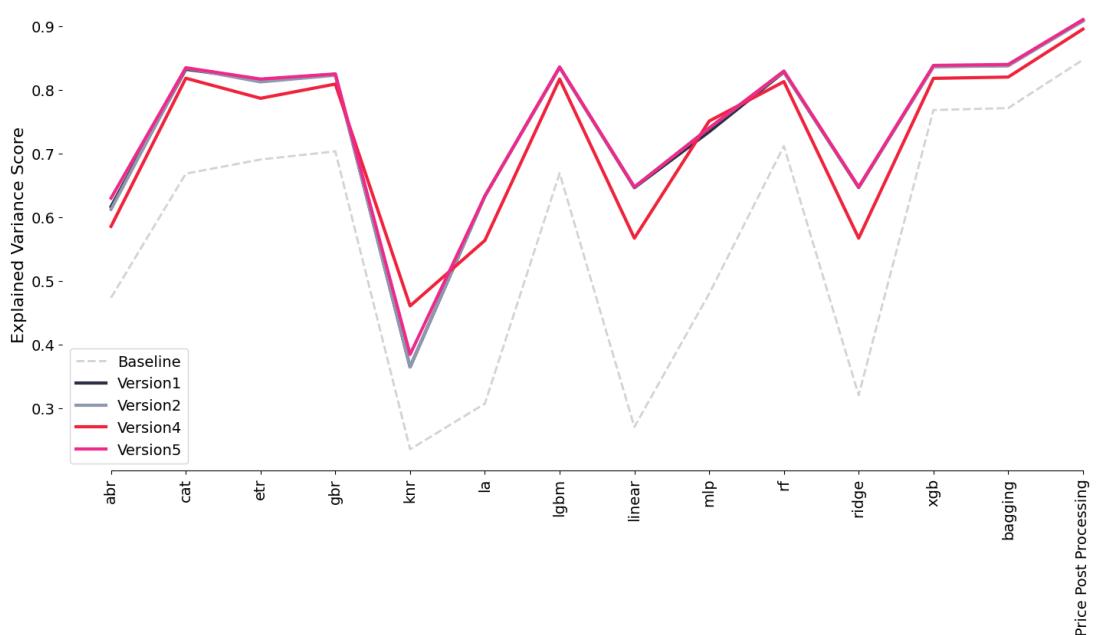
Hình 8.10: Cơ sở dữ liệu bất động sản

Hình ảnh 8.10 mô tả chi tiết thông tin bất động sản đã được làm sạch và lưu trữ.

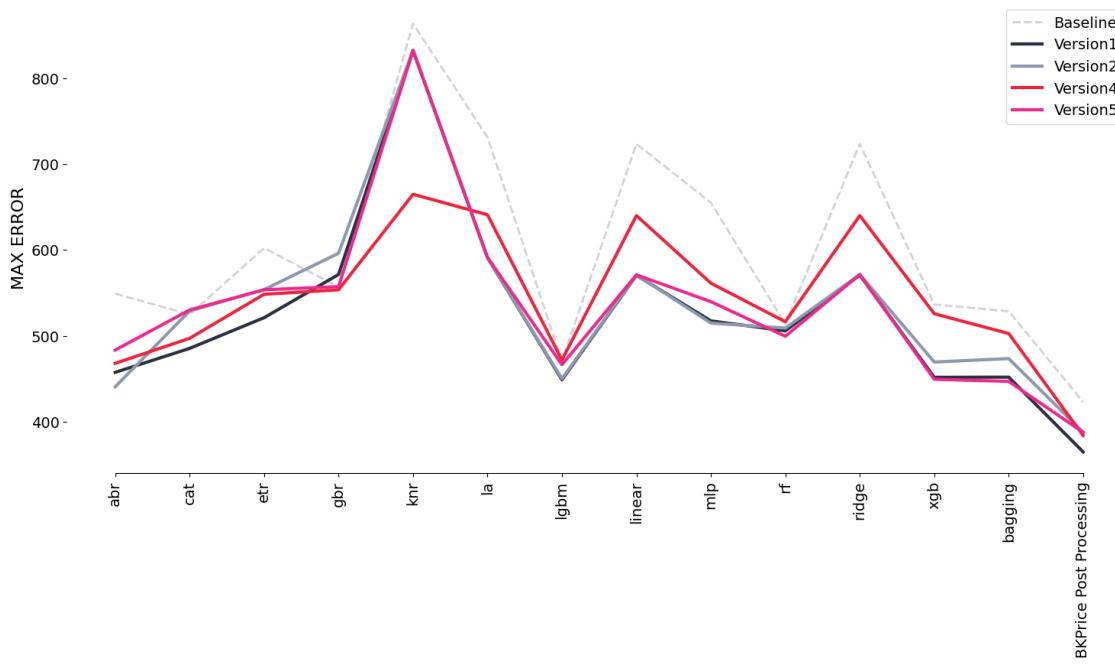
8.0.4 Biểu đồ benchmark kết quả



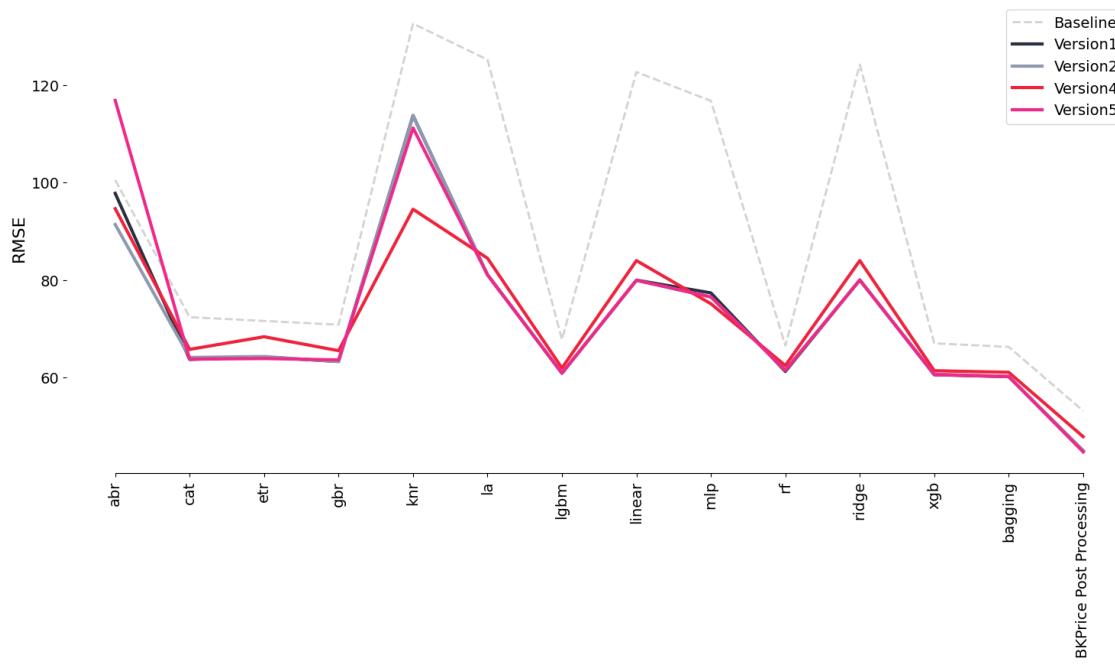
Hình 8.11: HN Data - RMSE



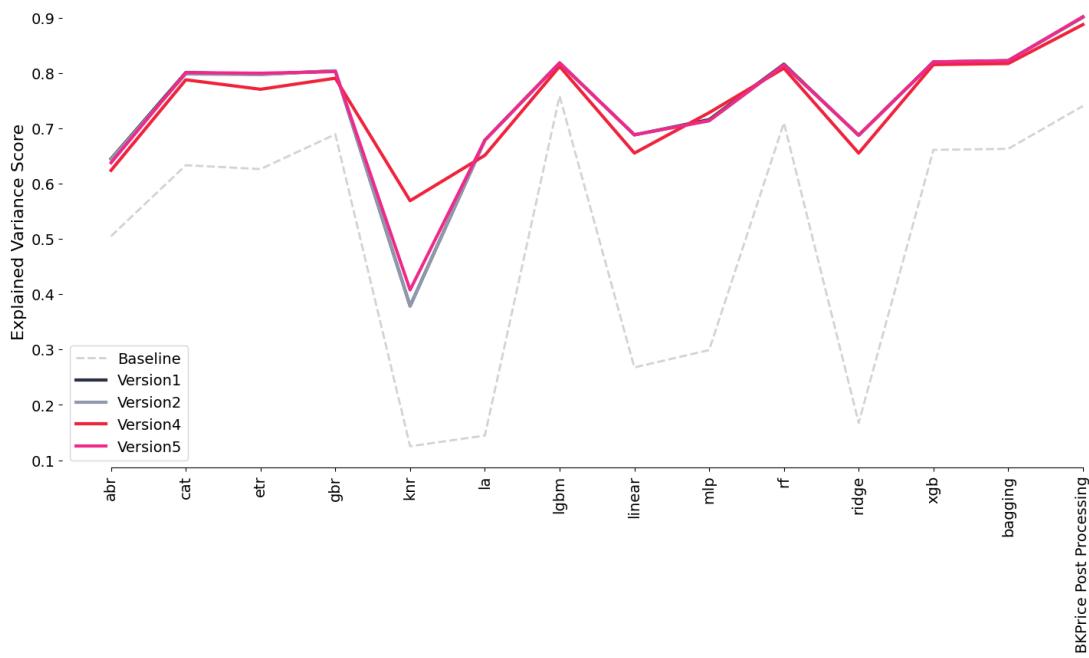
Hình 8.12: HN Data - Explained Variance Score



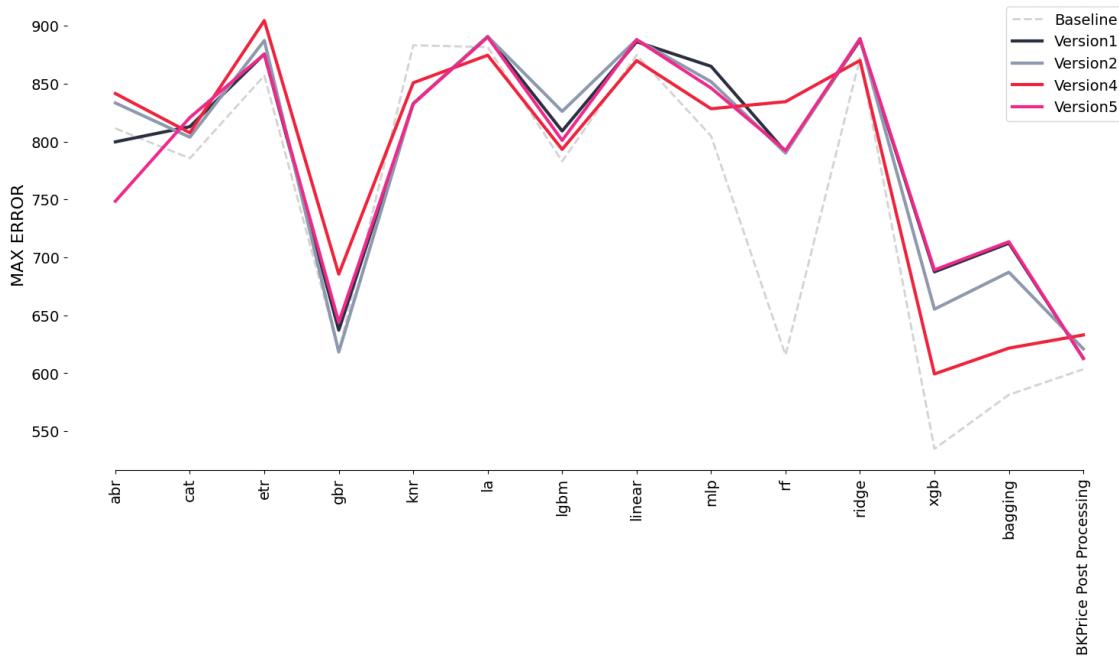
Hình 8.13: HN Data - Max Error



Hình 8.14: HCM Data - RMSE



Hình 8.15: HCM Data - Explained Variance Score



Hình 8.16: HCM Data - Max Error

Hình vẽ 8.11, 8.12, 8.13, 8.14, 8.15, 8.16 mô tả thông tin so sánh tập mô hình cùng với tập thuộc tính để xuất đối với dữ liệu bất động sản ở hai thành phố Hà Nội và Hồ Chí Minh

REFERENCE

- [1] “Tin tức thị trường bất động sản đầu năm 2024,” **url:** <https://batdongsan.com/tin-tuc/thi-truong-bat-dong-san-dau-nam-2024-806302>.
- [2] “Biggee - định giá nhà đất,” **url:** <https://biggee.vn/them-nha-dat>.
- [3] “Onehousing - công cụ định giá bất động sản onehousing,” **url:** <https://onehousing.vn/cong-cu/dinh-gia>.
- [4] “ensemble learning và các biến thể,” **url:** <https://viblo.asia/p/ensemble-learning-va-cac-bien-the-p1-WAyK80AkKxX>.
- [5] “What is principal component analysis (pca) and how it is used,” **url:** <https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186>.
- [6] “Pca explained variance concepts with python example,” **url:** <https://vitalflux.com/pca-explained-variance-concept-python-example/>.
- [7] “gaussian mixture model explained,” **url:** <https://builtin.com/articles/gaussian-mixture-model>.
- [8] “Quadtree algorithm in game,” **url:** <https://www.gradio.app/docs/python-client/introduction>.
- [9] “Creating a feature store with feast,” **url:** <https://kedion.medium.com/creating-a-feature-store-with-feast-part-1-37c380223e2f>.
- [10] “How to setup mlflow on ubuntu,” **url:** <https://airflow.apache.org/docs>.
- [11] “Airflow documentation,” **url:** <https://airflow.apache.org/docs>.
- [12] “Architechture, understand decoupled drupal from a to z,” **url:** <https://kyanon.digital/understand-decoupled-drupal-from-a-to-z/>.
- [13] “Kafka documentation,” **url:** <https://kafka.apache.org/documentation/>.
- [14] “Debezium documentation,” **url:** <https://debezium.io/documentation/reference/stable/index.html>.

- [15] “Stream your database changes with change data capture,” **url:** <https://medium.com/meroxa/stream-your-database-changes-with-change-data-capture-part-two-8fa4daee6afc>.
- [16] “House price prediction using machine learning algorithm - the caseof karachi city, pakistan,” **url:** https://www.researchgate.net/publication/348220705_House_Price_Prediction_using_Machine_Learning_Algorithm_-_The_Case_of_Karachi_City_Pakistan.
- [17] “Housing price prediction via improved machine learning techniques,” **url:** <https://www.sciencedirect.com/science/article/pii/S1877050920316318>.
- [18] “Housing price prediction via improved machine learning techniques,” **url:** https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/1822-b.e-ece-batchno-120.pdf.
- [19] “Housing price prediction using machine learning,” **url:** <https://www.irejournals.com/formatedpaper/1702692.pdf>.
- [20] “Housing price prediction using machine learning in python,” **url:** <https://www.geeksforgeeks.org/house-price-prediction-using-machine-learning-in-python>.
- [21] “Optimizing ensemble weights for machine learning models: A case study for housing price prediction,” **url:** https://link.springer.com/chapter/10.1007/978-3-030-30967-1_9.
- [22] “Gmm, gaussian mixture model,” **url:** <https://scikit-learn.org/stable/modules/mixture.html>.
- [23] “yêu tố nào ảnh hưởng tới giá bất động sản?,” **url:** ‘<https://cafeland.vn/kien-thuc/yeu-to-nao-anh-huong-toi-gia-bat-dong-san-124210.html>’.
- [24] “24 yếu tố ảnh hưởng đến giá trị bất động sản,” **url:** ‘<https://diaocxanhtoancau.com/tin-tuc/24-yeu-to-anh-huong-den-gia-tri-bat-dong-san.html>’.
- [25] “Demo case 1,” **url:** <https://batdongsan.com.vn/ban-nha-rieng-duong-tran-duy-hung-phuong-trung-hoa-4/ban-2-mat-ngo-pr40213606>.
- [26] “Demo case 2,” **url:** <https://batdongsan.com.vn/ban-nha-rieng-pho-hang-bai-phuong-hang-bai/can-ban-gap-4-tang-dan-xay-con-moi-hoan-kiem-dt-36m-mt-hon-7-1m-gia-cuc-tot-pr40257892>.

- [27] “Demo case 3,” **url**: <https://batdongsan.com.vn/ban-nha-rieng-duong-nguyen-trong-tuyen-phuong-1-23/ban-gap-hem-oto-gia-chi-5-ty790tr-rong-58m2-hem-343-xx-p1-tan-binh-pr40268401>.
- [28] “Demo case 4,” **url**: <https://batdongsan.com.vn/ban-nha-rieng-duong-giai-phong-phuong-phuong-mai/chinh-chu-can-ban-gap-can-tai-gia-dinh-chuyen-noi-o-dep-trung-tam-pr40251205>.
- [29] “Demo case 5,” **url**: <https://batdongsan.com.vn/ban-nha-mat-pho-duong-nguyen-thai-hoc-phuong-cau-ong-lanh/-hot-ban-khach-san-tien-quan1-ham-6-lau-doanh-thu-550tr-th-kinh-doanh-dinh-pr40247965>.

Appendices