ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - TIN HỌC

Nguyễn Đình Hưng
Nguyễn Lê Diệu Huyền
Phạm Thiên Phụng

ỨNG DỤNG HỌC SÂU CHO HỆ THỐNG BẤT ĐỘNG SẢN THÔNG MINH

ĐỒ ÁN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Thành phố Hồ Chí Minh 2024

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - TIN HỌC

Nguyễn Đình Hưng
Nguyễn Lê Diệu Huyền
Phạm Thiên Phụng

ỨNG DỤNG HỌC SÂU CHO HỆ THỐNG BẤT ĐỘNG SẢN THÔNG MINH

ĐỒ ÁN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

NGÀNH KHOA HỌC DỮ LIỆU
Mã số sinh viên: 20280041 - 20280047 - 20280075

NGƯỜI HƯỚNG DẪN:
TS. ĐOÀN THỊ TRÂM

Thành phố Hồ Chí Minh 2024

Lời cảm ơn

Lời đầu tiên, nhóm chúng tôi xin gửi lời cảm ơn chân thành đến **Trường Đại học Khoa học Tự nhiên - Đại học Quốc gia Thành phố Hồ Chí Minh** và **Ban chủ nhiệm Khoa Toán - Tin học** đã tạo điều kiện cho chúng tôi được học tập và nghiên cứu tại trường để từ đó chúng tôi có thể hoàn thành chương trình học nói chung và đồ án tốt nghiệp nói riêng.

Lời cảm ơn tiếp theo, nhóm chung em muốn gửi đến **ThS. Đoàn Thị Trâm**, cô là người đã hướng dẫn và góp ý cho nhóm chúng tôi phát triển những ý tưởng ban đầu, định hướng đề tài cho nhóm để có thể đi đến từng những sản phẩm được thử nghiệm và kết luận. Những lời động viên của cô đã tạo nguồn động lực thúc đẩy chúng tôi từng bước đi qua những khó khăn mà nhóm chúng tôi đã gặp phải trong quá trình làm đồ án tốt nghiệp.

Lời cảm ơn thứ ba nhóm chúng tôi xin được gửi đến các thầy cô tại Trường Đại học Khoa học Tự nhiên - Đại học Quốc gia Thành phố Hồ Chí Minh. Được học tập, nghiên cứu cùng các thầy cô, và trải nghiệm những hoạt động thú vị tại khoa Toán - Tin học đã giúp chúng tôi xây dựng nên nền tảng kiến thức vững chắc trong học tập. chúng tôi rất biết ơn sự dạy dỗ từ các thầy cô, chúng tôi đã học được rất nhiều điều từ các thầy cô để không chỉ hoàn thiện kiến thức mà còn là hoàn thiện bản thân mình hơn. Ngoài ra để giúp chúng tôi có những định hướng trong tương lai, khoa cũng đã tạo ra rất nhiều sự kiện để chúng tôi học hỏi và được tư vấn định hướng trong những bước đầu trên con đường sự nghiệp của mình.

Lời cuối cùng, nhóm chúng tôi được gửi lời cảm ơn đến gia đình, đến những người thân và bạn bè đã ở bên cạnh động viên và giúp đỡ nhóm chúng tôi để có thể hoàn thành đồ án tốt nghiệp.

Mục Lục

Danh sách hình vẽ

Danh sách bảng

Danh sách các từ viết tắt

- **AI:** Artificial Intelligence
- **API:** Application Programming Interface
- **BART:** Bidirectional and Auto-Regressive Transformers
- **BERT:** Bidirectional Encoder Representations from Transformers
- **BM25:** Best Matching 25
- **CPU:** Central Processing Unit
- **CSS:** Cascading Style Sheets
- **DL:** Deep Learning
- **DPR:** Dense Passage Retriever
- **GPT:** Generative Pre-training Transformer
- **GPU:** Graphics Processing Unit
- **HTML:** Hyper Text Markup Language
- **INR:** Interquartile Range
- **KNN:** K-Nearest Neighbor Regression
- **LLM:** Large Language Model
- **MAE:** Mean Absolute Error
- **ML:** Machine learning
- **MSE:** Mean Squared Error
- **NLP:** Natural Language Processing
- R^2**:** R-squared
- **RAG:** Retrieval Augmented Generation

- **RMSE:** Root Mean Squared Error
- **T5:** Text-to-Text Transfer Transformer
- **TF-IDF:** Term Frequency-Inverse Document Frequency
- **VRAM:** Video Random Access Memory

Danh sách các thuật ngữ

- **Activation Function:** Hàm toán học được sử dụng trong mạng nơ-ron để giới thiệu tính phi tuyến, giúp mô hình học được các mẫu phức tạp hơn trong dữ liệu.
- **Artificial Intelligence:** Là trí thông minh được thể hiện bằng máy móc, trái ngược với trí thông minh tự nhiên của con người.
- **BART:** Mô hình ngôn ngữ của Facebook, kết hợp các kỹ thuật tiên đoán chiều thuận và chiều ngược để cải thiện hiệu suất trên các nhiệm vụ như dịch ngôn ngữ và tóm tắt văn bản.
- **BERT:** Mô hình ngôn ngữ của Google, sử dụng phương pháp học có giám sát để hiểu ngữ cảnh của từ trong câu bằng cách xem xét cả chiều thuận và chiều ngược.
- **Bias-Corrected Estimates:** Các ước tính đã được điều chỉnh để loại bỏ sự thiên lệch, thường được sử dụng trong các thuật toán tối ưu hóa như Adam để cải thiện độ chính xác của các ước tính gradient và momentum.
- **BM25:** Là một hàm xếp hạng cải tiến từ TF-IDF, sử dụng thêm các tham số điều chỉnh để cân bằng ảnh hưởng của độ dài tài liệu và tần suất từ trong hệ thống truy xuất thông tin.
- **Bootstrap Aggregating (Bagging)**: Là kỹ thuật tạo ra nhiều tập dữ liệu con từ tập dữ liệu gốc bằng cách lấy mẫu ngẫu nhiên có thay thế.
- **Boosting:** Tạo ra một chuỗi các mô hình, trong đó mỗi mô hình mới được huấn luyện để sửa chữa các lỗi của các mô hình trước đó. Kết hợp các mô hình bằng cách trọng số hóa chúng dựa trên hiệu suất.
- **Chatbot:** Chương trình máy tính mô phỏng cuộc trò chuyện với con người qua các nền tảng tin nhắn, thường được sử dụng để cung cấp dịch vụ khách hàng tự động và tương tác người dùng.

- **Dense Passage Retrieval:** Kỹ thuật tìm kiếm thông tin sử dụng mô hình ngôn ngữ để mã hóa các đoạn văn bản vào không gian vector dày đặc, cho phép tìm kiếm nhanh và chính xác.
- **Dying Neurons:** Hiện tượng xảy ra trong mạng nơ-ron khi một số nơ-ron không còn kích hoạt (output luôn bằng 0) trong quá trình huấn luyện, dẫn đến mất khả năng học và giảm hiệu quả của mô hình.
- **Epoch:** Một chu kỳ hoàn chỉnh qua toàn bộ tập dữ liệu huấn luyện. Trong một epoch, mô hình học từ toàn bộ dữ liệu một lần, và nhiều epoch thường cần thiết để mô hình học tốt hơn.
- **Gradient Boosting:** là phương pháp học máy xây dựng một mô hình mạnh bằng cách lặp lại huấn luyện các mô hình yếu và kết hợp chúng để giảm thiểu sai số dự đoán.
- **Gradient Descent:** Thuật toán tối ưu hoá được dùng để tìm điểm tối ưu của hàm số.
- **Hyperparameters:** Các tham số được thiết lập trước khi huấn luyện mô hình, như tốc độ học, số lượng lớp, số lượng nơ-ron trong mỗi lớp, giúp điều chỉnh cách thức mô hình học và tổng quát hóa dữ liệu.
- **Learning-rate:** Learning-rate hay còn gọi là tốc độ học là một tham số trong các thuật toán học máy và tối ưu, xác định mức độ cập nhật của các trọng số mạng neural hay các tham số của mô hình dựa trên độ lỗi hiện tại và độ dốc của hàm mất mát.
- **Loss Function:** Loss function hay còn gọi là hàm mất mất là một hàm toán học đo lường sự khác biệt giữa giá trị dự đoán của mô hình và giá trị thực tế, giúp điều chỉnh trọng số của mô hình trong quá trình huấn luyện .
- **Mean Squared Error:** Phép đo thường dùng để đánh giá mức độ chính xác của mô hình trong việc dự đoán và tái tạo dữ liệu. MSE là phép đo lỗi bình phương trung bình giữa giá trị dự đoán và giá trị thực tế.
- **Mean Absolute Error:** Phép đo thường dùng để đánh giá mức độ chính xác của mô hình trong việc dự đoán và tái tạo dữ liệu. MAE là phép đo lỗi trung bình tuyệt đối giữa giá trị dự đoán và giá trị thực tế.
- **Momentum:** Là một kỹ thuật tối ưu hóa trong học máy, giúp tăng tốc độ hội tụ của thuật toán bằng cách thêm một phần của gradient trước đó vào gradient hiện tại, giảm thiểu dao động và tăng tốc độ học.

- **Natural Language Processing:** Lĩnh vực nghiên cứu và ứng dụng của trí tuệ nhân tạo tập trung vào việc xử lý và hiểu ngôn ngữ tự nhiên của con người.
- **Optimizer:** Thuật toán hoặc phương pháp dùng để điều chỉnh các tham số của mô hình học máy nhằm tối thiểu hóa loss function .
- **Outliers:** Các giá trị dữ liệu có sự khác biệt lớn so với phần còn lại của tập dữ liệu, có thể gây ảnh hưởng xấu đến kết quả của phân tích thống kê hoặc học máy .
- **Overfitting:** Là tình trạng xảy ra khi một mô hình học máy học quá chi tiết từ dữ liệu huấn luyện, bao gồm cả nhiễu và các chi tiết không liên quan.
- **Regularization:** Là kỹ thuật được sử dụng để giảm thiểu overfitting bằng cách thêm một ràng buộc vào hàm mục tiêu trong quá trình huấn luyện mô hình.
- **Reparameterization trick:** Kỹ thuật nhằm giúp xấp xỉ đạo hàm của mô hình dễ dàng hơn để có thể áp dụng các phương pháp tối ưu hóa gradient.
- **Scheduler:** Scheduler trong học máy là công cụ quản lý tốc độ học (learning rate) của mô hình trong quá trình huấn luyện, thường sử dụng để tối ưu hóa hiệu suất của mô hình .
- **Sequence:** Một chuỗi các đơn vị (như từ hoặc ký tự) trong một ngữ cảnh cụ thể, thường được xử lý trong mô hình ngôn ngữ.
- **Text-to-Text Transfer Transformer:** Mô hình ngôn ngữ của Google, sử dụng phương pháp tiếp cận biến mọi nhiệm vụ NLP thành một vấn đề chuyển đổi văn bản thành văn bản.
- **TF-IDF:** Là một kỹ thuật đánh giá mức độ quan trọng của từ trong tài liệu dựa trên tần suất xuất hiện và tính phổ biến của từ đó trong toàn bộ tập tài liệu
- **Token:** Đơn vị cơ bản trong văn bản mà mô hình ngôn ngữ xử lý, có thể là từ, ký tự, hoặc phần của từ.
- **Vanishing Gradients:** Vanishing gradients là vấn đề trong học sâu, khi gradient trở nên rất nhỏ, làm chậm hoặc ngăn chặn quá trình cập nhật trọng số của các lớp đầu tiên trong mạng nơ-ron sâu .

Lời nói đầu

Hiện nay, lĩnh vực bất động sản đang phát triển mạnh mẽ với nhiều dự án có quy mô khác nhau. Người mua cần nguồn tin đáng tin cậy để định giá bất động sản, từ đó đưa ra quyết định hợp lý. Người bán và nhà đầu tư cần định giá hợp lý để phản ánh đúng giá thị trường, tạo niềm tin và thu hút khách hàng. Ngoài ra, việc tư vấn bất động sản thường tốn kém và không đảm bảo tư vấn mọi lúc mọi nơi với thông tin cập nhật thường xuyên.

Đồ án tốt nghiệp của nhóm chúng tôi nghiên cứu ứng dụng máy học vào bất động sản nhằm tạo ra hệ thống thông minh dự đoán giá nhà và tư vấn thông tin cho người mua, người bán và nhà đầu tư. Chúng tôi hy vọng kết quả nghiên cứu này sẽ mang lại ý tưởng mới trong việc ứng dụng máy học vào bất động sản và các lĩnh vực khác, góp phần tự động hóa công việc và cải thiện trải nghiệm người dùng.

Nội dung báo cáo đồ án tốt nghiệp bao gồm 6 chương:

1. **Giới thiệu:** Lí do chọn đề tài và trình bày bài toán.
2. **Tổng quan lý thuyết:** Các khái niệm chung về lĩnh vực bất động sản, các hướng tiếp cận bài toán và các mô hình máy học được ứng dụng trong bài.
3. **Phương pháp thực hiện:** Bao gồm các phương pháp được thực hiện trong quá trình thu thập dữ liệu, tiền xử lý dữ liệu, và các mô hình máy học được ứng dụng trong bài.
4. **Thực nghiệm và kết quả:** Chi tiết quá trình cài đặt mô hình, các phương pháp đánh giá và kết quả thực nghiệm.
5. **Ứng dụng:** Ứng dụng phương pháp máy học vào lĩnh vực bất động sản.
6. **Kết luận và mở rộng:** Kết luận và trình bày định hướng trong tương lai của bài toán.

Chương 1

Giới thiệu

1.1 Lí do chọn đề tài

Giá bất động sản là một chỉ số quan trọng phản ánh tình hình kinh tế của một quốc gia, ảnh hưởng đến sự ổn định tài chính, tiêu dùng và chính sách tiền tệ của chính phủ. Thị trường bất động sản luôn biến động, việc dự đoán giá bất động sản giúp các nhà đầu tư, người mua và người bán đưa ra các quyết định hợp lý và hiệu quả hơn, giảm thiểu rủi ro và tăng cường khả năng sinh lợi từ các khoản đầu tư.

Đặc biệt với sự phát triển của ML và DL đã mở ra nhiều cơ hội mới trong việc phân tích và dự đoán giá bất động sản. Mặc dù có nhiều nghiên cứu về dự đoán giá bất động sản trên thế giới,nhưng lĩnh vực này ở Việt Nam và nhiều nước đang phát triển vẫn còn nhiều tiềm năng chưa được khai thác. Việc tập trung vào một thành phố cụ thể như Thành phố Hồ Chí Minh giúp tạo ra một mô hình dự đoán cụ thể và ứng dụng thực tế, từ đó có thể mở rộng ra các khu vực khác.

Một mô hình dự đoán giá bất động sản chính xác không chỉ giúp các nhà đầu tư và người mua bán bất động sản, mà còn hỗ trợ các nhà hoạch định chính sách trong việc điều chỉnh và quản lý thị trường, đồng thời giúp các ngân hàng và tổ chức tài chính đánh giá rủi ro tín dụng hiệu quả hơn.

Bên cạnh đó việc tăng trải nghiệm cho người dùng như tư vấn về các vấn đề mà những nhà đầu tư hay người mua, người bán gặp phải trong lĩnh vực vất động sản cũng đang là một vấn đề mà có thể nghiên cứu hơn. Việc nhân lực là con người không thể giúp tư vấn được mọi lúc mọi nơi cũng như không thể tư vấn đầy đủ tất cả các kiến thức trong lĩnh vực bất động sản. Nắm được nỗi đau này, nhóm chúng tôi đã tiến hành nghiên cứu và phát triển một chatbot có khả năng học tập và trả lời những câu hỏi, cũng như tư vấn cho

người dùng là những nhà đầu tư hay người mua, người bán ở lĩnh vực bất động sản về những kiến thức thuộc lĩnh vực này. Xa hơn nữa, có thể phát triển và mở rộng sang những lĩnh vực khác, nhằm tự động hóa những công việc cũng như giảm bớt sức người, tiền bạc.

Do đó chúng tôi tập trung nghiên cứu vào đề tài **"Ứng dụng học sâu cho hệ thống bất động sản thông minh"**. Đề tài này không chỉ mang lại cơ hội nghiên cứu sâu hơn trong lĩnh vực học máy, mà còn mở ra các hướng nghiên cứu mới về cách áp dụng công nghệ vào giải quyết các vấn đề kinh tế - xã hội.

1.2 Trình bài bày toán

Trong chủ đề này, chúng tôi tập trung nghiên cứu những phương pháp máy học tiên tiến để xây dựng một hệ thống trong lĩnh vực bất động ở khu vực Thành phố Hồ Chí Minh. Mục tiêu của chúng tôi là không chỉ đưa ra những dự đoán chính xác về giá trị bất động sản, mà còn cải tiến và áp dụng công nghệ để xây dựng một hệ thống thông minh có khả năng tư vấn những dịch vụ liên quan trong lĩnh vực bất động sản trong nước.

Cụ thể, chúng tôi sử dụng các phương pháp ML và DL để phân tích dữ liệu và đưa ra dự đoán về giá nhà. Những phương pháp này giúp chúng tôi hiểu rõ hơn về các yếu tố ảnh hưởng đến giá bất động sản, từ đó tạo ra các mô hình dự đoán với độ chính xác cao.

Bên cạnh việc dự đoán giá nhà, chúng tôi cũng xây dựng hệ thống tư vấn thông minh để tư vấn các dịch vụ liên quan như lựa chọn khu vực tiềm năng, đánh giá tình trạng thị trường, và đề xuất các chiến lược đầu tư hợp lý.

Dưới đây là những đóng góp của những nghiên cứu của nhóm chúng tôi về mặt học thuật cũng như về mặt ứng dụng:

- Đầu tiên, chúng tôi cung cấp một vài ý tưởng về phương pháp nghiên cứu trong việc dự đoán giá nhà.

- Thứ hai, chúng tôi đề xuất một vài phương pháp trong việc nghiên cứu về chatbot trong lĩnh vực bất động sản, góp phần tạo nên sự đa dạng cho các phương pháp trong lĩnh vực về nghiên cứu chatbot.

- Cuối cùng, ứng dụng hệ thống này bất động sản thông minh sẽ là một công cụ hữu ích cho cả người mua, người bán và các nhà đầu tư bất động sản và từ hệ thống này sẽ phát triển hơn trong nhiều lĩnh vực khác không chỉ riêng lĩnh vực bất động sản, qua đó góp phần vào việc tự động hóa

các công việc với độ chính xác cao, mang lại nhiều trải nghiệm tiện lợi cho người dùng, mọi lúc, mọi nơi.

Bộ dữ liệu này của chúng tôi có các thông tin về bất động sản ở khu vực Thành phố Hồ Chí Minh và được chúng tôi thu thập từ hai trang web uy tín là nhatot.com [1] và batdongsan.com.vn [2]. Đây là hai trang web chuyên cung cấp và hỗ trợ người dùng đăng thông tin về bất động sản tại khu vực này.

[1] https://www.nhatot.com/

[2] https://batdongsan.com.vn/

Chương 2

Tổng quan lý thuyết

2.1 Các khái niệm chung về lĩnh vực bất động sản

Bất động sản hay còn gọi là địa ốc hay nhà đất là một thuật ngữ pháp luật có ý nghĩa bao gồm đất đai và những gì dính liền vĩnh viễn với mảnh đất. Những thứ được xem là dính liền vĩnh viễn như là nhà cửa, ga ra, kiến trúc ở trên hoặc dầu khí, mỏ khoáng chất ở dưới mảnh đất đó. Những thứ có thể dỡ ra khỏi mảnh đất như nhà di động,lều, nhà tạm thì không được xem là bất động sản. [1]

Bất động sản nhà ở là bất động sản được sử dụng để sinh sống hoặc cho thuê. Nó có thể là một ngôi nhà đơn hộ hoặc một tòa nhà chung cư có nhiều hộ gia đình. Nhà ở có thể được phân loại theo cách chúng được kết nối với nhà ở lân cận và đất đai. Các loại hình sở hữu nhà khác nhau có thể được sử dụng cho cùng một loại hình vật lý. Ví dụ, nhà ở kết nối có thể thuộc sở hữu của một thực thể duy nhất và được cho thuê, hoặc thuộc sở hữu riêng biệt với thỏa thuận bao gồm mối quan hệ giữa các đơn vị và khu vực chung và các mối quan tâm.[2]

2.2 Các hướng tiếp cận bài toán

2.2.1 Mô hình dự đoán giá nhà

Dựa trên nghiên cứu của chúng tôi về các phương pháp nghiên cứu liên quan, phương pháp thu thập và xử lý dữ liệu bất động sản đã được áp dụng rộng rãi. Tuy nhiên, điểm yếu của các phương pháp hiện tại là không thể

[1] https://vi.wikipedia.org/wiki/Batdongsan

[2] https://vi.wikipedia.org/wiki/Batdongsannhao

cung cấp một cái nhìn chi tiết về các yếu tố ảnh hưởng đến giá trị của từng bất động sản trong các điều kiện thị trường khác nhau. Do đó, trong nghiên cứu này, chúng tôi tập trung vào việc nghiên cứu, thu thập dữ liệu và phân tích giá bất động sản tại thị trường Việt Nam, tập trung vào thị trường TP. Hồ Chí Minh, trong khoảng thời gian quý 4, năm 2023.

Có hai phương pháp dùng trong nghiên cứu này: phương pháp áp dụng Machine Learning và phương pháp áp dụng Deep Learning.

1. **Phương pháp áp dụng Machine Learning**

   Dataset -> **Preprocessing** -> **Machine learning**: Quá trình này bắt đầu từ việc thu thập dữ liệu bất động sản, tiến hành tiền xử lý và trích xuất đặc trưng nhằm đảm bảo rằng các thông tin thu được được sắp xếp, chuẩn hóa và loại bỏ nhiễu một cách chuyên nghiệp, dựa trên sự hiểu biết sâu rộng về lĩnh vực này. Trong phương pháp này, vai trò của người thực hiện rất quan trọng vì từ khâu tiền xử lý và kỹ thuật trích xuất đặc trưng, chất lượng và độ chính xác của dữ liệu trực tiếp ảnh hưởng đến kết quả cuối cùng của mô hình Machine Learning.

   Mục đích chính của việc áp dụng các mô hình Machine Learning là để dự đoán giá trị của bất động sản dựa trên các đặc tính và thuộc tính của chúng. Các mô hình này được lựa chọn và điều chỉnh sao cho phù hợp nhất với bài toán hồi quy, nơi mà chúng có khả năng xử lý và phân tích dữ liệu để đưa ra những dự đoán chính xác và tin cậy.[3]

2. **Phương pháp áp dụng Deep Learning**

   Dataset -> **Preprocessing** -> **Deep learning**: Quá trình áp dụng phương pháp áp dụng Deep Learning cũng tương tự với phương pháp áp dụng Machine Learning. Trong phương pháp này những khâu tiền xử lý dữ liệu và kỹ thuật trích xuất đặc trưng cũng ảnh hưởng đến kết quả cuối cùng của mô hình Deep Learning.

   Mục đích chính của phương pháp áp dụng Deep Learning cũng dùng để dự đoán giá trị của bất động sản dựa trên các đặc tính và thuộc tính của chúng. Và các mô hình này được thiết kế và huấn luyện để tự động học hỏi từ dữ liệu, nhận diện các mẫu phức tạp và đặc trưng không hiển nhiên, giúp cải thiện độ chính xác và tin cậy của dự đoán.[4]

[3]https://viblo.asia/p/machine-learning

[4]`https://aws.amazon.com/vi/what-is/deep-learning/`

2.2.2 Mô hình chatbot

Dựa trên những nghiên cứu của chúng tôi, hiện nay có rất nhiều mô hình chatbot ra đời, chatbot như ChatGPT của OpenAI hay Gemini của Google có thể trả lời những câu hỏi của người dùng và cung cấp cho người dùng nhiều thông tin bổ ích, nhưng chatbot hiện nay khó có thể đi sâu và cập nhật những xu thế của một lĩnh vực cụ thể. Nắm bắt được điều này, chúng tôi xây dựng một mô hình chatbot để có thể tư vấn sâu vào lĩnh vực bất động sản, qua đó có thể đem lại nhiều thông tin cũng như là những trải nghiệm mới mẻ, hiệu quả cho người dùng, giúp tiết kiệm được thời gian, cũng như là tiền bạc và nhân lực.

Dataset -> Data Processing -> Build RAG Model: Quá trình này cũng bắt đầu từ việc thu thập dữ liệu đến xử lý các dữ liệu để chuẩn bị cho việc đưa vào mô hình RAG và cuối cùng là xây dựng một mô hình RAG cụ thể để áp dụng vào bài toán tư vấn bất động sản.

Mục đích chính của việc xây dựng chatbot tư vấn bất động sản bằng RAG là kết hợp khả năng truy xuất thông tin chính xác từ cơ sở dữ liệu với khả năng tạo ra các câu trả lời tự nhiên và phù hợp ngữ cảnh. Điều này giúp chatbot cung cấp thông tin nhanh chóng, chính xác và thân thiện, nâng cao trải nghiệm khách hàng, đồng thời tối ưu hóa hiệu quả và chi phí hoạt động của doanh nghiệp.

2.3 Các mô hình

2.3.1 Mô hình dự đoán giá nhà

Hiệu quả và ưu điểm của phương pháp áp dụng ML và DL cho phép chúng tôi giải quyết bài toán.

Sau đây, chúng tôi sẽ trình bày phần lý thuyết của các mô hình ML và DL được chúng tôi tìm hiểu và sử dụng trong nghiên cứu này.

1. **Mô hình Machine Learning**

   (a) **Linear Regression**

   Linear Regression là một phương pháp hồi quy tuyến tính giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Mô hình này giả định rằng mối quan hệ giữa các biến là tuyến tính, nghĩa là biến phụ thuộc có thể được biểu diễn dưới dạng một tổ hợp tuyến tính của các biến độc lập.[1]

Phương trình mô hình Linear Regression có dạng sau:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

- Y là biến phụ thuộc,
- $X_1, X_2, \ldots, X_p$ là các biến độc lập,
- $\beta_0, \beta_1, \ldots, \beta_p$ là các hệ số hồi quy,
- ϵ là sai số ngẫu nhiên.

Mô hình Linear Regression sử dụng một bộ mã hóa để ánh xạ các đặc trưng của bất động sản. Mô hình cố gắng điều chỉnh phân phối của các giá trị này để gần với một phân phối chuẩn liên tục. Sau đó, thành phần còn lại của mô hình Linear Regression là bộ giải mã, cố gắng dự đoán giá của bất động sản.[3]

Mục tiêu của mô hình là tối thiểu hóa hàm mất mát:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

trong đó y_i là giá trị thực tế của giá bất động sản, $\hat{y}_i$ là giá trị dự đoán, và n là số lượng mẫu huấn luyện.

Hàm mục tiêu của mô hình Linear Regression tập trung vào việc tìm ra biểu diễn tiềm ẩn chứa các đặc trưng quan trọng của dữ liệu bất động sản. Kết quả này giúp mô hình dự đoán giá của các bất động sản một cách chính xác và hiệu quả.

Ưu điểm:

- Đơn giản và dễ hiểu: Phù hợp cho người mới bắt đầu trong thống kê và khoa học dữ liệu.
- Linh hoạt: Có thể áp dụng để mô hình hóa mối quan hệ tuyến tính giữa các biến.
- Thời gian tính toán nhanh: Phù hợp cho các bài toán có số lượng quan sát lớn.
- Dễ điều chỉnh: Các tham số của mô hình (như hệ số hồi quy) có thể được điều chỉnh dễ dàng để cải thiện hiệu suất.

Nhược điểm:

- Nhạy cảm với các giá trị ngoại lai: Các giá trị ngoại lai có thể ảnh hưởng lớn đến độ chính xác của mô hình.
- Không thể xử lý các mối quan hệ phức tạp: Linear Regression giới hạn trong việc mô hình hóa các mối quan hệ phi tuyến tính.

- Yêu cầu các biến độc lập phải độc lập tuyến tính: Điều này có thể là hạn chế đối với các bài toán phức tạp có sự phụ thuộc phức tạp giữa các biến.

(b) **Random Forest Regression**

Decision Tree là một phương pháp hồi quy phi tuyến tính được sử dụng rộng rãi để dự đoán các giá trị liên tục. Cây quyết định hoạt động bằng cách chia dữ liệu thành các tập con dựa trên các đặc điểm của dữ liệu, theo cách thức phân tách tối ưu để giảm thiểu sai số dự đoán.[4]

Cấu trúc của mô hình cây quyết định bao gồm các nodes và branches. Mỗi nút đại diện cho một điều kiện phân tách dựa trên một đặc điểm cụ thể của dữ liệu, và các nhánh xuất phát từ nút đó đại diện cho các tập con dữ liệu thỏa mãn điều kiện phân tách.

Phương trình mô hình Decision Tree cho hồi quy có thể được biểu diễn như sau:

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Trong đó:

- n là số lượng mẫu trong nút lá (leaf node).
- y_i là giá trị quan sát của biến phụ thuộc cho mẫu thứ i.

Bên cạnh đó mô hình **Random Forest** là một tập hợp các cây quyết định, hoạt động bằng cách kết hợp các dự đoán từ nhiều cây quyết định khác nhau để đưa ra dự đoán cuối cùng. Phương pháp này giúp cải thiện tính tổng quát và độ chính xác của mô hình so với việc sử dụng một cây quyết định đơn lẻ.

Mô hình Random Forest Regression sử dụng một tập hợp các cây quyết định để ánh xạ các đặc trưng của dữ liệu. Mô hình cố gắng kết hợp các dự đoán từ nhiều cây quyết định khác nhau để đưa ra dự đoán cuối cùng. Mỗi cây quyết định được huấn luyện trên một tập con ngẫu nhiên của dữ liệu và các đặc trưng để giảm thiểu sai số dự đoán và tránh overfitting.[4]

Mục tiêu của mô hình là tối thiểu hóa hàm mất mát:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

trong đó y_i là giá trị thực tế của giá trị dự đoán, $\hat{y}_i$ là giá trị dự đoán từ mô hình, và n là số lượng mẫu huấn luyện.

Hàm mục tiêu của mô hình Random Forest Regression tập trung vào việc tối ưu hóa khả năng dự đoán của rừng cây bằng cách tìm ra các đặc trưng quan trọng và kết hợp chúng một cách hiệu quả. Kết quả này giúp mô hình dự đoán các giá trị một cách chính xác và tin cậy hơn.

Ưu điểm:

- Giảm overfitting: Bằng cách kết hợp nhiều cây quyết định, Random Forest giúp giảm nguy cơ overfitting so với việc sử dụng một cây quyết định đơn lẻ.
- Tăng độ chính xác: Random Forest thường có độ chính xác cao hơn so với nhiều mô hình hồi quy khác.
- Khả năng xử lý dữ liệu phi tuyến tính: Random Forest có thể xử lý tốt các mối quan hệ phi tuyến tính giữa các biến.
- Tự động chọn lựa đặc trưng: Random Forest có khả năng đánh giá tầm quan trọng của các biến đầu vào và tự động chọn lựa những đặc trưng quan trọng nhất.

Nhược điểm:

- Tính toán phức tạp: Random Forest đòi hỏi nhiều tài nguyên tính toán, đặc biệt khi số lượng cây quyết định lớn.
- Giải thích mô hình: Mô hình Random Forest thường khó giải thích hơn so với các mô hình hồi quy tuyến tính.

(c) **K-Nearest Neighbor Regression**

K-Nearest Neighbor Regression là một phương pháp hồi quy phi tuyến tính dựa trên việc dự đoán giá trị của một điểm dữ liệu bằng cách tính trung bình của giá trị của K điểm gần nhất trong tập huấn luyện.[10]

Cấu trúc của mô hình K-Nearest Neighbor bao gồm các điểm dữ liệu (data points) và một tham số K, đại diện cho số lượng điểm gần nhất được sử dụng để tính dự đoán.

Mô hình K-Nearest Neighbor Regression sử dụng các điểm dữ liệu lân cận để dự đoán giá trị của điểm dữ liệu mới. Mô hình dựa trên giả định rằng các điểm dữ liệu gần nhau trong không gian đặc trưng có giá trị mục tiêu tương tự. KNN Regression hoạt động bằng cách

tìm kiếm K điểm gần nhất trong tập dữ liệu huấn luyện và tính trung bình giá trị mục tiêu của các điểm này để đưa ra dự đoán.[1]

Mục tiêu của mô hình là tối thiểu hóa hàm mất mát:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

trong đó y_i là giá trị thực tế của giá trị cần dự đoán, $\hat{y}_i$ là giá trị dự đoán từ mô hình, và n là số lượng mẫu huấn luyện.

Hàm mục tiêu của mô hình KNN Regression tập trung vào việc tối ưu hóa khả năng dự đoán bằng cách sử dụng thông tin từ các điểm dữ liệu lân cận. Kết quả này giúp mô hình dự đoán các giá trị một cách chính xác và hiệu quả hơn trong các tình huống không có mối quan hệ tuyến tính rõ ràng giữa các biến.

Ưu điểm:

- Đơn giản và dễ triển khai: KNN là một trong những thuật toán đơn giản nhất trong học máy và dễ áp dụng cho nhiều bài toán.
- Không cần giả định về phân phối của dữ liệu: KNN không yêu cầu các giả định về phân phối của dữ liệu, do đó có thể áp dụng cho các dạng dữ liệu đa dạng.
- Hiệu quả với các vấn đề có cấu trúc không rõ ràng và không có mối quan hệ tuyến tính rõ ràng: KNN có thể hoạt động tốt trong các trường hợp dữ liệu có cấu trúc phức tạp hoặc không có mối quan hệ tuyến tính rõ ràng giữa các biến.

Nhược điểm:

- Độ phức tạp tính toán cao khi số lượng điểm dữ liệu lớn: KNN yêu cầu tính toán khoảng cách giữa điểm dữ liệu cần dự đoán và tất cả các điểm dữ liệu trong tập huấn luyện, điều này có thể rất tốn thời gian đối với các tập dữ liệu lớn.
- Đòi hỏi nhiều tài nguyên để lưu trữ toàn bộ dữ liệu huấn luyện: KNN cần lưu trữ toàn bộ tập dữ liệu huấn luyện trong bộ nhớ để thực hiện dự đoán, điều này có thể là một vấn đề với các tập dữ liệu lớn.
- Độ chính xác giảm khi K quá lớn, và dễ bị ảnh hưởng bởi nhiễu: KNN có thể dễ bị ảnh hưởng bởi nhiễu và khi giá trị của K quá lớn, mô hình có thể trở nên quá đơn giản và dễ bị underfitting.

(d) **XGBoost Regression**

XGBoost Regression là một phương pháp hồi quy để mô hình hóa mối quan hệ giữa một biến phụ thuộc và một tập hợp các biến độc lập. Phương pháp này mở rộng từ Gradient Boosting và nhắm vào việc tối ưu hóa hiệu suất và khả năng tự điều chỉnh của mô hình.[5]

Mô hình XGBoost được xây dựng bằng cách kết hợp nhiều cây quyết định yếu thành một mô hình dự đoán mạnh mẽ hơn. Công thức tổng quát của XGBoost Regression có thể được biểu diễn như sau:

$$Y = \phi(X) + \epsilon$$

Trong đó:

- Y là biến phụ thuộc (target variable),
- X là tập hợp các biến độc lập,
- $\phi(X)$ là hàm dự đoán của mô hình XGBoost, được hình thành từ sự kết hợp tuyến tính và phi tuyến tính của các cây quyết định,
- ϵ là sai số ngẫu nhiên.

Mô hình XGBoost Regression sử dụng một tập hợp các cây quyết định để dự đoán giá trị của điểm dữ liệu mới. Mô hình dựa trên kỹ thuật boosting, nơi các cây quyết định được xây dựng tuần tự, mỗi cây mới cố gắng sửa lỗi của cây trước đó. XGBoost sử dụng thuật toán Gradient Boosting, trong đó mỗi cây được huấn luyện để giảm thiểu sai số của mô hình tổng thể.[5]

Mục tiêu của mô hình là tối thiểu hóa hàm mất mát:

$$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

trong đó y_i là giá trị thực tế của giá trị cần dự đoán, $\hat{y}_i$ là giá trị dự đoán từ mô hình, và n là số lượng mẫu huấn luyện.

Hàm mục tiêu của mô hình XGBoost Regression tập trung vào việc tối ưu hóa khả năng dự đoán của toàn bộ mô hình bằng cách sử dụng thông tin từ tất cả các cây quyết định trong quá trình huấn luyện. Kết quả này giúp mô hình dự đoán các giá trị một cách chính xác và hiệu quả hơn trong các tình huống có mối quan hệ phức tạp giữa các biến.

Ưu điểm:

- Hiệu suất cao: XGBoost thường cho kết quả tốt hơn so với nhiều phương pháp khác, đặc biệt là khi xử lý dữ liệu lớn và phức tạp.

- Tính linh hoạt: Có thể áp dụng cho mô hình hóa các mối quan hệ tuyến tính và phi tuyến tính giữa các biến.
- Tính toán hiệu quả: Mô hình XGBoost được tối ưu hóa để đáp ứng yêu cầu tính toán cao và thời gian thực.
- Tính khả điều chỉnh: Các tham số của mô hình (như số lượng cây, độ sâu cây) có thể được điều chỉnh để cải thiện hiệu suất mô hình.

Nhược điểm:

- Nhạy cảm với ngoại lệ: Các giá trị ngoại lai có thể ảnh hưởng đáng kể đến hiệu suất của mô hình XGBoost.
- Yêu cầu dữ liệu tinh chỉnh: Để đạt được hiệu suất tối ưu, mô hình XGBoost yêu cầu một số lượng lớn dữ liệu huấn luyện và tinh chỉnh tham số phù hợp.

2. **Mô hình Deep Learning**

Deep Learning là một phương pháp trong trí tuệ nhân tạo và khoa học dữ liệu, được sử dụng để mô hình hóa các mối quan hệ phức tạp và phi tuyến tính giữa các biến đầu vào và đầu ra. Phương pháp này dựa trên các mạng nơ-ron nhân tạo với nhiều tầng ẩn, cho phép tự động học hỏi và trích xuất các đặc trưng từ dữ liệu.[12]

Cấu trúc cơ bản của một mô hình Deep Learning thường bao gồm:

- **Tầng đầu vào**: Nhận dữ liệu đầu vào, bao gồm một hoặc nhiều biến độc lập.
- **Các tầng ẩn**: Bao gồm nhiều nơ-ron được kết nối với nhau, thực hiện các phép biến đổi phi tuyến tính và trích xuất các đặc trưng từ dữ liệu đầu vào.
- **Tầng đầu ra**: Sản xuất kết quả dự đoán cuối cùng, có thể là một biến phụ thuộc hoặc một tập hợp các giá trị dự đoán.

Quá trình huấn luyện mô hình Deep Learning bao gồm lan truyền tiến, tính toán hàm mất mát và lan truyền ngược để điều chỉnh các trọng số nhằm giảm thiểu sai số và cải thiện độ chính xác của mô hình.

Deep Learning là khả năng tự động trích xuất đặc trưng, khả năng xử lý mối quan hệ phức tạp, và tính linh hoạt cao trong nhiều lĩnh vực như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên (NLP) và dự đoán chuỗi thời gian. Tuy nhiên, Deep Learning cũng có một số hạn chế như yêu cầu lượng dữ liệu lớn, thời gian huấn luyện dài, và khả năng dễ bị quá

khớp nếu không có biện pháp điều chỉnh phù hợp. Mặc dù phức tạp hơn so với các mô hình học máy truyền thống, Deep Learning mang lại hiệu quả cao trong việc xử lý các bài toán có độ phức tạp và khối lượng dữ liệu lớn.[7]

2.3.2 Mô hình chatbot

Giới Thiệu

Retrieval-Augmented Generation [14] (RAG) là một phương pháp trong lĩnh vực xử lý ngôn ngữ tự nhiên, kết hợp giữa khả năng tìm kiếm thông tin (retrieval) và khả năng sinh văn bản (generation). RAG được giới thiệu bởi Facebook AI Research để cải thiện hiệu suất của các hệ thống hỏi đáp và tạo văn bản.

Các Thành Phần Chính của RAG

1. **Bộ tìm kiếm (Retriever):** Bộ tìm kiếm có nhiệm vụ tìm kiếm các tài liệu hoặc đoạn văn bản có liên quan từ một kho dữ liệu lớn dựa trên câu truy vấn đầu vào. Các kỹ thuật tìm kiếm thường được sử dụng bao gồm TF-IDF[18], BM25[17], và các mô hình ngôn ngữ được huấn luyện trước như BERT[6], DPR[11].

2. **Bộ sinh văn bản (Generator):** Bộ sinh văn bản có nhiệm vụ tạo ra câu trả lời hoặc văn bản dựa trên các tài liệu hoặc đoạn văn bản đã được tìm kiếm bởi bộ tìm kiếm. Bộ sinh văn bản thường sử dụng các mô hình ngôn ngữ như BART[13], T5[16] hoặc GPT[15].

3. **Kết hợp (Combination):** RAG kết hợp kết quả của bộ tìm kiếm và bộ sinh văn bản để tạo ra câu trả lời cuối cùng. Quá trình này thường bao gồm hai bước:

   (a) **Bước 1:** Sử dụng bộ tìm kiếm để lấy các tài liệu có liên quan.

   (b) **Bước 2:** Sử dụng bộ sinh văn bản để tạo câu trả lời dựa trên các tài liệu đã tìm kiếm.

Kiến trúc của RAG

RAG có hai biến thể chính: RAG-Sequence và RAG-Token.

1. **RAG-Sequence:** RAG-Sequence tạo ra câu trả lời bằng cách sử dụng bộ sinh văn bản để sinh ra toàn bộ chuỗi văn bản (sequence) dựa trên các tài liệu tìm kiếm.
2. **RAG-Token:** RAG-Token tạo ra câu trả lời bằng cách sinh ra từng token (đơn vị nhỏ nhất của văn bản) một, dựa trên các tài liệu tìm kiếm và các token đã sinh ra trước đó.

Công thức

1. **Hàm xác suất:** Trong RAG, xác suất của câu trả lời y dựa trên câu truy vấn x và các tài liệu z được tìm kiếm được biểu diễn như sau:

$$P(y|x) = \sum_{z \in Z} P(z|x)P(y|x,z)$$

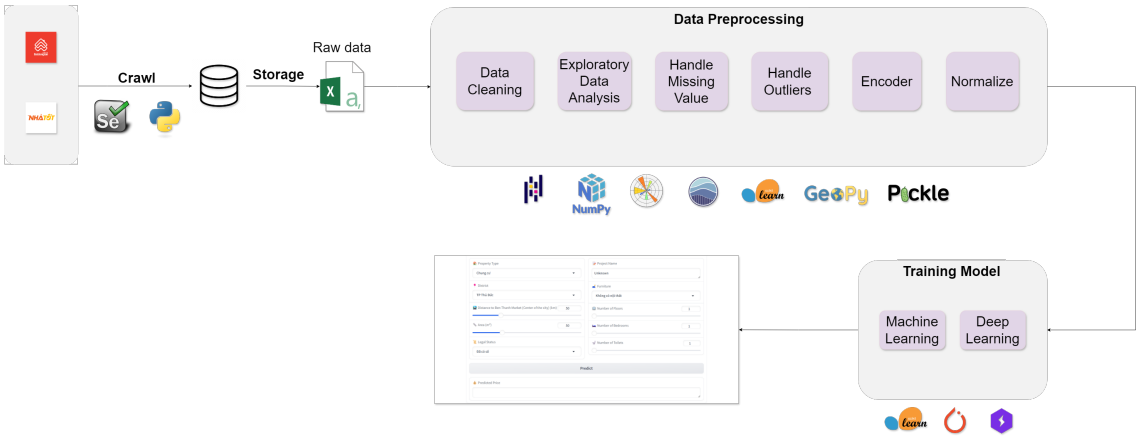
trong đó Z là tập hợp các tài liệu có liên quan được tìm kiếm bởi bộ tìm kiếm.

2. **Hàm tối ưu hóa:** Quá trình huấn luyện của RAG nhằm tối ưu hóa hàm mất mát (loss function), chẳng hạn như hàm cross-entropy:

$$\mathcal{L} = -\log P(y|x)$$

Chương 3

Phương pháp thực hiện



Hình 3.1: Phương pháp thực hiện của mô hình dự đoán giá nhà

3.1 Tiền xử lý dữ liệu

Dữ liệu được chúng tôi thu thập được từ hai websites là nhatot.com và batdongsan.com.vn, gồm hơn 50000 điểm dữ liệu ban đầu về bất động sản tại thị trường Thành phố Hồ Chí Minh, dữ liệu bao gồm 26 trường thông tin. Dữ liệu chủ yếu được thu thập trong khoảng thời gian từ đầu tháng 10 đến giữa tháng 12 năm 2023.

Chúng tôi sử dụng thư viện Selenium [1] và ngôn ngữ lập trình Python để xây dựng code dành cho việc thu thập dữ liệu. Các dữ liệu bị thiếu được xử lý bằng hai cách. Đầu tiên, chúng tôi tiến hành lọc ra những dữ liệu nhiều giá trị Null và so sánh với tổng giá trị ban đầu, nhóm chúng tôi quyết định loại bỏ những thuộc tính với số giá trị Null chiếm lượng lớn so với tổng giá trị

[1] https://selenium-python.readthedocs.io/

ban đầu. Thứ hai, chúng tôi sử dụng phương thức IterativeImputer của thư viện Scikit-learn để điền những dữ liệu bị thiếu bằng cách dự đoán những giá trị cần điền vào, nhằm đảm bảo đầy đủ các điểm dữ liệu cũng như bảo toàn tính nhất quán của dữ liệu trước khi đưa vào mô hình máy học. Với các điểm dữ liệu ngoại lai, chúng tôi sử dụng bằng phương pháp INR (Interquartile Range, hay Phạm vi tứ phân vị) là một kỹ thuật phổ biến trong việc làm sạch và phân tích dữ liệu, giúp loại bỏ những giá trị cực đoan, từ đó cải thiện chất lượng dữ liệu và độ chính xác của các mô hình phân tích. Sau đó chúng tôi sẽ mã hóa các giá trị của từng thuộc tính bằng OrdinalEncoder của thư viện Scikit-learn và chuẩn hóa chúng bằng StandardScaler của thư viện Scikit-learn trước khi tiến hành training model.

3.2 Mô hình dự đoán giá nhà

3.2.1 Mô hình dự đoán giá nhà sử dụng Machine Learning

Ở phương pháp áp dụng các mô hình Machine Learning chúng tôi thực hiện các thí nghiệm trên các mô hình họ Regression nhằm dự đoán giá trị bất động sản ở khu vực Hồ Chí Minh.

Các mô hình Regression được huấn luyện để dự đoán giá trị của biến phụ thuộc dựa trên các biến độc lập. Mục tiêu chính của huấn luyện các mô hình Regression là tìm ra mối quan hệ tuyến tính hoặc phi tuyến tính giữa các biến độc lập và biến phụ thuộc trong dữ liệu.

Để xây dựng mô hình cũng như đánh mô hình chúng tôi sử dụng thư viện scikit-learn trong Python để xây dựng và huấn luyện mô hình. Và sau đây là cách những mô hình ML thực hiện nhiệm vụ với dữ liệu của chúng tôi.

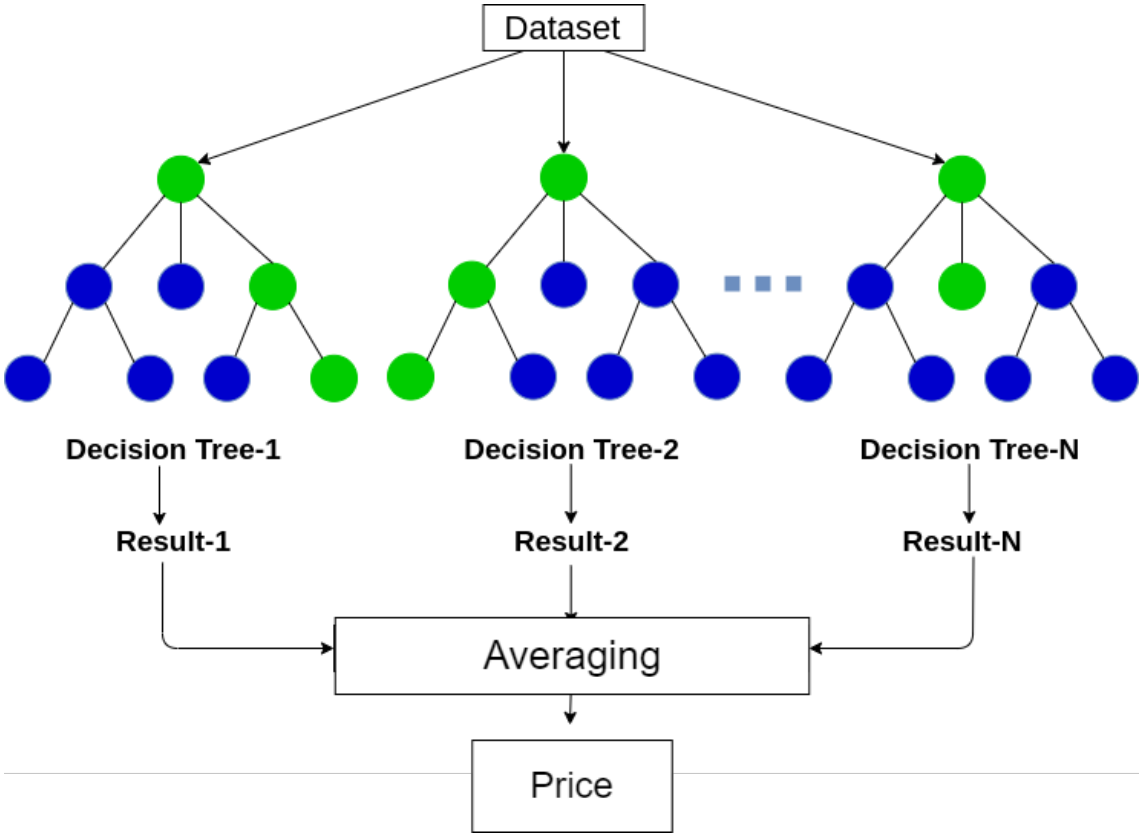
1. **Linear Regression**

   Trong bài toán dự đoán giá nhà, chúng tôi sử dụng mô hình Linear Regression để xây dựng mối quan hệ giữa các đặc trưng của căn nhà và giá nhà. Bộ dữ liệu hơn 40000 điểm dữ liệu với các đặc trưng sau: Property_type, Area, Floors, Bedrooms, Toilets, Legal_status, Furniture,Project_name, District, và Distance. Biến phụ thuộc trong mô hình này là Price.

Thì mô hình hồi quy tuyến tính được biểu diễn bằng phương trình:

$$\text{Price} = \beta_0 + \left(\begin{array}{l} \beta_1 \cdot \text{Property\_type} \\ +\beta_2 \cdot \text{Area} \\ +\beta_3 \cdot \text{Floors} \\ +\beta_4 \cdot \text{Bedrooms} \\ +\beta_5 \cdot \text{Toilets} \\ +\beta_6 \cdot \text{Legal\_status} \\ +\beta_7 \cdot \text{Furniture} \\ +\beta_8 \cdot \text{Project\_name} \\ +\beta_9 \cdot \text{District} \\ +\beta_{10} \cdot \text{Distance} \end{array} \right) + \epsilon$$

2. **Random Forest Regression**



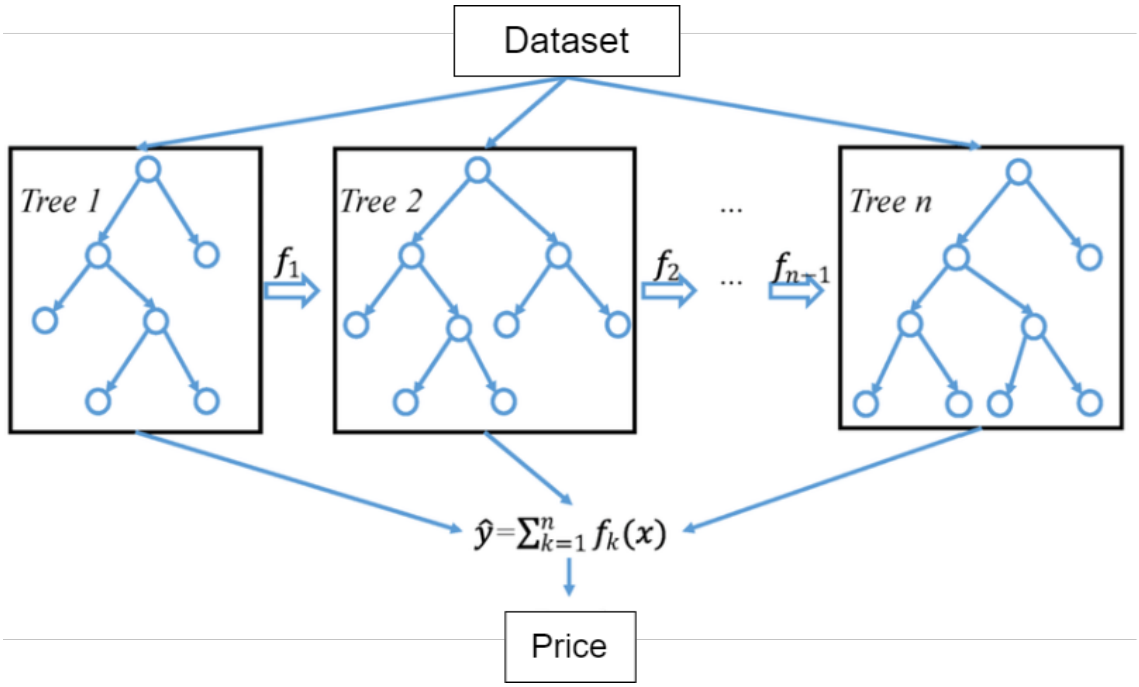
Hình 3.2: Random Forest Regression

Trong trường hợp của chúng tôi, mỗi cây quyết định trong mô hình sẽ được huấn luyện dựa trên một tập con của các điểm dữ liệu và các đặc trưng như Property_type, Area, Floors, và các đặc trưng khác. Và dự đoán cuối cùng của mô hình Random Forest được tính bằng cách lấy trung bình của các dự đoán từ tất cả các cây trong rừng, giúp cải thiện độ chính xác và giảm thiểu sai số so với việc sử dụng một cây quyết định đơn lẻ.

3. **K-Nearest Neighbor Regression**

   Trong mô hình KNN Regression, giá trị dự đoán cho biến Price của một điểm dữ liệu mới được tính dựa trên giá trị của các điểm dữ liệu gần nhất trong tập huấn luyện. Đầu tiên, khoảng cách giữa điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện được tính toán dựa trên các đặc trưng như Property_type, Area, Floors, Bedrooms, Toilets, và Legal_status. Khoảng cách này có thể được tính bằng khoảng cách Euclid hoặc Manhattan. Sau đó, k điểm dữ liệu trong tập huấn luyện có khoảng cách nhỏ nhất đến điểm dữ liệu mới được xác định. Giá trị dự đoán cho Price của điểm dữ liệu mới được tính bằng trung bình cộng của các giá trị Price của k điểm dữ liệu gần nhất. Bằng cách này, giá trị dự đoán phản ánh sự tương đồng giữa điểm dữ liệu mới và các điểm dữ liệu gần nhất trong không gian đặc trưng.

4. **XGBoost Regression**

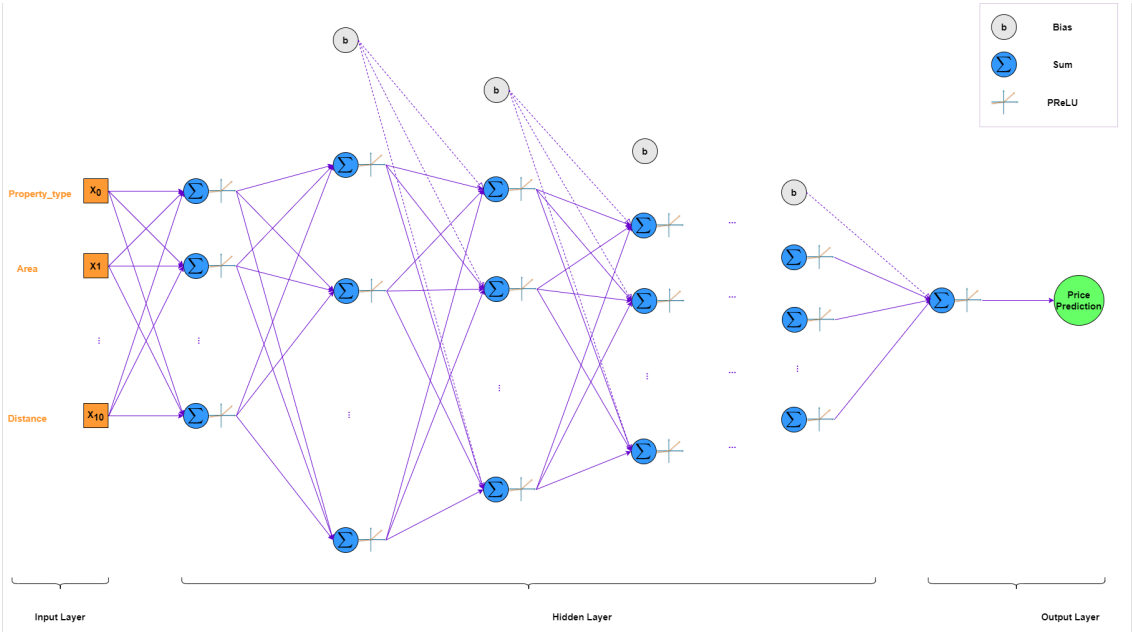


Hình 3.3: XGBoost Regression

Mô hình XGBoost khá tương tự mô hình Random Forest đều kết hợp từ những cây quyết định nhưng ở XGBoost nhưng đoán cuối cùng của mô hình XGBoost được tính bằng cách tổng hợp các dự đoán của tất cả các cây bằng phương pháp gradient boosting, cụ thể là thực hiện cộng dồn dự đoán từ từng cây, mỗi cây học từ các lỗi của cây trước đó, để điều chỉnh và cải thiện dự đoán cuối cùng. Các đặc trưng như Property_type, Area, Floors, và các đặc trưng khác được sử dụng để xây dựng các cây quyết định

3.2.2 Mô hình dự đoán giá nhà sử dụng Deep Learning

Cách hoạt động



Hình 3.4: Regression Neural Network Architecture

1. **Linear Layer**: Mỗi lớp tuyến tính (Linear Layer [12]) thực hiện một phép biến đổi tuyến tính trên biến đầu vào. Với đầu vào x, và đầu ra của một lớp tuyến tính được tính là $Wx + b$, trong đó W là ma trận trọng số và b là vector bias.

2. **Activation Function - PReLU**: Sau mỗi lớp Linear, hàm kích hoạt (Activation Function) PReLU được áp dụng để tạo ra tính phi tuyến tính (Non-linear) vào mô hình. Công thức của PReLU có thể biểu diễn như sau:

$$f(x) = \begin{cases} x & \text{nếu } x > 0 \\ \alpha x & \text{nếu } x \leq 0 \end{cases}$$

   Trong đó, α là tham số học được, liên quan đến độ dốc của phần âm của hàm cho nơ ron đầu vào, giúp tránh vấn đề chết của các neuron (neuron dying problem). Việc α có thể thay đổi và được học thông qua quá trình huấn luyện cho phép mô hình được điều chỉnh linh hoạt hơn, tăng hiệu quả trong việc xử lý các tín hiệu đầu vào đa dạng.

3. **Loss Function - MSELoss**: Hàm mất mát MSELoss, hay còn gọi là Mean Squared Error (MSE), được sử dụng để tính toán độ lệch bình

phương trung bình giữa giá trị dự đoán và giá trị thực tế. Công thức:

$$MSELoss = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Trong đó:

- n là số lượng mẫu.
- y_i là giá trị thực tế của mẫu thứ i.
- $\hat{y}_i$ là giá trị dự đoán của mẫu thứ i.

Hàm mất mát này phạt nặng các sai số lớn hơn nhờ vào việc bình phương độ lệch, do đó mô hình sẽ có xu hướng dự đoán chính xác hơn và ít bị ảnh hưởng bởi các giá trị ngoại lai.

4. **Optimization Algorithm - Adadelta**: Sau mỗi bước cập nhật trọng số, thuật toán tối ưu Adadelta được áp dụng để điều chỉnh các thông số của mô hình. Công thức của Adadelta có thể biểu diễn như sau:

$$\theta_{t+1} = \theta_t - \frac{\sqrt{\Delta x_{t-1} + \epsilon}}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_t$$

Trong đó:

- θ_t là giá trị của thông số tại thời điểm t.
- g_t là gradient của hàm mất mát tại thời điểm t.
- $E[g^2]_t$ là giá trị trung bình động của bình phương gradient tại thời điểm t.
- Δx_t là giá trị trung bình động của các bước cập nhật trọng số.
- ϵ là một giá trị rất nhỏ để tránh chia cho 0.

Adadelta sử dụng một phương pháp tối ưu tự điều chỉnh mà không cần xác định trước tốc độ học (learning rate), giúp cho việc huấn luyện mô hình ổn định hơn và tránh được sự dao động mạnh của gradient.

5. **Scheduler - CosineAnnealingLR**: Bộ điều chỉnh tốc độ học CosineAnnealingLR điều chỉnh tốc độ học theo một chu kỳ cosine. Điều này giúp mô hình thoát khỏi các hố cục bộ và tối ưu hóa hiệu quả hơn. Trong mỗi chu kỳ, tốc độ học sẽ giảm từ giá trị ban đầu đến một giá trị tối thiểu theo hàm cosine, sau đó tăng trở lại, tạo điều kiện cho mô hình có cơ hội khám phá không gian tham số tốt hơn và tìm được các cực tiểu toàn cục.

6. **Regularization Technique - Dropout**: sử dụng thêm các lớp dropout với tỷ lệ thả rơi (dropout rate) là 0.1 để tránh overfitting. Dropout là một kỹ thuật hiệu quả để giảm overfitting bằng cách ngẫu nhiên bỏ qua một số neuron trong quá trình huấn luyện. Điều này có nghĩa là trong mỗi lần huấn luyện, một tỷ lệ phần trăm các neuron được chọn ngẫu nhiên sẽ không được cập nhật trọng số, giúp mô hình không quá phụ thuộc vào bất kỳ đặc trưng nào và do đó cải thiện khả năng tổng quát hóa của mô hình.

Ưu điểm của mô hình dự đoán giá nhà trong bài toán hồi quy

- **Khả năng học phi tuyến tính:** Việc sử dụng hàm kích hoạt PReLU giúp mô hình có khả năng học các mối quan hệ phi tuyến tính trong dữ liệu, làm cho mô hình trở nên mạnh mẽ hơn trong việc dự đoán các giá trị phức tạp. Công thức của PReLU có thể biểu diễn như sau:

$$f(x) = \begin{cases} x & \text{nếu } x > 0 \\ \alpha x & \text{nếu } x \leq 0 \end{cases}$$

  Trong đó, α là tham số học được, liên quan đến độ dốc của phần âm của hàm cho nơ ron đầu vào, giúp tránh vấn đề chết của các neuron (neuron dying problem) và làm cho mô hình linh hoạt hơn.

- **Bền vững với outliers:** Sử dụng hàm mất mát L1 giúp mô hình ít nhạy cảm hơn với outliers so với hàm mất mát L2, làm cho các dự đoán trở nên ổn định hơn. Công thức của L1 loss:

$$L1 = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}|$$

- **Tối ưu hóa hiệu quả:** Adadelta điều chỉnh tỷ lệ học tự động dựa trên các gradient trong các bước trước đó, giúp duy trì ổn định và khả năng hội tụ trong huấn luyện. Công thức của Adadelta:

$$\theta_{t+1} = \theta_t - \frac{\sqrt{\Delta x_{t-1} + \epsilon}}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_t$$

  Adadelta không yêu cầu thiết lập các tham số học cố định, tự động điều chỉnh tỷ lệ học và tránh vấn đề gradient vanishing hay exploding, làm cho quá trình huấn luyện hiệu quả hơn và mô hình tổng quát tốt hơn.

- **Điều chỉnh động tốc độ học:** CosineAnnealingLR điều chỉnh tốc độ học theo một chu kỳ cosine, giúp mô hình thoát khỏi các hố cục bộ và

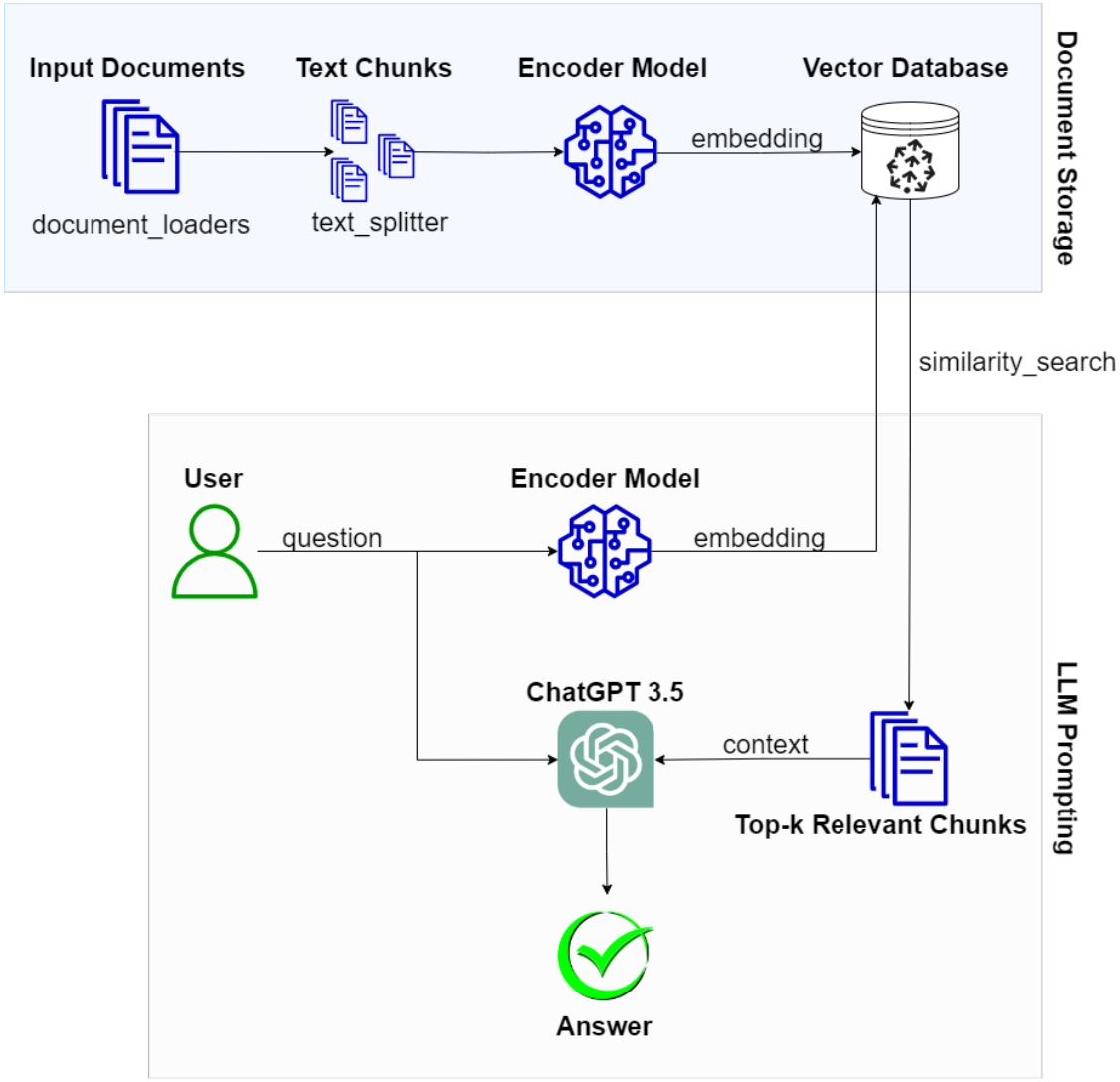
tối ưu hóa hiệu quả hơn. Trong mỗi chu kỳ, tốc độ học sẽ giảm từ giá trị ban đầu đến một giá trị tối thiểu theo hàm cosine, sau đó tăng trở lại, tạo điều kiện cho mô hình có cơ hội khám phá không gian tham số tốt hơn và tìm được các cực tiểu toàn cục.

- **Tránh vấn đề vanishing gradients:** Sử dụng PReLU thay vì ReLU giúp tránh vấn đề vanishing gradients và dying neurons, giúp mô hình huấn luyện ổn định hơn và hội tụ tốt hơn.
- **Tránh overfitting:** Để tránh overfitting, tôi đã thêm các lớp dropout với tỷ lệ thả rơi (dropout rate). Dropout là một kỹ thuật hiệu quả để giảm overfitting bằng cách ngẫu nhiên bỏ qua một số neuron trong quá trình huấn luyện. Điều này có nghĩa là trong mỗi lần huấn luyện, một tỷ lệ phần trăm các neuron được chọn ngẫu nhiên sẽ không được cập nhật trọng số, giúp mô hình không quá phụ thuộc vào bất kỳ đặc trưng nào và do đó cải thiện khả năng tổng quát hóa của mô hình.

3.3 Mô hình chatbot

3.3.1 Kiến trúc tổng quan mô hình chatbot

Mô hình chatbot được xây dựng với kiến trúc tổng quan gồm hai quá trình chạy song song với nhau lần lượt là Document Storage và LLM Prompting, được mô tả như hình bên dưới:



Hình 3.5: RAG Architecture Model

1. **Document Storage**

   (a) **Tài liệu đầu vào**

   - **Document Loaders (Trình tải tài liệu)**: Tải các tài liệu (ví dụ: PDF, file văn bản) vào hệ thống. Đọc dữ liệu từ một file văn bản sử dụng hàm `Load_document_from_file`.

   (b) **Text Chunks (Các đoạn văn bản)**

   - **Text Splitter (Chia đoạn văn bản)**: Chia các tài liệu thành những đoạn văn bản nhỏ, vì xử lý các đoạn văn bản nhỏ hơn sẽ nhanh hơn và hiệu quả hơn so với việc xử lý toàn bộ tài liệu lớn. Đồng thời việc chia văn bản thành các đoạn nhỏ giúp tìm kiếm dễ dàng và chính xác hơn, nhóm chúng tôi sẽ sử dụng một hàm `read_sentences_from_document` để giúp chia văn bản thành các văn bản nhỏ.

(c) **Mô hình mã hóa**

- **Encoding (Mã hóa)**: Sử dụng mô hình Sentence-BERT `intfloat/multilingual-e5-small` từ thư viện `sentence-transformers` của Hugging Face để mã hóa các câu thành các embedding vector có kích thước 384.

(d) **Cơ sở dữ liệu vector**

- **Storing Embeddings (Lưu trữ embeddings)**: Lưu trữ các embeddings trong Pinecone - là một vector database - để truy xuất hiệu quả. Cơ sở dữ liệu vector này được cấu hình và khởi tạo nếu chưa tồn tại, sau đó các vector embeddings sẽ được upsert vào cơ sở dữ liệu.

(e) **Tìm kiếm tương đồng**

- **Retrieving Relevant Chunks (Truy xuất các đoạn văn bản liên quan)**: Khi một câu hỏi được đặt ra, chuyển câu hỏi thành một embedding thông qua mô hình mã hóa ở trên và truy vấn (query) các đoạn văn bản tương tự từ cơ sở dữ liệu vector. Truy vấn khi thực hiện trên Pinecone sẽ trả về các index (chỉ mục) của các vector data tương đồng dựa theo hàm Cosine, kèm theo metadata của mỗi vector đó. Thường metadata sẽ bao gồm các text, date của vector đó, các text này sẽ là văn bản chưa được encode (tức là dữ liệu nguyên bản). Các text này sau đó được sử dụng làm context cho mô hình ngôn ngữ của OpenAI để tạo ra câu trả lời. Qua đó việc cần làm tiếp theo là chọn Top-k các vector tương đồng nhất mà đáp ứng được câu trả hỏi của người dùng.

2. **LLM Prompting**

- **LLM Prompting**: Xây dựng một pipeline cuộc trò chuyện cho người dùng, các thành phần chính trong LLM Prompting sẽ là:
  - **User:** Người dùng sẽ sử dụng thiết bị đầu cuối, truy cập thông qua các phương thức khác nhau để sử dụng chatbot, có thể là trên website, app, ...
  - **Encoder Model:** Ở quá trình này sẽ được sử dụng cùng một model với Encoder Model ở quá trình Document Storage, bởi vì cùng một câu từ nếu sử dụng Encoder Model khác nhau sẽ cho ra kết quả khác nhau. Câu hỏi của người sẽ được encode để tạo thành một vector và truy vấn đến database để tìm kiếm trong cơ

sở dữ liệu và tìm ra các vector có sự tương đồng nhất đối với câu hỏi.

- **Top-k Relevant Chunks:** Đây là top-k các dữ liệu được truy vấn về từ cơ sở dữ liệu mà có sự tương đồng nhất với question của user. Phương thức để tính toán sự tương đồng là Cosine, sau đó sẽ chọn ra top-k sự tương đồng nhất để tạo context cho LLM Model.
- **LLM Model:** Sử dụng API ChatGPT 3.5 sẽ phù hợp đối với việc không có đủ VRAM để khởi chạy mô hình LLM trên máy tính cá nhân. LLM Model sẽ nhận Context từ Top-k Relevant Chunks và Question của người dùng để tạo Answer. Answer sẽ được hiển thị cho người dùng thông qua User Interface.

3.3.2 Cách thức hoạt động của mô hình chatbot

Mô hình chatbot trong lĩnh vực bất động sản sử dụng RAG và ChatGPT-3.5 hoạt động qua các bước chính: nhập liệu, truy xuất thông tin, tích hợp thông tin và sinh văn bản. Dưới đây là mô tả chi tiết từng bước trong quy trình này.

1. **Nhập liệu:** Quá trình bắt đầu khi người dùng nhập câu hỏi hoặc yêu cầu thông tin vào hệ thống chatbot. Các câu hỏi này có thể bao gồm:
   - Giá nhà ở một khu vực cụ thể.
   - Tình trạng bất động sản hiện tại.
   - Thông tin về các tiện ích xung quanh khu vực bất động sản.
   - Xu hướng thị trường bất động sản trong một khu vực nhất định.

   Ví dụ: Người dùng có thể hỏi "Giá nhà trung bình ở quận 1, TP.HCM hiện tại là bao nhiêu?"

2. **Truy xuất thông tin:** Sau khi nhận được câu hỏi từ người dùng, hệ thống sẽ sử dụng mô hình truy xuất thông tin để tìm kiếm các đoạn văn bản liên quan từ cơ sở dữ liệu đã được thu thập trước đó. Dữ liệu này được crawl từ nhiều nguồn khác nhau, bao gồm:
   - Trang web bất động sản.
   - Báo cáo thị trường.
   - Các bài viết, tin tức liên quan đến bất động sản.
   - Thông tin từ các cơ quan quản lý bất động sản.

Mô hình truy xuất sẽ tìm kiếm và chọn ra các đoạn văn bản có liên quan nhất đến câu hỏi của người dùng.

Ví dụ: Khi nhận được câu hỏi về giá nhà ở quận 1, hệ thống sẽ tìm kiếm các dữ liệu về giá nhà, xu hướng giá, và các yếu tố ảnh hưởng đến giá nhà trong khu vực đó.

3. **Tích hợp thông tin:** Các đoạn văn bản được truy xuất sẽ được kết hợp lại để tạo ra một ngữ cảnh phong phú hơn cho quá trình sinh văn bản. Quá trình này bao gồm:

   - Loại bỏ các thông tin không liên quan hoặc trùng lặp.
   - Kết hợp các đoạn thông tin từ nhiều nguồn khác nhau để tạo ra một bức tranh toàn cảnh.
   - Tạo ra một ngữ cảnh đầy đủ và chi tiết để mô hình sinh văn bản có thể sử dụng.

   Ví dụ: Thông tin về giá nhà trung bình, xu hướng giá trong 6 tháng qua, và các tiện ích xung quanh (như trường học, bệnh viện, siêu thị) sẽ được kết hợp để cung cấp một cái nhìn toàn diện về khu vực quận 1.

4. **Sinh văn bản:** Cuối cùng, hệ thống sẽ sử dụng mô hình sinh văn bản (ChatGPT-3.5) để tạo ra câu trả lời hoàn chỉnh và tự nhiên dựa trên thông tin đã được tích hợp. Mô hình này sẽ:

   - Sử dụng ngữ cảnh được cung cấp để sinh ra văn bản mạch lạc và dễ hiểu.
   - Đảm bảo rằng câu trả lời đầy đủ và chính xác, phản ánh đúng các thông tin đã được truy xuất và tích hợp.

   Ví dụ: Mô hình sẽ tạo ra câu trả lời như sau: "Giá nhà trung bình ở quận 1, TP.HCM hiện tại là khoảng 100 triệu VND/m². Trong 6 tháng qua, giá nhà đã tăng khoảng 5%. Khu vực này có nhiều tiện ích như trường học, bệnh viện, và siêu thị lớn, rất thuận tiện cho cuộc sống hàng ngày."

3.3.3 Ưu và nhược điểm của mô hình được xây dựng bởi RAG

Loại mô hình	Ưu điểm	Nhược điểm
Mô hình sử dụng RAG	**Chất lượng thông tin cao**: Mô hình RAG có khả năng truy xuất và tích hợp thông tin từ nhiều nguồn, giúp đảm bảo câu trả lời đầy đủ và chính xác hơn. **Khả năng tổng hợp tốt**: Bằng cách kết hợp thông tin từ nhiều nguồn, mô hình có thể tạo ra các câu trả lời phong phú và chi tiết hơn so với các mô hình chỉ sử dụng một nguồn thông tin. **Linh hoạt**: RAG có thể được áp dụng trong nhiều lĩnh vực khác nhau như tìm kiếm thông tin, dịch vụ khách hàng, giáo dục và nghiên cứu khoa học.	**Độ phức tạp cao**: Cấu trúc của RAG phức tạp hơn so với các mô hình sinh văn bản thông thường, yêu cầu tài nguyên tính toán lớn và thời gian huấn luyện dài. **Phụ thuộc vào chất lượng cơ sở dữ liệu**: Hiệu quả của mô hình phụ thuộc nhiều vào chất lượng và độ phong phú của cơ sở dữ liệu được sử dụng để truy xuất thông tin. **Khả năng sai lệch thông tin**: Nếu cơ sở dữ liệu chứa thông tin sai lệch hoặc không đầy đủ, mô hình có thể tạo ra các câu trả lời không chính xác hoặc thiên lệch.
Mô hình sinh văn bản truyền thống	Mô hình truyền thống như GPT-3 có thể tạo ra văn bản một cách nhanh chóng mà không cần truy xuất thông tin từ các nguồn bên ngoài.	Chất lượng câu trả lời có thể bị giới hạn bởi kiến thức đã học được trong quá trình huấn luyện và không thể cập nhật thông tin mới một cách liên tục.
Mô hình truy xuất thông tin	Các mô hình truy xuất thông tin có thể nhanh chóng tìm kiếm và cung cấp thông tin cụ thể từ một cơ sở dữ liệu lớn.	Chỉ cung cấp thông tin thô, không có khả năng sinh văn bản mạch lạc và tự nhiên như mô hình RAG.
Mô hình kết hợp	Kết hợp cả khả năng truy xuất và sinh văn bản, giúp tạo ra các câu trả lời phong phú và chính xác hơn.	Độ phức tạp cao và yêu cầu tài nguyên tính toán lớn, cần cân nhắc khi triển khai trong các hệ thống thực tế.

Bảng 3.1: Bảng so sánh giữa các loại mô hình khác được sử dụng trong xây dựng chatbot

3.3.4 Ưu điểm của RAG trong tư vấn bất động sản

- **Độ chính xác cao:** Nhờ vào khả năng truy xuất thông tin từ một lượng dữ liệu lớn, RAG có thể cung cấp các câu trả lời chính xác và cập nhật, đáp ứng được nhu cầu tìm kiếm thông tin nhanh chóng của khách hàng.

- **Tính tự nhiên trong câu trả lời:** Khả năng tạo ngôn ngữ tự nhiên giúp cho câu trả lời của chatbot trở nên thân thiện và dễ hiểu, cải thiện trải nghiệm người dùng.

- **Tùy biến dựa trên ngữ cảnh:** RAG có thể tùy chỉnh câu trả lời dựa trên ngữ cảnh của câu hỏi, giúp cung cấp thông tin phù hợp và chi tiết hơn cho từng khách hàng cụ thể.

- **Tiết kiệm thời gian và chi phí:** Việc tự động hóa quá trình tư vấn giúp giảm thiểu thời gian và chi phí cho doanh nghiệp, đồng thời tăng hiệu quả làm việc của nhân viên tư vấn.

Chương 4

Thực nghiệm và kết quả

4.1 Dữ liệu

Dữ liệu để phục vụ cho bài toán dự đoán giá nhà được chúng tôi thu thập từ hai websites là batdongsan.com.vn và nhatot.com bằng thư viện Selenium và sử dụng ngôn ngữ lập trình Python.

Dữ liệu thu thập sẽ bao gồm các chung cư và nhà mặt đất đang được rao bán. Các dữ liệu thu thập bước đầu sẽ gồm các thông tin: như Loại bất động sản địa chỉ, giá bán, diện tích, thời điểm đăng bán, ... Sau khi hoàn thành việc thu thập dữ liệu, nhóm đã thu được hơn 50000 điểm dữ liệu gồm các thuộc tính:

Tên thuộc tính	**Ý nghĩa**
Property_type	Loại hình bất động sản: Nhà biệt thự; Nhà mặt phố; Nhà ngõ, hẻm; Nhà phố liên kế; Chung cư
Area	Diện tích của bất động sản (m^2
Width	Chiều rộng của bất động sản (m)
Length	Chiều dài của bất động sản (m)
Frontage	Mặt tiền (m), chỉ có ở loại hình nhà đất
Number_of_floors	Số tầng, chỉ có ở loại hình nhà đất
Number_of_bedrooms	Số phòng ngủ
Number_of_toilets	Số toilets
Legal_status	Tình trạng pháp lý: Đã có sổ, Đang chờ sổ, Giấy tờ khác, Unknown
Furniture	Nội thất: Có nội thất, Không nội thất, Bàn giao thô
House_orientation	Hướng nhà: Đông, Tây, Nam, Bắc, Đông Nam, Đông Bắc, Tây Nam, Tây Bắc
Balcony_orientation	Hướng ban công: Đông, Tây, Nam, Bắc, Đông Nam, Đông Bắc, Tây Nam, Tây Bắc
Access_road	Đường vào (m), chỉ có ở loại hình nhà đất
Posting_date	Ngày đăng
Expiry_date	Ngày hết hạn
Type_of_lists	Loại tin
Project_name	Tên dự án, chỉ có ở loại hình chung cư
Street	Tên đường
Ward	Tên Xã/Phường/Thị trấn
District	Tên Quận/Huyện
Province	Tên Tỉnh thành
Distance	Khoảng cách từ Quận/Huyện của bất động sản đến trung tâm thành phố (Bến Thành)
Price	Giá của bất động sản

Bảng 4.1: Bảng thuộc tính của dữ liệu

Ở đây chúng tôi đã tiến hành loại bỏ một số cột trước khi training model như: Frontage, Street, Ward, Province, Width, Length, House_orientation, Balcony_orientation, Access_road, Posting_date, Expiry_date,
Type_of_listing. Dữ liệu sau khi được tiền xử lý sẽ bao gồm 10 thuộc tính là: Property_type, Area, Floors, Bedrooms, Toilets, Legal_status, Furniture, Project_name, District, Distance và Price gồm hơn 40000 điểm dữ liệu, với Price là label của tập dữ liệu. Sau đó, chúng tôi sẽ sử dụng phương thức trong Model Selection của thư viện scikit-learn là train_test_split để chia dữ liệu thành các tập train, test và validation với tỉ lệ là 8:1:1.

4.2 Các phương pháp đánh giá

4.2.1 Metrics đánh giá mô hình dự đoán giá nhà

1. **Varian Score (R^2 Score)**

- **Định nghĩa:** Varian Score [9] hay R^2 Score đo lường tỷ lệ biến thiên của dữ liệu được giải thích bởi mô hình. Nó có giá trị từ 0 đến 1 (đôi khi có thể nhỏ hơn 0 nếu mô hình rất tệ).
- **Công thức:**

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Trong đó y_i là giá trị thực, $\hat{y}_i$ là giá trị dự đoán và $\bar{y}$ là giá trị trung bình của y.

2. **Mean Absolute Error (MAE)**

- **Định nghĩa:** MAE [8] đo lường sai số tuyệt đối trung bình giữa các giá trị dự đoán và giá trị thực. Nó cho biết sai số trung bình của các dự đoán mà không quan tâm đến hướng của sai số.
- **Công thức:**

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

3. **Mean Squared Error (MSE)**

- **Định nghĩa:** MSE [2] đo lường sai số bình phương trung bình giữa các giá trị dự đoán và giá trị thực. Nó đánh giá mức độ chênh lệch giữa các giá trị dự đoán và giá trị thực với trọng số lớn hơn cho các sai số lớn.
- **Công thức:**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

4. **Root Mean Squared Error (RMSE)**

- **Định nghĩa:** RMSE là căn bậc hai của MSE, cung cấp một thước đo về sai số dự đoán trong cùng đơn vị với giá trị thực.
- **Công thức:**

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

4.2.2 Metrics đánh giá mô hình RAG trong thực tế

Chúng tôi sử dụng các metrics sau của thư viện Ragas để đánh giá mô hình chatbot:

1. **Context Precision**

- **Định Nghĩa**: Tỷ lệ giữa số ngữ cảnh đúng mà mô hình chọn ra và tổng số ngữ cảnh mà mô hình đã trả về.
- **Công Thức**:

$$\text{Context Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Ý Nghĩa**: Đo lường khả năng của mô hình trong việc cung cấp ngữ cảnh chính xác liên quan đến câu hỏi.

2. **Faithfulness**

- **Định Nghĩa**: Đo lường mức độ mà câu trả lời của mô hình phản ánh đúng thông tin từ ngữ cảnh.
- **Công Thức**: Không có công thức cụ thể, nhưng thường được đánh giá dựa trên sự so sánh giữa câu trả lời và ngữ cảnh.
- **Ý Nghĩa**: Chỉ số này cho biết độ tin cậy của câu trả lời.

3. **Answer Relevancy**

- **Định Nghĩa**: Tỷ lệ giữa số câu trả lời liên quan đến câu hỏi và tổng số câu trả lời được cung cấp.
- **Công Thức**:

$$\text{Answer Relevancy} = \frac{\text{Relevant Answers}}{\text{Total Answers}}$$

- **Ý Nghĩa**: Đo lường khả năng của mô hình trong việc cung cấp câu trả lời phù hợp với câu hỏi.

4. **Context Recall**

- **Định Nghĩa**: Tỷ lệ giữa số ngữ cảnh đúng mà mô hình thu hồi được và tổng số ngữ cảnh đúng có trong dữ liệu.
- **Công Thức**:

$$\text{Context Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **Ý Nghĩa**: Đánh giá khả năng của mô hình trong việc nhận diện và sử dụng tất cả các ngữ cảnh quan trọng.

Ngoài ra chatbot còn được đánh giá theo các tiêu chí dưới đây:

- **Độ chính xác (Accuracy):** Độ chính xác là thước đo quan trọng để đánh giá mức độ mà các câu trả lời của mô hình phản ánh đúng thông tin có sẵn trong cơ sở dữ liệu. Để tính toán độ chính xác, ta so sánh các câu trả lời của mô hình với các câu trả lời đúng đã biết trước. Độ chính xác cao cho thấy mô hình có khả năng truy xuất và cung cấp thông tin đúng đắn.
- **Độ hài lòng của người dùng (User Satisfaction):** Độ hài lòng của người dùng được đánh giá thông qua khảo sát hoặc phản hồi của người dùng về chất lượng và tính hữu ích của các câu trả lời. Các phương pháp thu thập bao gồm bảng câu hỏi, đánh giá sao, hoặc các bình luận trực tiếp từ người dùng. Một mô hình được đánh giá cao về độ hài lòng thường cung cấp thông tin chính xác, dễ hiểu và hữu ích.
- **Tỉ lệ truy xuất thành công (Successful Retrieval Rate):** Đây là tỉ lệ phần trăm các lần mô hình thành công trong việc tìm kiếm và truy xuất các đoạn văn bản liên quan từ cơ sở dữ liệu so với tổng số lần truy xuất. Một tỉ lệ truy xuất thành công cao cho thấy mô hình có khả năng tìm kiếm và truy xuất thông tin hiệu quả, đáp ứng được yêu cầu của người dùng.
- **Thời gian phản hồi (Response Time)** Thời gian phản hồi là khoảng thời gian từ khi người dùng đặt câu hỏi đến khi nhận được câu trả lời từ chatbot. Thời gian phản hồi ngắn cho thấy hệ thống hoạt động hiệu quả và có khả năng xử lý truy vấn nhanh chóng, điều này rất quan trọng trong việc cung cấp trải nghiệm người dùng tốt.
- **Tính đa dạng (Diversity)** Tính đa dạng đánh giá mức độ mà các câu trả lời của mô hình không bị lặp lại hoặc quá giống nhau. Tính đa dạng cao cho thấy mô hình có khả năng cung cấp nhiều góc nhìn khác nhau về một chủ đề, giúp người dùng có được cái nhìn toàn diện hơn. Điều này rất quan trọng trong các hệ thống đối thoại thông minh nhằm tránh sự nhàm chán và đơn điệu.

4.3 Kết quả thực nghiệm

4.3.1 Mô hình dự đoán giá nhà

Chi tiết triển khai:

- Đối với các mô hình Machine Learning chúng tôi triển khai trên Google Colab với scikit-learn, huấn luyện trên CPU.

- Đối với mô hình Mạng Neural Hồi Quy chúng tôi được triển khai trong PyTorch, Pytorch Lightning và huấn luyện trên GPU NVIDIA GeForce RTX 4060 Ti bằng bộ tối ưu AdamW, learning rate scheduler CosineAnnealingLR trong 450 epoch, với batch size là 128 và learning-rate là 0.1

Trong giai đoạn huấn luyện và kiểm tra mô hình dự đoán giá nhà của chúng tôi, chúng tôi đã áp dụng một phương pháp mô hình ML và DL bằng cách huấn luyện và kiểm tra dữ liệu bằng những thuật toán khác nhau. Cách tiếp cận này cho phép chúng tôi đánh giá hiệu suất và tính hiệu quả của từng thuật toán trong việc dự đoán giá nhà. Thuật toán được sử dụng trong nghiên cứu của chúng tôi bao gồm Linear Regression, Random Forest Regression, và K-Nearest Neighbor Regression, XGBoost Regression và Mạng Neural Hồi Quy (Regression Neural Network). Bằng cách sử dụng nhiều thuật toán, chúng tôi nhằm mục đích nắm bắt một loạt các kỹ thuật mô hình hóa và xác định phương pháp có hiệu suất tốt nhất cho nhiệm vụ dự đoán giá nhà cụ thể của chúng tôi. Các chỉ số đánh giá như MSE, RMSE, MAE và R^2 được sử dụng để so sánh và đánh giá độ chính xác và khả năng dự đoán của mô hình đã được huấn luyện.

Model	**Score**	**MSE**	**RMSE**	**MAE**
Linear Regression	0.6364	10.9652	3.3113	2.2173
Random Forest Regression	0.8576	4.2944	2.0723	1.1346
K-Nearest Neighbor Regression	0.8231	5.3372	2.3102	1.2392
XGBoost Regression	0.866	4.0401	2.0100	1.1483
Regression Neural Network	**0.9114**	**2.57**	**1.5827**	**0.9855**

Bảng 4.2: Bảng kết quả thực nghiệm dự đoán giá nhà

Kết quả thực nghiệm cho thấy mô hình sử áp dụng Deep Learning tốt hơn so với mô hình áp dụng Machine Learning, cụ thể là mô hình Neural Hồi Quy cho kết quả tốt nhất trong các mô hình được chúng tôi sử dụng với Variance Score là 0.9114, các chỉ số MSE, RMSE và MAE lần lượt là 2.57, 1.5827 và 0.9855. Trong phương pháp áp dụng Machine Learning, mô hình XGBoost Regression cho kết quả tốt nhất với Variance Score là 0.866, các chỉ số MSE, RMSE và MAE lần lượt là 4.0401, 2.0100 và 1.1483. Điều này cho thấy Regression Neural Network là lựa chọn tối ưu nhất trong bối cảnh nghiên cứu của chúng tôi, mang lại sự cải thiện đáng kể trong độ chính xác và khả năng dự đoán so với các phương pháp Machine Learning truyền thống.

4.3.2 Mô hình chatbot

Chi tiết triển khai: Đối với mô hình chatbot sử dụng RAG chúng tôi được triển khai trong Pytorch Lightning, OpenAI, và dùng encoder là Sentence-Transformer và chạy trên GPU NVIDIA GeForce RTX 4060 Ti.

Dưới đây là kết quả đánh giá đối với chatbot sử dụng các phương thức đánh giá trong thư viện Ragas:

Model	Context Precision	Faithfulness	Answer Relevancy	Context Recall
Chatbot	1.0000	0.4500	0.7778	0.9583

Bảng 4.3: Bảng kết quả đánh gia chatbot

Mô hình cho thấy điểm mạnh trong việc chọn lọc ngữ cảnh và cung cấp câu trả lời có liên quan. Với chỉ số Context Precision đạt 1.0000 và Context Recall là 0.9583, mô hình thể hiện khả năng xác định và sử dụng ngữ cảnh chính xác rất tốt. Answer Relevancy đạt 0.7778 cho thấy câu trả lời thường phù hợp với yêu cầu của câu hỏi. Tuy nhiên, Faithfulness chỉ đạt 0.4500, cho thấy mức độ trung thực của câu trả lời vẫn chưa cao do chi phí tính toán khá cao và chi phí để vận hành khá lớn nên bị giới hạn về số lượng token khi trả lời, việc dữ liệu quá lớn cũng là một vấn đề khi embedding dữ liệu.

Ngoài ra kết quả thực nghiệm của mô hình chatbot còn được thể hiện qua các tiêu chí đánh giá được trình bày dưới đây:

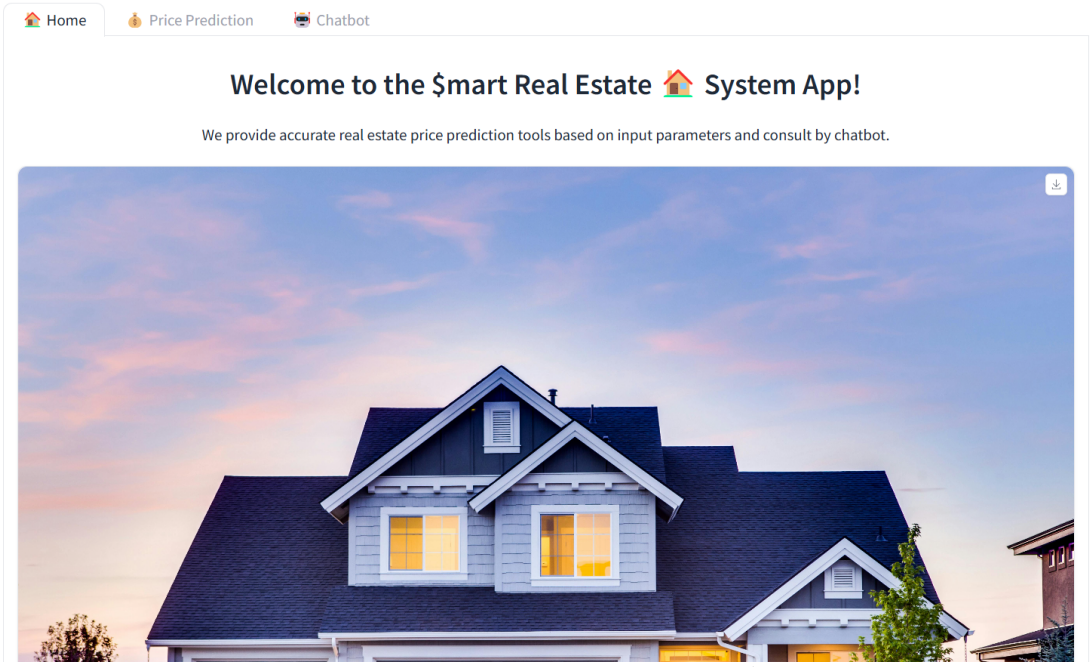
- **Độ chính xác:** Chatbot cung cấp thông tin khá chính xác về dữ liệu bất động sản được thu thập. Câu trả lời đầy đủ và cung cấp tất cả các thông tin cần thiết như giá, diện tích, tiện ích, v.v, tùy thuộc vào câu hỏi của người dùng.
- **Độ hài lòng của người dùng:** Chúng tôi xây dựng mô hình chatbot với giao diện và trải nghiệm người dùng dễ sử dụng và thân thiện.
- **Tỉ lệ truy xuất thành công:** Hệ thống tìm kiếm của chatbot phải chính xác trong việc truy xuất thông tin từ cơ sở dữ liệu, không trả lời những câu trả lời thiếu tính chính xác.
- **Tính đa dạng:** Chatbot có thể đưa ra một số câu trả lời đa dạng ở những câu hỏi cụ thể, có thể cải thiện câu trả lời, nhưng cần cải thiện nhiều để tăng sự đa dạng cũng như là tăng khả năng học hỏi của chatbot.

Nhìn chung thì kết quả thực nghiệm của chatbot của nhóm cũng đạt được kết quả tương đối khả quan.

Chương 5

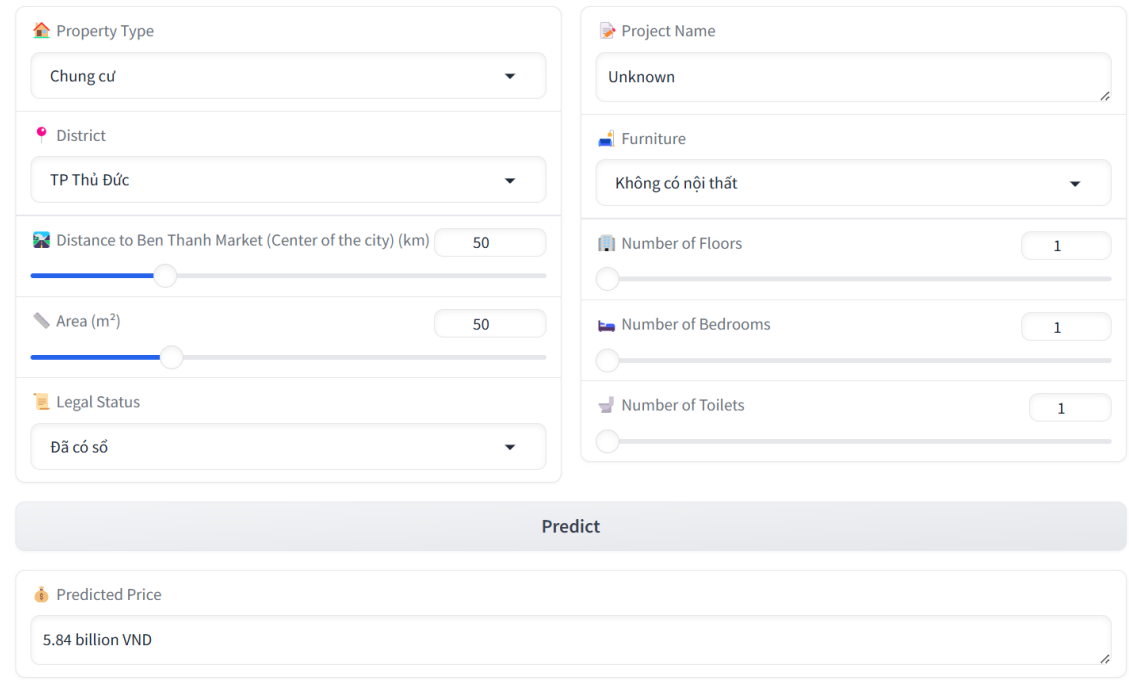
Ứng dụng

Chúng tôi hiện tại đang tiến hành nghiên cứu, phát triển và ứng dụng công nghệ về ngành bất động sản bằng cách xây dựng ứng dụng dự đoán giá trị tài sản và chatbot hỗ trợ khách hàng bằng thư viện Gradio kết hợp với HTML và CSS, sử dụng ngôn ngữ lập trình Python. Công nghệ này giúp tăng độ chính xác trong dự đoán và cải thiện trải nghiệm người dùng với các dịch vụ tư vấn tự động và thông minh nhằm giải quyết bài toán được đặt ra ban đầu tại lĩnh vực bất động sản.



Hình 5.1: Giao diện trang chủ của hệ thống

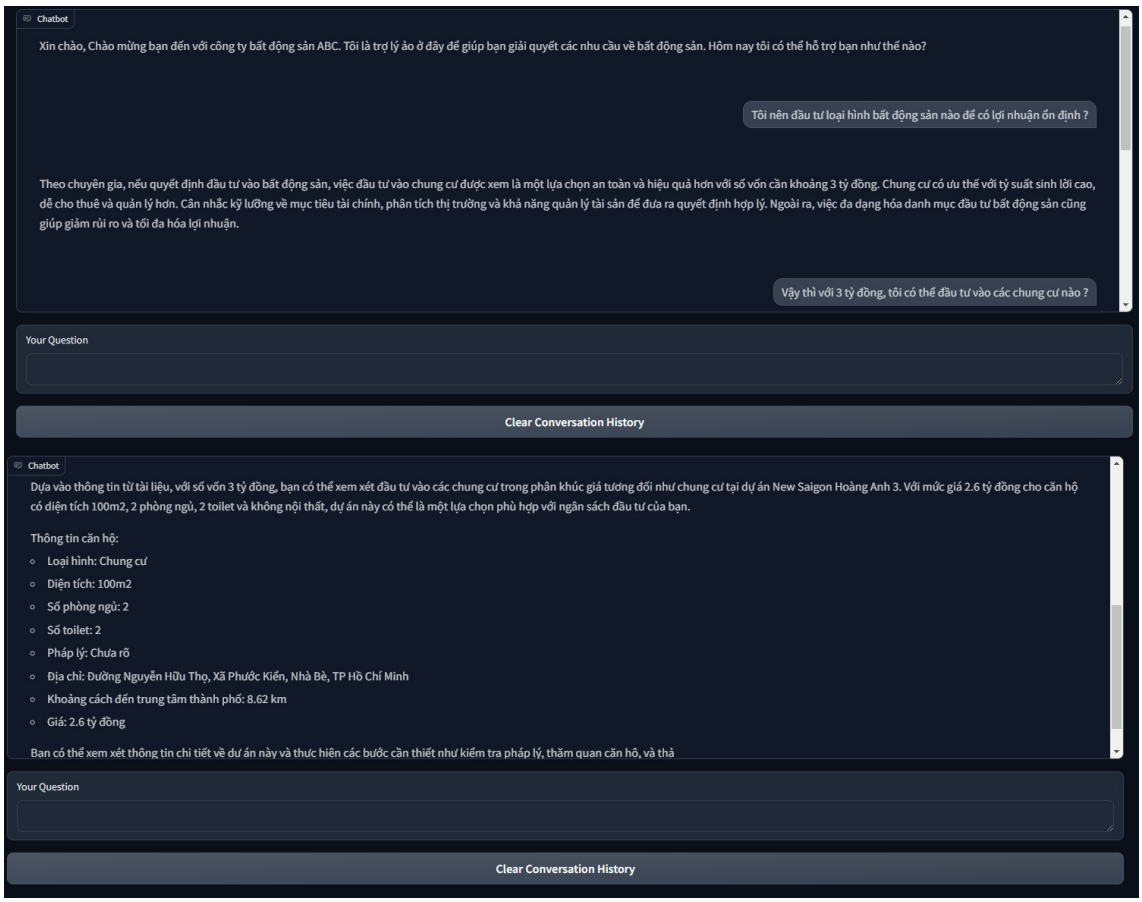
5.1 Xây dựng ứng dụng dự đoán giá bất động sản



Hình 5.2: Giao diện về tab dự đoán giá nhà

Để thuận tiện cho người dùng trong việc có thể sử dụng ứng dụng Deep Learning vào dự đoán giá bất động sản, chúng tôi đã tiến hành xây dựng một giao diện dùng để dự đoán giá bất động sản. Với giao diện này, người dùng chỉ cần nhập vào các giá trị được biểu diễn ở các thuộc tính như diện tích, số phòng ngủ, số phòng tắm, vị trí, và nhiều yếu tố khác, sau đó nhấn vào nút "Predict". Kết quả trả về sẽ là giá được dự đoán của bất động sản từ các thông tin mà người dùng nhập vào.

5.2 Xây dựng chatbot tư vấn bất động sản



Hình 5.3: Giao diện về tab chatbot

Với sự phát triển mạnh mẽ của trí tuệ nhân tạo, việc ứng dụng các mô hình ngôn ngữ tiên tiến để tạo ra các chatbot hỗ trợ tư vấn bất động sản đã trở nên phổ biến. Một trong những mô hình nổi bật hiện nay là RAG. RAG kết hợp giữa khả năng tìm kiếm thông tin và khả năng tạo ngôn ngữ tự nhiên, mang lại nhiều lợi ích cho ngành tư vấn bất động sản.

Lợi ích của chatbot tư vấn bất động sản:

- **Tiết kiệm thời gian:** Chatbot có thể cung cấp thông tin ngay lập tức cho người dùng, giúp tiết kiệm thời gian so với việc phải tìm kiếm hoặc chờ đợi phản hồi từ nhân viên tư vấn.
- **Hỗ trợ 24/7:** Chatbot có thể hoạt động liên tục 24/7, cung cấp hỗ trợ không ngừng nghỉ cho người dùng.
- **Tăng cường trải nghiệm người dùng:** Với khả năng tương tác tự nhiên và chính xác, chatbot có thể cải thiện trải nghiệm của người dùng khi tìm kiếm thông tin về bất động sản.
- **Phân tích và cá nhân hóa:** Chatbot có thể phân tích nhu cầu và sở thích của người dùng để đưa ra các gợi ý bất động sản phù hợp, nâng

cao hiệu quả tư vấn.

Với ứng dụng này, người dùng có thể dễ dàng đặt câu hỏi về thị trường bất động sản, nhận được các gợi ý về mua bán, cho thuê, hoặc đầu tư, và nhiều thông tin hữu ích khác mà không cần phải tương tác trực tiếp với nhân viên tư vấn.

5.3 Kết hợp hai ứng dụng

Việc kết hợp ứng dụng dự đoán giá bất động sản và chatbot tư vấn bất động sản mang lại nhiều lợi ích vượt trội:

- **Tính toàn diện:** Kết hợp hai ứng dụng này tạo ra một hệ thống hoàn chỉnh, từ việc cung cấp thông tin tư vấn, giải đáp thắc mắc đến dự đoán giá trị bất động sản, đáp ứng mọi nhu cầu của người dùng.
- **Tiện lợi và hiệu quả:** Người dùng có thể vừa được tư vấn chi tiết về bất động sản, vừa có thể nhanh chóng dự đoán giá trị bất động sản mà họ quan tâm, giúp quá trình ra quyết định mua bán trở nên nhanh chóng và chính xác hơn.
- **Cải thiện trải nghiệm người dùng:** Sự kết hợp này giúp người dùng có một trải nghiệm mượt mà, không cần phải chuyển đổi giữa nhiều ứng dụng hoặc nền tảng khác nhau. Họ có thể nhận được tất cả thông tin cần thiết chỉ trong một hệ thống duy nhất.
- **Phân tích và gợi ý nâng cao:** Trong tương lai, chatbot có thể sử dụng dữ liệu từ ứng dụng dự đoán giá để cung cấp các gợi ý chính xác và phù hợp hơn với nhu cầu và ngân sách của người dùng.

Việc tích hợp ứng dụng dự đoán giá bất động sản và chatbot tư vấn bất động sản giúp tạo ra một công cụ mạnh mẽ và linh hoạt, hỗ trợ tối đa cho người dùng trong việc tìm kiếm, đánh giá và quyết định về bất động sản một cách thông minh và hiệu quả.

Chương 6

Kết luận và mở rộng

6.1 Kết luận

Bằng việc áp dụng các kiến thức và kỹ thuật về Khoa học Máy tính vào việc nghiên cứu thị trường bất động sản, nghiên cứu này đã thu được những kết quả khả quan. Bằng cách thử nghiệm bài toán trên cả mô hình ML và DL cho chúng tôi thấy được mô hình DL cho kết quả thí nghiệm vượt bậc trong việc dự đoán đúng giá bất động sản. Bên cạnh đó nhờ kết quả vượt bậc của mô hình DL mang lại giúp chúng tôi xây dựng được một phần mềm dự đoán giá bất động sản một cách chính xác và hiệu quả trong quá trình thử nghiệm phần mềm.

Ngoài ra, hệ thống chatbot được chúng tôi xây dựng giúp tư vấn với những kiến thức chủ yếu thuộc lĩnh vực bất động sản cũng cho ra được những kết quả khả quan, trong tương lai có thể giúp người dùng có những trải nghiệm mới mẻ hơn và năm bắt được thông tin mọi lúc mọi nơi hơn

6.2 Mở rộng và hướng phát triển của đồ án trong tương lai

Đối với đề tài này, chúng tôi dự định mở rộng nghiên cứu về phương diện phương pháp và phương diện ứng dụng.

Về mặt phương pháp và ứng dụng, chúng tôi sẽ tiếp tục tiến hành thí nghiệm để nâng cao độ chính xác và chất lượng của mô hình. Nhằm hạn chế sự sai sót trong quá trình xây dựng phần mềm dự đoán giá nhà. Ngoài ra, chúng tôi cũng tiếp tục nghiên cứu xây dựng, hoàn thiện hơn về mặt đánh giá cũng như nâng cao chất lượng về câu trả lời của ứng dụng chatbot tư vấn của trong lĩnh vực bất động sản ở thị trường Thành phố Hồ Chí Minh. Trong

tương lai chúng tôi sẽ mở rộng thêm về phạm vi trả lời các câu hỏi cũng như là phân tích dự đoán giá lên quy mô rộng hơn.

Tài liệu tham khảo

[1] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[9] W. Hoeffding. A non-parametric test of independence. *The Annals of Mathematical Statistics*, 19(4):546–557, 1948.

[10] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An introduction to statistical learning. *Springer Science & Business Media*, 2013.

[11] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[12] Bengio Y. & Hinton G. LeCun, Y. Deep learning. *Nature*, 521(7553):436–444, 2015.

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2020.

[14] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, , et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.

[15] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI preprint*, 2018.

[16] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[17] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[18] Gerard Salton and Michael J. McGill. *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., 1989.