TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO NGHIÊN CỨU KHOA HỌC

XÂY DỰNG HỆ THỐNG DỰ ĐOÁN GIÁ NHÀ BẰNG THUẬT TOÁN HỒI QUY

Sinh viên thực hiện: Nguyễn Kim Tuấn **Nam, Nữ: Nam**

Dân tộc: Kinh

Lớp, khoa: Công Nghệ Thông Tin 6 – Công Nghệ Thông Tin Năm thứ: 3/4

Ngành học: Công Nghệ Thông Tin

Sinh viên thực hiện: Nguyễn Thảo Linh Nam, Nữ: Nữ

Dân tộc: Kinh

Lớp, khoa: Công Nghệ Thông Tin 1 – Công Nghệ Thông Tin Năm thứ: 3/4

Ngành học: Công Nghệ Thông Tin

Người hướng dẫn: Th.s. Ngô Thị Thanh Hòa

Hà Nội, năm 2025

LỜI CẢM ƠN

Trước khi báo cáo được trình bày, chúng em xin được gửi lời cảm ơn chân thành đến cô - giảng viên khoa Công nghệ Thông tin, Trường Công nghệ thông tin và truyền thông.

Cô đã hướng dẫn chúng em trong quá trình thực hiện báo cáo nghiên cứu khoa học với chủ đề "xây dựng hệ thống dự đoán giá nhà bằng thuật toán hồi quy". Cô đã truyền đạt cho chúng em những kiến thức về chuyên môn cũng như các kiến thức xoay quanh.

Cô cũng luôn sẵn sàng trả lời các thắc mắc của chúng em, hỗ trợ chúng em trong việc giải quyết các vấn đề phát sinh trong quá trình thực hiện báo cáo nghiên cứu khoa học. Nhờ có sự hướng dẫn và hỗ trợ tận tình của cô, chúng em đã hiểu sâu hơn về mô hình hồi quy tuyến tính và có thể áp dụng kiến thức đó vào thực tiễn trong tương lai.

Chúng em xin chân thành cảm ơn cô đã tận tình hướng dẫn và giúp đỡ chúng em trong quá trình học tập và hy vọng sẽ có cơ hội được học tập và nghiên cứu thêm từ cô trong tương lai. Chúng em hiểu rằng trong quá trình hoàn thành bài báo cáo nghiên cứu khoa học, không tránh khỏi thiếu sót và chúng em sẵn sàng chấp nhận mọi góp ý, phản hồi từ thầy để hoàn thiện bài báo cáo của mình.

Kính chúc cô sức khỏe, hạnh phúc và thành công trong sự nghiệp truyền đạt kiến thức cho thế hệ trẻ!

Trân trọng cảm ơn cô!

Nhóm sinh viên

Nguyễn Kim Tuấn

Nguyễn Thảo Linh

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

DANH MỤC KÝ HIỆU VÀ VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1		
2		

LỜI MỞ ĐẦU

Trong bối cảnh phát triển mạnh mẽ của trí tuệ nhân tạo (AI) và học máy (Machine Learning), các thuật toán dự đoán ngày càng đóng vai trò quan trọng trong nhiều lĩnh vực, đặc biệt là trong bất động sản. Giá nhà luôn biến động do nhiều yếu tố như vị trí, diện tích, tiện ích, thị trường kinh tế, và xu hướng đầu tư. Vì vậy, việc xây dựng một hệ thống có khả năng dự đoán giá nhà chính xác sẽ giúp ích rất nhiều cho các cá nhân, doanh nghiệp và nhà đầu tư trong việc đưa ra quyết định mua bán hợp lý. Trong nghiên cứu này, chúng tôi đề xuất một hệ thống dự đoán giá nhà dựa trên thuật toán hồi quy, một phương pháp phổ biến và hiệu quả trong bài toán phân tích dữ liệu.

Trí tuệ nhân tạo đã trở thành một lĩnh vực nghiên cứu quan trọng, giúp máy tính có thể học hỏi từ dữ liệu và đưa ra dự đoán chính xác mà không cần lập trình cứng nhắc. Các ứng dụng của AI trong dự đoán, đặc biệt là trong tài chính, y tế và bất động sản, ngày càng phát triển mạnh mẽ. Trong số các phương pháp dự đoán, hồi quy là một kỹ thuật quan trọng giúp phân tích mối quan hệ giữa các yếu tố đầu vào và biến mục tiêu. Với tính toán đơn giản nhưng hiệu quả, hồi quy đã được áp dụng rộng rãi trong các mô hình dự báo.

Trong nghiên cứu này, chúng tôi tập trung vào việc xây dựng một hệ thống dự đoán giá nhà bằng thuật toán hồi quy. Hệ thống này sẽ sử dụng các yếu tố như diện tích, số phòng ngủ, vị trí và các đặc điểm khác để phân tích và dự đoán mức giá phù hợp. Bằng cách ứng dụng hồi quy tuyến tính và hồi quy phi tuyến, chúng tôi sẽ đánh giá hiệu suất của các mô hình, từ đó lựa chọn phương pháp tối ưu nhất cho bài toán.

Nội dung nghiên cứu được chia thành ba phần chính:

Chương 1: Trình bày tổng quan về trí tuệ nhân tạo, lịch sử lâu đời và ứng dụng trong một số lĩnh vực. Chương này giúp làm rõ vai trò của AI trong cách các mô hình dự đoán có thể cải thiện tính chính xác của thị trường bất động sản.

Chương 2: Giới thiệu về thuật toán hồi quy, bao gồm các thuật toán hồi quy tuyến tính và hồi quy logistic, đồng thời chọn ra thuật toán phù hợp với dự đoán giá nhà.

Chương 3: Thực nghiệm và xây dựng hệ thống, bao gồm thu thập dữ liệu, tiền xử lý, triển khai mô hình, đánh giá kết quả và đề xuất cải tiến trong tương lai.

Nghiên cứu này không chỉ cung cấp một hệ thống có khả năng dự đoán giá nhà mà còn mở ra cơ hội áp dụng các thuật toán học máy vào thực tiễn. Bằng cách kết hợp dữ liệu thực tế và các thuật toán tối ưu, nghiên cứu sẽ giúp cải thiện độ chính xác của dự báo, hỗ trợ người mua, người bán và các nhà đầu tư đưa ra quyết định thông minh hơn trên thị trường bất động sản.

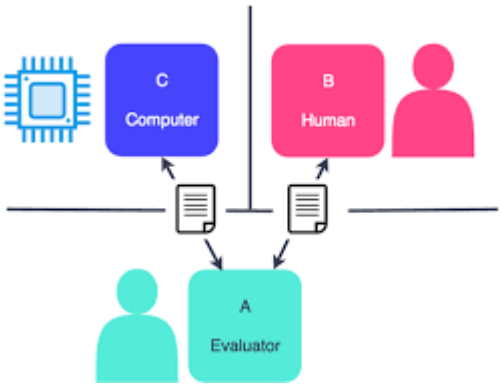
CHƯƠNG 1: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO VÀ DỰ BÁO GIÁ NHÀ.

1.1. Tổng quan về trí tuệ nhân tạo.

1.1.1. Khái niệm trí tuệ nhân tạo.

Trí tuệ nhân tạo (AI) là khả năng tư duy và học tập của một chương trình máy tính hoặc máy móc. Đây cũng là một lĩnh vực nghiên cứu nhằm giúp máy tính trở nên "thông minh hơn". Máy tính có thể tự hoạt động mà không cần mã hóa bằng lệnh. John McCarthy đã đưa ra khái niệm "trí tuệ nhân tạo" lần đầu tiên vào năm 1955.

Con người có mong muốn tạo ra máy tính có trí thông minh của con người từ nhiều thế kỷ trước. Năm 1950, Alan Turing đưa ra cách hình thức hóa các khái niệm thuật toán và tính toán trên máy Turing – một mô hình máy trừu tượng mô tả bản chất việc xử lý các ký hiệu hình thức hay còn được gọi là phép thử Turing. Theo Turing: "Trí tuệ là những gì có thể đánh giá được thông qua các trắc nghiệm thông minh"

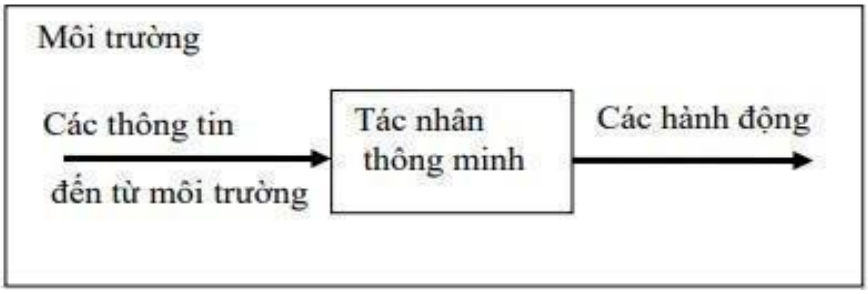


Hình 1.1. Mô tả phép thử Turing.

Phép thử Turing là một cách để trả lời câu hỏi "máy tính có biết nghĩ không?". Alan Turing đề xuất bộ kiểm thử (Turing test): Trong trắc nghiệm

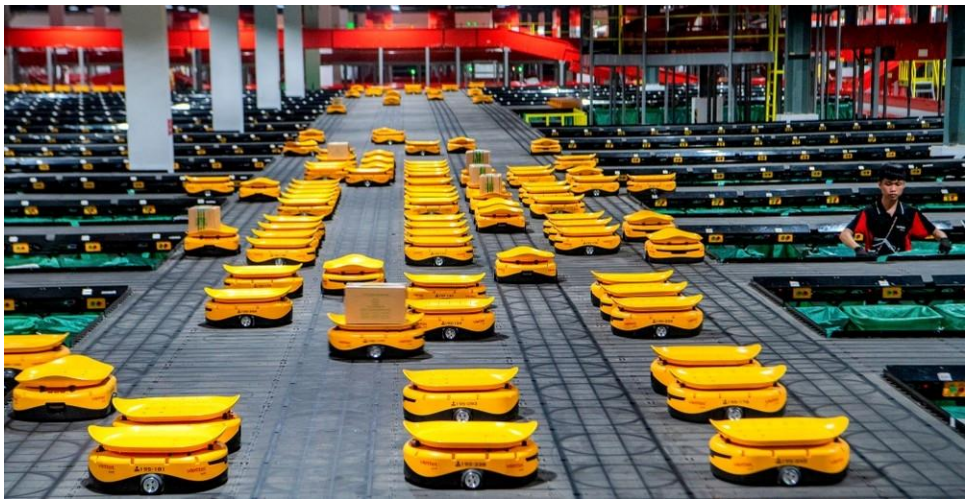
này, một máy tính và một người tham gia trắc nghiệm được đặt vào trong các căn phòng cách biệt với một người thứ hai (người thẩm vấn). Người thẩm vấn không biết được chính xác đối tượng nào là người hay máy tính, và cũng chỉ có thể giao tiếp với hai đối tượng đó thông qua các phương tiện kỹ thuật như một thiết bị soạn thảo văn bản, hay thiết bị đầu cuối. Người thẩm vấn có nhiệm vụ phân biệt người với máy tính bằng cách chỉ dựa trên những câu trả lời của họ đối với những câu hỏi được truyền qua thiết bị liên lạc này. Trong trường hợp nếu người thẩm vấn không thể phân biệt được máy tính với người thì khi đó theo Turing máy tính này có thể được xem là thông minh.

Hiện nay nhiều nhà nghiên cứu quan niệm rằng, TTNT là lĩnh vực nghiên cứu sự thiết kế các tác nhân thông minh (intelligent agent). Tác nhân thông minh là bất cứ cái gì tồn tại trong môi trường và hành động một cách thông minh.



Hình 1.2. Mô hình tác nhân thông minh.

Mục tiêu của AI là tạo ra máy tính có khả năng nhận thức, suy luận và phản ứng giống như con người. Xây dựng TTNT là tìm cách biểu diễn tri thức và phát hiện tri thức từ các thông tin có sẵn để đưa vào trong máy tính. Để máy tính có các khái niệm nhận thức, suy luận, phản ứng thì ta cần phải cung cấp tri thức cho nó.



Hình 1.3. Hình ảnh Robot tự hành và đội ngũ lao động trình độ đại học sẽ thay thế công nhân phân loại hàng tại Viettel Post.

1.1.2. Lịch sử phát triển trí tuệ nhân tạo.

Trí tuệ nhân tạo (AI - Artificial Intelligence) là một trong những lĩnh vực công nghệ phát triển nhanh nhất trong thế kỷ 21, với khả năng thay đổi sâu rộng nhiều ngành nghề và lĩnh vực trong cuộc sống. Tuy nhiên, AI không phải là một khái niệm mới mà đã có lịch sử hình thành và phát triển kéo dài hàng thập kỷ, thậm chí hàng thế kỷ.

Giai đoạn sơ khai – Những ý tưởng nền tảng (Trước năm 1950)

Ý tưởng về các cỗ máy có thể suy nghĩ như con người đã xuất hiện từ thời cổ đại. Trong thần thoại Hy Lạp, Hephaestus – vị thần thợ rèn – đã chế tạo ra những bức tượng kim loại sống động có khả năng suy nghĩ và hành động như con người. Ở Trung Quốc và Ấn Độ cổ đại cũng có những câu chuyện về những bức tượng, hình nhân có thể hoạt động một cách tự động.

Về mặt lý luận, trong thế kỷ 17 - 18, các nhà triết học như René Descartes và Gottfried Wilhelm Leibniz đã suy nghĩ về khả năng mô phỏng tư duy con người bằng những quy tắc logic. Triết gia Thomas Hobbes từng nói

rằng: *"Lập luận chỉ là tính toán"*, một quan điểm sau này có ảnh hưởng lớn đến ngành AI.

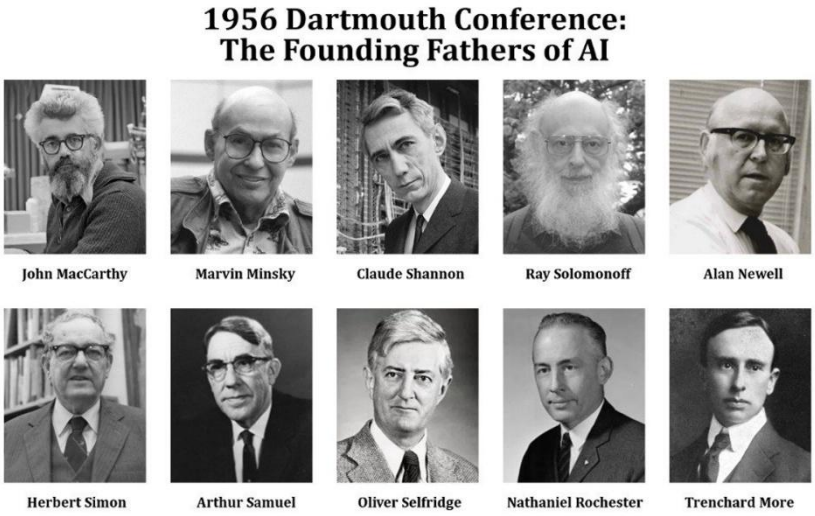
Vào thế kỷ 19, nhà toán học George Boole đã phát minh ra đại số Boolean, một hệ thống toán học sử dụng các giá trị 0 và 1 – đây chính là nền tảng của máy tính hiện đại và cũng là nguyên tắc cốt lõi của AI

Năm 1936, Alan Turing, một trong những nhà tiên phong quan trọng nhất của khoa học máy tính, đã đề xuất "Máy Turing", một mô hình toán học có thể thực hiện bất kỳ phép tính logic nào nếu được cung cấp các tập hợp quy tắc phù hợp. Điều này đặt nền móng cho khái niệm máy tính có thể mô phỏng trí thông minh con người.

Giai đoạn hình thành trí tuệ nhân tạo (1950 - 1970)

Sau Thế chiến thứ Hai, các nhà khoa học bắt đầu suy nghĩ nghiêm túc về việc xây dựng máy móc có khả năng "tư duy". Điển hình như hai sự kiện nổi bật sau:

- *Alan Turing và bài kiểm tra Turing (1950):* Năm 1950, Alan Turing công bố bài báo *"Computing Machinery and Intelligence"*, trong đó ông đề xuất một phương pháp để đánh giá xem liệu một cỗ máy có thể được xem là "thông minh" hay không. Đây chính là "Bài kiểm tra Turing" (Turing Test), một nguyên tắc vẫn còn ảnh hưởng đến AI ngày nay.
- *Hội nghị Dartmouth (1956) – Cột mốc khai sinh AI:* Năm 1956, tại Hội nghị Dartmouth, thuật ngữ "Artificial Intelligence" (Trí tuệ nhân tạo) chính thức được đặt ra bởi nhà khoa học John McCarthy. Hội nghị này quy tụ nhiều tên tuổi lớn như Marvin Minsky, Allen Newell, Herbert Simon... và đặt nền móng cho AI như một ngành nghiên cứu độc lập.



Hình 1.4: Những người tham gia hội nghị Dartmouth năm 1956

Cũng trong giai đoạn này, các nhà khoa học phát triển những chương trình AI đầu tiên, tập trung vào việc giải quyết vấn đề toán học và trò chơi:

- Logic Theorist (1955 - 1956): Đây được xem là chương trình AI đầu tiên, do Allen Newell và Herbert Simon phát triển, có thể chứng minh các định lý toán học.
- General Problem Solver (1957): Một chương trình máy tính có khả năng giải quyết nhiều loại bài toán logic khác nhau.
- Chương trình chơi cờ của IBM (1956 - 1959): Có thể đánh bại những người chơi nghiệp dư.

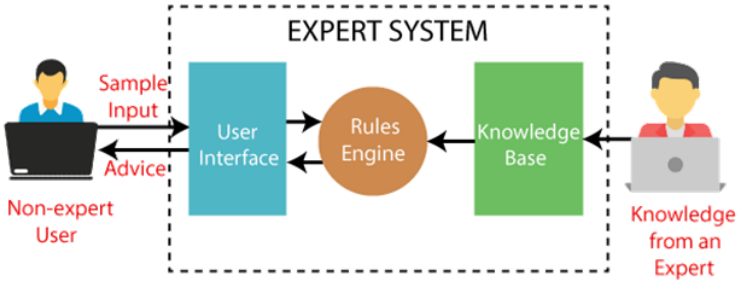
Mặc dù có những thành tựu ban đầu, nhưng AI thời kỳ này vẫn gặp hạn chế do khả năng tính toán yếu của máy tính.

Thời kỳ hoàng kim và mùa đông AI (1970 - 1990)

- *Sự phát triển mạnh mẽ của AI (1970 - 1980):* Trong những năm 1970 và 1980, AI có một bước tiến lớn với sự ra đời của các hệ thống chuyên gia

(Expert Systems). Đây là các chương trình có thể mô phỏng khả năng suy luận của con người trong các lĩnh vực cụ thể, có thể kể đến:

- DENDRAL: Phân tích hóa học.
- MYCIN: Chẩn đoán y khoa và đề xuất phương pháp điều trị.



Hình 1.5. Sơ đồ khối thể hiện hoạt động của một Expert Systems

Hệ thống chuyên gia trở thành một xu hướng mạnh mẽ, giúp AI có nhiều ứng dụng thực tế hơn.

- *Mùa đông AI (1987 - 1993):* Tuy nhiên, vào cuối những năm 1980, AI bắt đầu chững lại do:
  - Chi phí phát triển cao: Hệ thống AI đòi hỏi tài nguyên tính toán lớn nhưng phần cứng lúc đó chưa đủ mạnh.
  - Kỳ vọng quá cao: Giới nghiên cứu và chính phủ đã đặt ra những kỳ vọng quá lớn, nhưng AI chưa thể đáp ứng được.
  - Sự cạnh tranh từ các thuật toán truyền thống: Nhiều bài toán có thể được giải quyết bằng các phương pháp toán học đơn giản hơn thay vì dùng AI.

  Hệ quả là các khoản tài trợ cho AI bị cắt giảm mạnh, kéo theo sự sụp đổ của nhiều dự án nghiên cứu.

- *Sự phục hồi và phát triển (1990 - 2010):* Cuối những năm 1990, AI dần quay trở lại với sự phát triển của học máy (Machine Learning) và sự gia tăng mạnh mẽ của dữ liệu số.
    - 1997: IBM Deep Blue đánh bại Garry Kasparov – một cột mốc quan trọng, chứng minh máy tính có thể vượt trội con người trong các trò chơi chiến lược.
    - 2000s: Sự trỗi dậy của dữ liệu lớn (Big Data): AI ngày càng hiệu quả nhờ vào lượng dữ liệu khổng lồ có sẵn để huấn luyện máy học.
- *AI hiện đại (2010 - nay):* Trong thời trong thời gian này AI đã có bước tiến vượt bậc điển hình như:
    - 2011: IBM Watson thắng cuộc thi Jeopardy!, đánh bại hai nhà vô địch con người.
    - 2016: AlphaGo của DeepMind đánh bại kỳ thủ cờ vây hàng đầu thế giới, mở ra kỷ nguyên AI có thể tự học mà không cần lập trình cụ thể.
    - 2020s: Sự bùng nổ của AI trong đời sống với sự xuất hiện của các mô hình ngôn ngữ lớn (LLM) như ChatGPT, Bard, Claude...

Hình 1.6. AlphaGo của DeepMind.

1.1.3. Các lĩnh vực nghiên cứu và ứng dụng cơ bản.

1.1.3.1. Các lĩnh vực nghiên cứu.

Trí tuệ nhân tạo không phải là một ngành khoa học đơn lẻ mà bao gồm nhiều lĩnh vực nghiên cứu khác nhau.

a) Học máy và nhận dạng mẫu (Machine Learning & Pattern Recognition)

Học máy (Machine Learning - ML) là một nhánh của trí tuệ nhân tạo, tập trung vào việc phát triển các thuật toán cho phép máy tính có thể học hỏi từ dữ liệu mà không cần lập trình rõ ràng. Thay vì được cung cấp một tập hợp các quy tắc cứng nhắc, hệ thống AI sử dụng ML sẽ phân tích dữ liệu, tìm ra các mẫu (patterns) và đưa ra dự đoán dựa trên kinh nghiệm học được.

Học máy thường được chia thành ba loại chính:

- Học có giám sát (Supervised Learning): Máy tính học từ các dữ liệu được gán nhãn sẵn. Ví dụ, một hệ thống nhận diện email spam sẽ học từ các email đã được đánh dấu là spam hoặc không spam trước đó.
- Học không giám sát (Unsupervised Learning): Máy tính tự tìm ra các mẫu dữ liệu mà không có nhãn trước. Một ví dụ điển hình là hệ thống AI có thể phân nhóm khách hàng dựa trên hành vi mua sắm mà không có hướng dẫn cụ thể từ con người.
- Học tăng cường (Reinforcement Learning): Máy tính học thông qua thử và sai, được thưởng hoặc bị phạt dựa trên hành động của mình. Đây là công nghệ được áp dụng trong các trò chơi như AlphaGo của DeepMind.

Ứng dụng của học máy và nhận dạng mẫu:

- Nhận diện khuôn mặt trên điện thoại thông minh.
- Phát hiện gian lận tài chính trong các giao dịch ngân hàng.
- Hệ thống gợi ý phim và sản phẩm của Netflix, Amazon.
- Dự đoán xu hướng tiêu dùng của khách hàng trong ngành thương mại điện tử.

Nhờ vào học máy và nhận dạng mẫu, AI có thể dần dần cải thiện khả năng đưa ra quyết định mà không cần sự can thiệp của con người.

b) Học sâu (Deep Learning)

Học sâu là một nhánh mở rộng của học máy, sử dụng mạng nơ-ron nhân tạo (Artificial Neural Networks) để mô phỏng cách hoạt động của não bộ con người. Học sâu cho phép AI phân tích dữ liệu với độ phức tạp cao hơn và tự động trích xuất các đặc trưng quan trọng.

- Mạng nơ-ron nhân tạo gồm nhiều lớp (layers), trong đó mỗi lớp thực hiện một chức năng cụ thể như nhận diện đường viền, hình dạng, màu sắc trong hình ảnh.
- Số lượng lớn dữ liệu giúp hệ thống AI có thể nhận diện được các đặc trưng phức tạp mà con người khó có thể lập trình thủ công.
- Sự phát triển của phần cứng, đặc biệt là GPU giúp học sâu xử lý dữ liệu nhanh hơn, mở ra nhiều ứng dụng mạnh mẽ hơn.

Ứng dụng của học sâu trong cuộc sống:

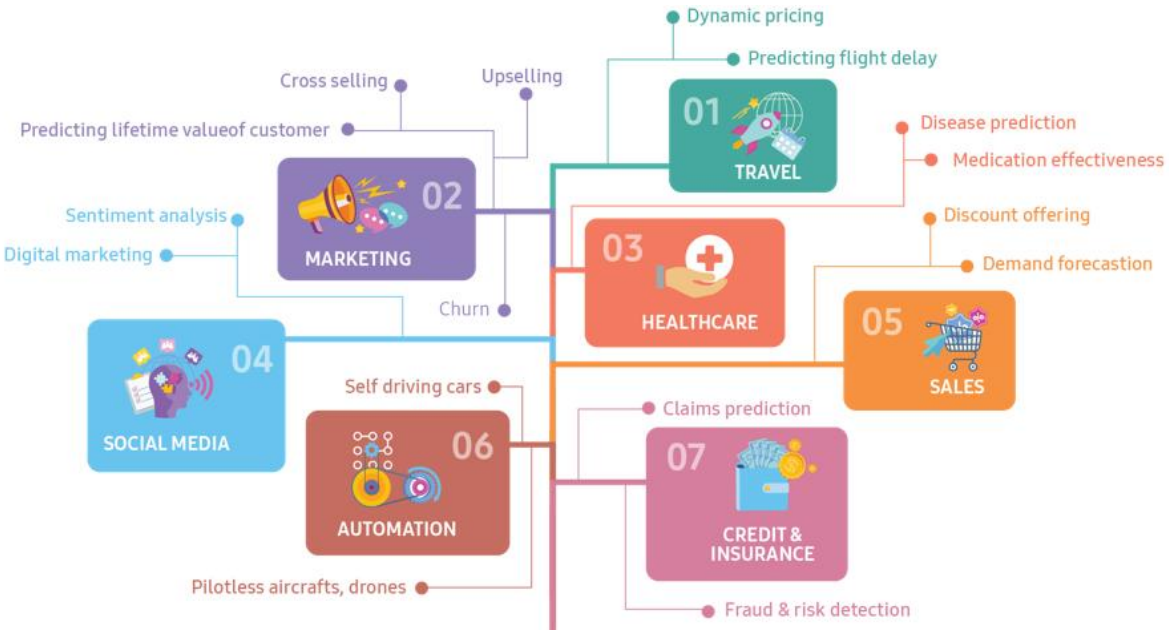
- Xe tự lái: Sử dụng học sâu để nhận diện vật thể, phân tích môi trường xung quanh và đưa ra quyết định lái xe an toàn.

- Chẩn đoán y tế: AI có thể phát hiện ung thư từ ảnh chụp X-quang với độ chính xác cao hơn bác sĩ.
- Chatbot và trợ lý ảo: Các hệ thống như ChatGPT, Siri, Google Assistant có thể hiểu và phản hồi một cách tự nhiên hơn nhờ học sâu.
- Deepfake: Công nghệ giúp tạo ra các video giả mạo bằng cách thay đổi khuôn mặt trong video gốc.

Học sâu đang trở thành một trong những công nghệ cốt lõi giúp AI ngày càng thông minh và ứng dụng rộng rãi hơn trong cuộc sống.

1.1.3.2. Ứng dụng của trí tuệ nhân tạo

Trí tuệ nhân tạo đang thay đổi nhiều ngành công nghiệp và lĩnh vực trong đời sống, từ y tế, tài chính, giáo dục đến giao thông, giải trí. Dưới đây là một số ứng dụng quan trọng của AI.



Hình 1.7 Ứng dụng trí tuệ nhân tạo hiện nay.

a) Thị giác máy tính (Computer Vision)

Thị giác máy tính là lĩnh vực nghiên cứu giúp máy móc có thể "nhìn thấy" và phân tích hình ảnh, video như con người. Công nghệ này sử dụng học máy và học sâu để nhận diện vật thể, phân tích hình ảnh và đưa ra quyết định.

Ứng dụng của thị giác máy tính gồm có:

- Nhận diện khuôn mặt trên điện thoại thông minh và hệ thống an ninh.
- Kiểm tra chất lượng sản phẩm trong nhà máy sản xuất.
- Hỗ trợ chẩn đoán bệnh qua ảnh chụp X-quang, MRI.
- Nhận diện biển số xe trong hệ thống giao thông thông minh.

b) Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP)

Xử lý ngôn ngữ tự nhiên giúp máy tính hiểu, phân tích và giao tiếp bằng ngôn ngữ con người một cách tự nhiên nhất. Sau đây là một số ứng dụng của xử lý ngôn ngữ tự nhiên.

- Dịch thuật tự động (Google Translate, DeepL).
- Phân tích cảm xúc trong bình luận, mạng xã hội.
- Tạo nội dung tự động như bài báo, truyện ngắn, quảng cáo.

c) Nhận dạng giọng nói (Speech Recognition)

Nhận dạng giọng nói giúp AI có thể hiểu và chuyển đổi giọng nói thành văn bản hoặc phản hồi bằng giọng nói. Một số ứng dụng của nhận dạng giọng nói như sau:

- Trợ lý ảo như Siri, Google Assistant, Alexa.
- Chuyển đổi giọng nói thành văn bản cho người khiếm thính.

- Hệ thống tổng đài tự động trong chăm sóc khách hàng.

d) Hệ chuyên gia (Expert Systems)

Hệ chuyên gia mô phỏng suy luận của con người để đưa ra quyết định trong các lĩnh vực chuyên môn. Dưới đây là ứng dụng của hệ chuyên gia:

- Hỗ trợ bác sĩ chẩn đoán bệnh và kê đơn thuốc.
- Dự báo tài chính, quản lý rủi ro đầu tư.
- Tối ưu hóa quy trình sản xuất trong nhà máy.

e) Trò chơi (AI trong Game)

AI giúp tạo ra các nhân vật thông minh hơn, hỗ trợ cá nhân hóa trải nghiệm của người chơi. Một số ứng dụng nổi bật của AI trong game

- Các NPC (nhân vật không phải người chơi) có hành vi linh hoạt hơn.
- AI hỗ trợ người chơi bằng cách đề xuất chiến thuật.
- Sử dụng AI để tổng hợp các phản hồi của người chơi từ đó cải thiện chất lượng game.

f) Người máy (Robotics)

Robot thông minh có thể thực hiện nhiều nhiệm vụ thay thế con người, từ công nghiệp đến dịch vụ. Dưới đây là một số ứng dụng tiêu biểu:

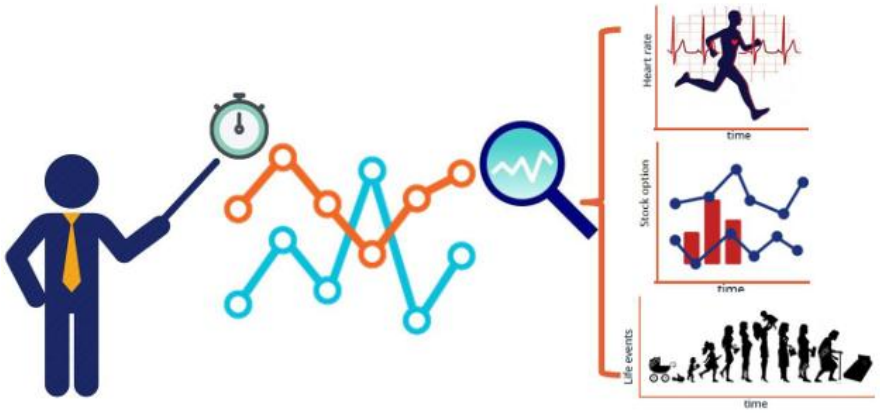
- Robot công nghiệp trong lắp ráp, sản xuất.
- Robot y tế hỗ trợ phẫu thuật.
- Robot dịch vụ trong nhà hàng, khách sạn, y tế.

1.2. Tổng quan dự đoán.

1.2.1. Lịch sử bài toán dự báo.

Bài toán dự đoán (hay dự báo) là một bài toán đã có từ rất lâu và đã phát triển qua nhiều giai đoạn. Dưới đây là ba giai đoạn chính về bài toán dự đoán.

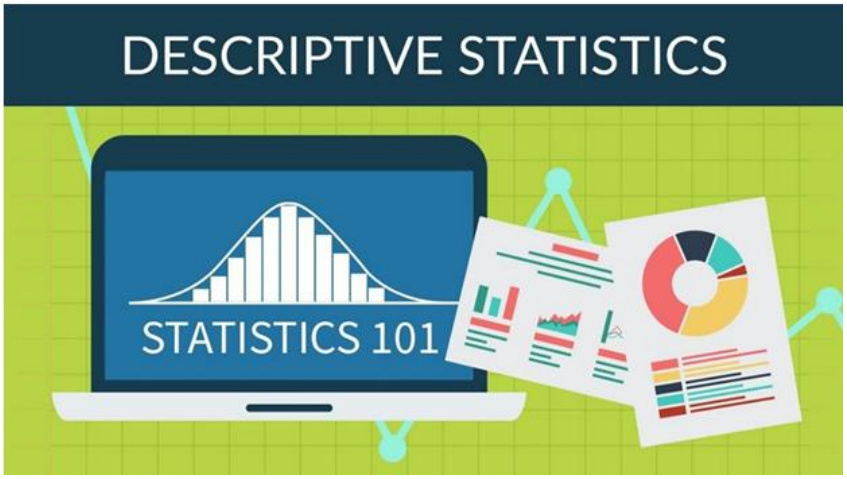
Giai đoạn sơ khai: Phương pháp thống kê truyền thống (Trước 1950 - 1980s). Trong những năm đầu, các phương pháp dự đoán phân tích dữ liệu từ việc tìm ra quy luật toán học dựa vào mô hình thống kê. Trong đó có một số thuật toán nổi bật như:



Hình 1.8: Bài toán dự báo theo chuỗi thời gian.

- Hồi quy tuyến tính (Linear Regression): Mô hình toán học đơn giản để tìm mối quan hệ giữa các biến.
- Mô hình chuỗi thời gian (Time Series Models): Bao gồm ARIMA, MA (Moving Average), AR (AutoRegressive), GARCH.
- Mô hình bình phương tối thiểu (OLS - Ordinary Least Squares): Dự đoán dựa trên dữ liệu quá khứ bằng cách tối ưu hóa sai số bình phương.
- Các phương pháp này có ưu điểm là tính toán nhanh chóng, nhưng bị hạn chế khi dữ liệu có tính phi tuyến hoặc độ phức tạp cao.

Giai đoạn phát triển: Ứng dụng trí tuệ nhân tạo và học máy (1990s - 2010s). Sự bùng nổ của công nghệ máy tính đã giúp các mô hình AI và Machine Learning (ML) bắt đầu được sử dụng để cải thiện độ chính xác của dự đoán.



Hình 1.9: Mô tả bằng biểu đồ

- Một số thuật toán quan trọng xuất hiện:
    - Cây quyết định (Decision Tree) và Random Forest: Dùng để phân loại và dự đoán dữ liệu phi tuyến.
    - Học máy có giám sát (Supervised Learning) và không giám sát (Unsupervised Learning): Giúp máy học từ dữ liệu mà không cần lập trình trực tiếp.
    - Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN): Mô hình dựa trên hệ thần kinh nhân tạo để dự đoán dữ liệu phức tạp hơn.

Các mô hình này giúp dự đoán chính xác hơn so với các phương pháp truyền thống, nhưng vẫn gặp khó khăn khi dữ liệu lớn và tính toán phức tạp.

Giai đoạn hiện đại: Học sâu và dự đoán trên quy mô lớn (2010s - Hiện tại). Sự phát triển của dữ liệu lớn (Big Data), điện toán đám mây (Cloud Computing), phần cứng GPU và TPU đã giúp AI có bước tiến vượt bậc.

Hình 1.10 : Học sâu ứng dụng trong phần mềm chat GPT

- Các mô hình tiên tiến xuất hiện:
    - Học sâu (Deep Learning): Với các mạng nơ-ron phức tạp như CNN, RNN, LSTM, Transformer giúp dự đoán dữ liệu phi cấu trúc như hình ảnh, văn bản, video.
    - Thuật toán tăng cường học (Boosting Algorithms): Như XGBoost, LightGBM giúp tối ưu hóa dự đoán bằng cách kết hợp nhiều mô hình nhỏ.
    - Mô hình dự đoán AI thế hệ mới: GPT, BERT, GAN giúp phân tích và dự đoán dữ liệu văn bản, hình ảnh và thị trường tài chính.

Hiện nay, AI và học máy đã trở thành công cụ quan trọng trong dự đoán tài chính, dự báo khí hậu, y tế, hành vi người tiêu dùng và các lĩnh vực khác.

1.2.2. Tình hình nghiên cứu trong nước.

Ở nước ta hiện nay, bài toán dự đoán được ứng dụng chủ yếu trong kinh tế, tài chính, bất động sản, y tế và khoa học dữ liệu. Các nghiên cứu tập trung vào việc áp dụng các mô hình học máy để phân tích và dự đoán dữ liệu thực tế.

Dự đoán tài chính và kinh tế

- Các nghiên cứu tập trung vào dự đoán chỉ số chứng khoán, tỷ giá hối đoái, lạm phát và tăng trưởng GDP.
- Phương pháp truyền thống như hồi quy tuyến tính, ARIMA, mô hình GARCH vẫn được sử dụng, nhưng gần đây các mô hình học máy như Random Forest, XGBoost, LSTM, Transformer đã được áp dụng để cải thiện độ chính xác.

Dự đoán giá bất động sản

- Các mô hình AI được áp dụng để dự đoán giá nhà đất dựa trên diện tích, vị trí, tiện ích xung quanh.
- Một số doanh nghiệp bất động sản lớn như Vinhomes, Novaland đã ứng dụng AI để phân tích xu hướng thị trường và dự đoán giá trị bất động sản.

Dự đoán trong y tế và khoa học dữ liệu

- AI đã được ứng dụng trong chẩn đoán bệnh tật, dự báo dịch bệnh, phân tích hình ảnh y khoa.
- Một số bệnh viện lớn đã hợp tác với các công ty công nghệ để phát triển hệ thống AI hỗ trợ bác sĩ trong điều trị.

Hạn chế: Thiếu dữ liệu lớn, hạ tầng tính toán còn hạn chế và chưa có nhiều nghiên cứu chuyên sâu như các nước phát triển.

1.2.3. Tình hình nghiên cứu ngoài nước.

Trên thế giới, nghiên cứu về dự đoán đã phát triển mạnh mẽ nhờ vào dữ liệu lớn và tiến bộ trong công nghệ AI. Một số xu hướng nghiên cứu chính bao gồm:

Dự đoán tài chính và thị trường

- AI và học sâu được áp dụng để phân tích thị trường chứng khoán, dự đoán lãi suất, tiền điện tử và biến động kinh tế.
- Các mô hình như LSTM, GAN, Reinforcement Learning được các tập đoàn tài chính lớn như JPMorgan, Goldman Sachs sử dụng.

Dự đoán khí hậu và thời tiết

- Các mô hình AI đã giúp cải thiện dự báo bão, biến đổi khí hậu, nguy cơ thiên tai với độ chính xác cao hơn so với phương pháp truyền thống.
- Google DeepMind, NASA đang nghiên cứu mô hình AI để dự đoán hiện tượng El Niño, mực nước biển dâng.

Dự đoán trong y tế và dược phẩm

- AI giúp phát hiện bệnh sớm, tối ưu hóa điều trị và tìm kiếm thuốc mới.
- IBM Watson, Google Health, OpenAI đang phát triển các mô hình dự đoán ung thư, bệnh tim mạch dựa trên dữ liệu y khoa lớn.

Nhìn chung, nghiên cứu quốc tế đã tiến xa hơn nhờ hạ tầng công nghệ mạnh mẽ, dữ liệu chất lượng cao và đầu tư từ các tập đoàn công nghệ lớn.

1.3. Dự báo giá nhà bằng phương pháp hồi quy.

Dự báo giá nhà là một bài toán kinh điển trong lĩnh vực bất động sản và khoa học dữ liệu. Việc xác định giá trị của một bất động sản có ý nghĩa quan

trọng đối với các cá nhân, doanh nghiệp và tổ chức tài chính. Giá nhà không chỉ phản ánh giá trị vật chất mà còn phản ánh cung - cầu trên thị trường, ảnh hưởng của chính sách kinh tế, sự phát triển đô thị và nhiều yếu tố khác.

1.3.1 Giới thiệu chung

Giá nhà là một biến ngẫu nhiên, do đó, việc dự đoán giá nhà là một vấn đề khó khăn và phức tạp. Có nhiều yếu tố ảnh hưởng đến giá nhà, bao gồm cả các yếu tố nội tại và ngoại sinh. Các yếu tố nội tại bao gồm vị trí, diện tích, tiện ích, chất lượng xây dựng,... Các yếu tố ngoại sinh bao gồm tình hình kinh tế xã hội, chính sách của nhà nước,...

Dự đoán giá nhà bằng thuật toán hồi quy là một phương pháp dựa trên học máy thống kê. Phương pháp này sử dụng dữ liệu lịch sử về giá nhà và các yếu tố ảnh hưởng đến giá nhà để xây dựng một mô hình dự đoán giá nhà. Mô hình dự đoán giá nhà sẽ sử dụng các yếu tố ảnh hưởng đến giá nhà để dự đoán giá nhà trong tương lai.

Các thuật toán hồi quy thường được sử dụng trong dự đoán giá nhà bao gồm:

- Hồi quy tuyến tính: Hồi quy tuyến tính là một thuật toán hồi quy đơn giản nhưng hiệu quả. Hồi quy tuyến tính giả định rằng mối quan hệ giữa giá nhà và các yếu tố ảnh hưởng đến giá nhà là tuyến tính.
- Hồi quy phi tuyến: Hồi quy phi tuyến là một thuật toán hồi quy phức tạp hơn hồi quy tuyến tính. Hồi quy phi tuyến cho phép mối quan hệ giữa giá nhà và các yếu tố ảnh hưởng đến giá nhà là phi tuyến.
- Hồi quy đa biến: Hồi quy đa biến là một thuật toán hồi quy cho phép mô tả mối quan hệ giữa giá nhà và nhiều yếu tố ảnh hưởng đến giá nhà.

Các bước thực hiện dự đoán giá nhà bằng thuật toán hồi quy bao gồm:

- Thu thập dữ liệu: Dữ liệu cần được thu thập một cách đầy đủ và chính xác để xây dựng mô hình dự đoán giá nhà có độ tin cậy cao. Dữ liệu cần bao gồm các thông tin về giá nhà, các yếu tố ảnh hưởng đến giá nhà,...
- Phân tích dữ liệu: Dữ liệu được phân tích để xác định các mối quan hệ giữa giá nhà và các yếu tố ảnh hưởng đến giá nhà.
- Xây dựng mô hình: Mô hình dự đoán giá nhà được xây dựng dựa trên thuật toán hồi quy.
- Đánh giá mô hình: Mô hình dự đoán giá nhà được đánh giá để xác định độ chính xác của mô hình.

Dự đoán giá nhà bằng thuật toán hồi quy là một phương pháp hiệu quả và được sử dụng rộng rãi trong thực tế. Tuy nhiên, phương pháp này cũng có một số hạn chế, chẳng hạn như: phương pháp này phụ thuộc vào chất lượng của dữ liệu đầu vào hay phương pháp này có thể bị ảnh hưởng bởi các biến số ngoại lai.

Để khắc phục các hạn chế này, các nhà nghiên cứu đang tiếp tục phát triển các phương pháp dự đoán giá nhà bằng thuật toán hồi quy mới, hiệu quả hơn.

1.3.2. Một số thách thức

Dưới đây là một số thách thức trong dự đoán giá nhà bằng thuật toán hồi quy:

- Tính đa dạng của dữ liệu: Dữ liệu về giá nhà và các yếu tố ảnh hưởng đến giá nhà có thể rất đa dạng. Điều này có thể khiến việc thu thập, phân tích và xây dựng mô hình dự đoán giá nhà trở nên khó khăn.
- Tính không ổn định của giá nhà: Giá nhà có thể biến động do nhiều yếu tố, chẳng hạn như tình hình kinh tế - xã hội, chính sách của nhà nước,... Điều này có thể khiến mô hình dự đoán giá nhà trở nên kém chính xác.
- Tính phi tuyến của mối quan hệ giữa giá nhà và các yếu tố ảnh hưởng đến giá nhà: Mối quan hệ giữa giá nhà và các yếu tố ảnh hưởng đến giá nhà có thể là phi tuyến. Điều này có thể khiến các thuật toán hồi quy tuyến tính không hiệu quả.

Để giải quyết các thách thức này, các nhà nghiên cứu đang tiếp tục phát triển các phương pháp dự đoán giá nhà bằng thuật toán hồi quy mới, hiệu quả hơn. Các phương pháp này thường sử dụng các kỹ thuật học máy tiên tiến, chẳng hạn như học máy sâu, học máy phi tuyến,...

Kết luận chương 1

Chương 1 đã trình bày về tổng quan về trí tuệ nhân tạo và lịch sử phát triển và ứng dụng của AI. Ở chương này còn cho ta thấy cái nhìn tổng quan về bài toán dự đoán. Từ đó rút ra được thuật toán hồi quy là một phương pháp hiệu quả để dự đoán giá nhà, nhưng vẫn gặp thách thức do dữ liệu đa dạng, tính không ổn định của thị trường và mối quan hệ phi tuyến giữa các yếu tố, đòi hỏi các phương pháp cải tiến để nâng cao độ chính xác.

CHƯƠNG 2: TỔNG QUAN VỀ THUẬT TOÁN HỒI QUY.

2.1. Giới thiệu về thuật toán hồi quy.

Thuật toán hồi quy (Regression Algorithm) là một phương pháp quan trọng trong học máy (Machine Learning) và thống kê (Statistics), được sử dụng để mô hình hóa mối quan hệ giữa một hoặc nhiều biến độc lập (biến đầu vào - X) với một biến phụ thuộc (biến đầu ra - Y).

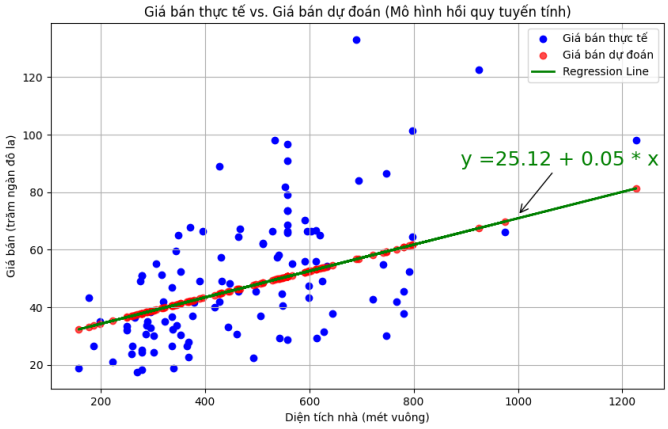
Mục tiêu chính của hồi quy là tìm ra một mô hình toán học có thể dự đoán giá trị của biến phụ thuộc dựa trên biến độc lập. Mô hình hồi quy hoạt động bằng cách xác định các hệ số hồi quy (β), mô tả mức độ ảnh hưởng của các biến đầu vào lên biến đầu ra.

2.2. Một số thuật toán hồi quy.

2.2.1. Hồi quy tuyến tính.

2.2.1.1. Định nghĩa hồi quy tuyến tính

Hồi quy tuyến tính là một phương pháp thống kê giúp tìm ra mối quan hệ giữa một hoặc nhiều biến đầu vào (features) và biến mục tiêu (target).



Hình 2.1. Mô hình hồi quy tuyến tính

2.2.1.2. Hồi quy tuyến tính đơn biến (Simple Linear Regression)

Hồi quy tuyến tính đơn biến là mô hình đơn giản nhất, chỉ xem xét một yếu tố duy nhất để dự đoán giá nhà.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Hình 2.2: Công thức toán học hồi quy tuyến tính đơn biến

Trong đó:

- Y là giá nhà (biến mục tiêu).
- X là biến đầu vào (ví dụ: diện tích nhà).
- β_0 là hệ số chặn (Intercept), thể hiện giá trị của Y khi X= 0
- β_1 là hệ số của biến X, cho biết mức độ ảnh hưởng của yếu tố đó lên giá nhà.
- ϵ là sai số của mô hình.

2.2.1.3. Hồi quy tuyến tính đa biến (Multiple Linear Regression)

Trong thực tế, giá nhà phụ thuộc vào nhiều yếu tố cùng lúc, do đó ta cần sử dụng hồi quy tuyến tính đa biến.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon$$

Hình 2.3: Công thức toán học hồi quy tuyến tính đa biến

Trong đó:

- $X_1, X_2, ..., X_n$ là các biến độc lập như diện tích, số phòng, vị trí, v.v.
- $\beta_1, \beta_2, ..., \beta_n$ là các hệ số hồi quy.

2.2.1.4. Ưu điểm và nhược điểm của hồi quy tuyến tính

a) Ưu điểm:

Dễ hiểu và dễ triển khai: Hồi quy tuyến tính là một trong những thuật toán đơn giản và trực quan nhất trong học máy. Mô hình chỉ cần ước lượng một tập hợp các hệ số (coefficients) cho mỗi biến đầu vào, sau đó sử dụng chúng để dự đoán giá trị đầu ra. Người dùng có thể dễ dàng hiểu được mối quan hệ giữa các biến độc lập và biến phụ thuộc.

Tính toán nhanh chóng, phù hợp với dữ liệu nhỏ và vừa: Hồi quy tuyến tính có thời gian huấn luyện nhanh và yêu cầu tài nguyên tính toán thấp, ngay cả với tập dữ liệu lớn. Do đó, nó phù hợp cho các bài toán yêu cầu dự báo nhanh chóng.

- Với dữ liệu nhỏ: Hồi quy tuyến tính có thể xử lý tốt mà không cần đến các thuật toán phức tạp.
- Với dữ liệu vừa và lớn: Mô hình vẫn có thể được áp dụng nếu dữ liệu có cấu trúc phù hợp và quan hệ giữa các biến là tuyến tính.

So với các mô hình phức tạp như mạng nơ-ron nhân tạo hay Random Forest, hồi quy tuyến tính có thể huấn luyện nhanh hơn rất nhiều.

Có thể mở rộng sang hồi quy tuyến tính đa biến: Mặc dù hồi quy tuyến tính đơn biến chỉ xét một yếu tố ảnh hưởng đến giá nhà, nhưng mô hình có thể được mở rộng thành hồi quy tuyến tính đa biến để xét nhiều yếu tố cùng lúc. Điều này giúp mô hình linh hoạt hơn khi áp dụng vào các bài toán thực tế.

Dễ dàng điều chỉnh và cải tiến: Hồi quy tuyến tính có thể được cải tiến bằng cách:

- Chuẩn hóa dữ liệu để giảm ảnh hưởng của các biến có giá trị lớn.
- Thêm các biến mới để phản ánh nhiều yếu tố hơn.

- Áp dụng các phương pháp hồi quy Ridge/Lasso để tránh quá khớp (overfitting).

Mô hình hồi quy tuyến tính có thể dễ dàng được mở rộng và cải thiện mà không cần thay đổi toàn bộ cấu trúc của nó.

b) Nhược điểm

Giả định mối quan hệ tuyến tính giữa các biến: Hồi quy tuyến tính giả định rằng mối quan hệ giữa biến đầu vào (features) và biến đầu ra (target) là tuyến tính. Tuy nhiên, trong thực tế, quan hệ giữa giá nhà và các yếu tố như vị trí, diện tích, tiện ích có thể không hoàn toàn tuyến tính.

Vì vậy, có thể sử dụng hồi quy phi tuyến tính như hồi quy bậc hai (Polynomial Regression), Random Forest, hoặc Deep Learning hoặc biến đổi dữ liệu để mô hình phản ánh thực tế tốt hơn.

Nhạy cảm với dữ liệu ngoại lai (Outliers)

Hồi quy tuyến tính có thể bị ảnh hưởng mạnh bởi các điểm dữ liệu bất thường (outliers). Nếu có một căn nhà có giá cực kỳ cao hoặc cực kỳ thấp so với mặt bằng chung, mô hình có thể bị lệch và dự đoán không chính xác.

Không hoạt động tốt khi có đa cộng tuyến (Multicollinearity)

Hồi quy tuyến tính giả định rằng các biến độc lập (features) không có quan hệ quá chặt chẽ với nhau. Nếu có hiện tượng đa cộng tuyến, mô hình có thể không hoạt động tốt.

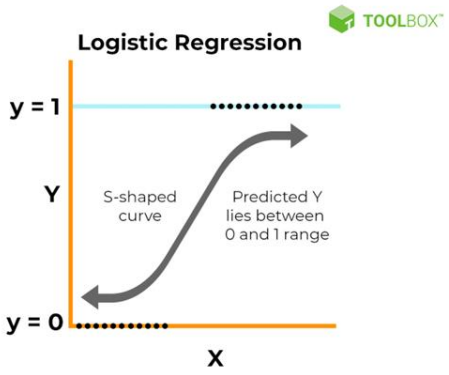
Không thể mô hình hóa các quan hệ phức tạp

Hồi quy tuyến tính không thể xử lý tốt các mối quan hệ phi tuyến tính giữa các biến. Trong khi đó, các phương pháp như Random Forest, XGBoost hoặc mạng nơ-ron nhân tạo có thể mô hình hóa các quan hệ phức tạp tốt hơn.

2.2.2. Hồi quy Logistic.

2.2.2.1. Khái niệm:

Hồi quy Logistic là một kỹ thuật phân tích dữ liệu sử dụng toán học để tìm ra mối quan hệ giữa hai yếu tố dữ liệu. Sau đó, kỹ thuật này sử dụng mối quan hệ đã tìm được để dự đoán giá trị của những yếu tố đó dựa trên yếu tố còn lại. Dự đoán thường cho ra một số kết quả hữu hạn, như có hoặc không. Hồi quy Logistic mang các ưu điểm như sau: mô hình đơn giản, dễ áp dụng, có thể xử lý khối lượng lớn dữ liệu ở tốc độ cao và có khả năng hiển thị cao



Hình 2.4: Hồi quy Logistic

2.2.2.2. Ưu điểm của mô hình hồi quy logistic

- **Tính đơn giản:** Các mô hình hồi quy logistic ít phức tạp về mặt toán học hơn các phương pháp ML khác. Do đó, chúng ta có thể triển khai chúng ngay cả khi đội ngũ không ai có chuyên môn sâu về ML.

- **Tốc độ:** Các mô hình hồi quy logistic có thể xử lý khối lượng lớn dữ liệu ở tốc độ cao bởi chúng cần ít khả năng điện toán hơn, chẳng hạn như bộ

nhớ và sức mạnh xử lý. Điều này khiến các mô hình hồi quy logistic trở nên lý tưởng đối với những tổ chức đang bắt đầu với các dự án ML để đạt được một số thành tựu nhanh chóng.

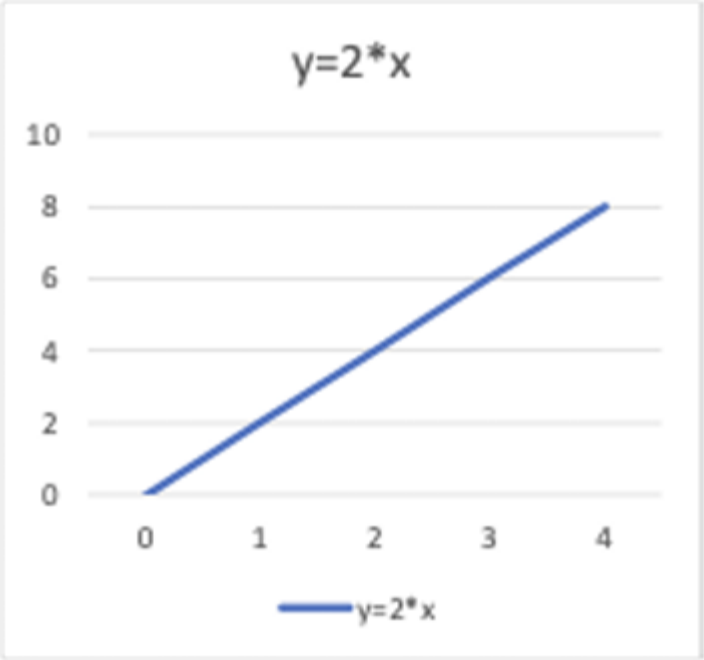
- Sự linh hoạt: Chúng ta có thể sử dụng hồi quy logistic để tìm đáp án cho các câu hỏi có hai hoặc nhiều kết quả hữu hạn. Chúng ta cũng có thể sử dụng phương pháp này để xử lý trước dữ liệu. Ví dụ: có thể sắp xếp dữ liệu với một phạm vi giá trị lớn, chẳng hạn như giao dịch ngân hàng, thành một phạm vi giá trị hữu hạn, nhỏ hơn nhờ hồi quy logistic. Sau đó, chúng ta có thể xử lý tập dữ liệu nhỏ hơn này với các kỹ thuật ML khác để phân tích chính xác hơn.

- Khả năng hiển thị: Phân tích hồi quy logistic cung cấp cho nhà phát triển khả năng nhìn nhận các quy trình phần mềm nội bộ rõ hơn so với các kỹ thuật phân tích dữ liệu khác. Khắc phục sự cố và sửa lỗi cũng trở nên dễ dàng hơn do các phép toán ít phức tạp hơn.

2.2.2.3. Cách thức mô hình hồi quy logistic hoạt động

Một số khái niệm liên quan đến hồi quy logistic:

- Phương trình: Trong toán học, phương trình cho ta mối quan hệ giữa hai biến: x và y. Chúng ta có thể sử dụng các phương trình hoặc hàm này để vẽ đồ thị theo trục x và trục y bằng cách nhập các giá trị khác nhau của x và y. Ví dụ: nếu chúng ta vẽ đồ thị cho hàm y = 2*x, chúng ta sẽ có một đường thẳng như hình dưới đây. Do đó hàm này còn được gọi là hàm tuyến tính.



Hình 2.5: Hàm tuyến tính

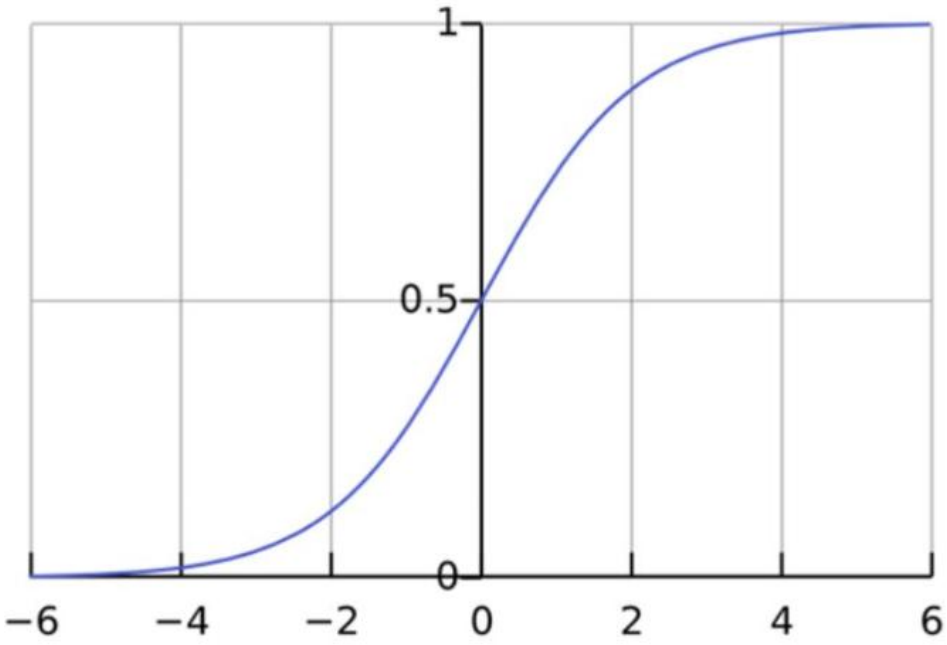
- Biến: Trong thống kê, biến là các yếu tố dữ liệu hoặc thuộc tính có giá trị khác nhau. Bất kỳ phân tích nào cũng có một số biến nhất định là biến độc lập hoặc biến giải thích. Những thuộc tính này là nguyên nhân của một kết quả. Các biến khác là biến phụ thuộc hoặc biến đáp ứng; giá trị của chúng phụ thuộc vào các biến độc lập. Nhìn chung, hồi quy logistic khám phá cách các biến độc lập ảnh hưởng đến một biến phụ thuộc bằng cách xem xét các giá trị dữ liệu lịch sử của cả hai biến.

Trong ví dụ ở trên, X được gọi là biến độc lập, biến dự đoán hoặc biến giải thích vì nó có một giá trị đã xác định. y được gọi là biến phụ thuộc, biến kết quả hoặc biến đáp ứng vì giá trị của nó không xác định.

- Hàm hồi quy logistic: Hồi quy logistic là một mô hình thống kê sử dụng hàm logistic, hay hàm logit trong toán học làm phương trình giữa x và y. Hàm logit ánh xạ y làm hàm sigmoid của x.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Hình 2.6: Phương trình hồi quy logistic



Hình 2.7: Biểu diễn đồ thị của phương trình hồi quy logistic

Chúng ta thấy, hàm logit chỉ trả về các giá trị giữa 0 và 1 cho biến phụ thuộc, dù giá trị của biến độc lập là gì. Đây là cách hồi quy logistic ước tính giá trị của biến phụ thuộc. Phương pháp hồi quy logistic cũng lập mô hình phương trình giữa nhiều biến độc lập và một biến phụ thuộc.

- Phân tích hồi quy logistic đa biến:

Trong nhiều trường hợp, nhiều biến giải thích ảnh hưởng đến giá trị của biến phụ thuộc. Để lập mô hình các tập dữ liệu đầu vào như vậy, công thức hồi quy logistic phải giả định mối quan hệ tuyến tính giữa các biến độc lập khác

nhau. Bạn có thể sửa đổi hàm sigmoid và tính toán biến đầu ra cuối cùng như sau:

$$y = f(\beta 0 + \beta 1x1 + \beta 2x2 + \ldots \beta nxn)$$

Ký hiệu β đại diện cho hệ số hồi quy. Mô hình logit có thể đảo ngược tính toán các giá trị hệ số này khi bạn cho nó một tập dữ liệu thực nghiệm đủ lớn có các giá trị đã xác định của cả hai biến phụ thuộc và biến độc lập.

2.2.3. Lựa chọn thuật toán.

Bảng 2.1. So sánh thuật toán Hồi quy tuyến tính và hồi quy logistic

Thuật toán	Hồi quy tuyến tính	Hồi quy logistic
Mục đích	Dự đoán giá trị liên tục của biến phụ thuộc (Y)	Dự đoán xác suất và phân loại biến phụ thuộc (Y)
Loại bài toán	Hồi quy dự đoán giá trị thực	Phân loại dự đoán xác suất của một nhóm (binary hoặc multi-class)
Dạng đầu ra	Giá trị thực số ($Y \in R$)	Xác suất ($0 \leq P(Y) \leq 1$), sau đó ánh xạ thành nhãn lớp
Phương trình toán học	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$	$P(Y=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}}$
Hàm kích hoạt	Không có – giá trị dự đoán là số thực	Hàm Sigmoid để giới hạn giá trị đầu ra từ 0 đến 1
Hàm mất mát	Sai số trung bình bình phương (MSE - Mean Squared Error)	Entropy chéo (Binary Cross-Entropy hoặc Categorical Cross-Entropy)
Ứng dụng	Khi cần dự đoán giá trị số cụ	hi cần phân loại dữ liệu vào

	thể	nhóm
Mô hình hóa mối quan hệ	Mô hình hóa mối quan hệ tuyến tính giữa biến đầu vào và đầu ra	Mô hình hóa mối quan hệ phi tuyến giữa biến đầu vào và xác suất
Xử lý ngoại lệ	Dễ bị ảnh hưởng bởi giá trị ngoại lệ	Ít bị ảnh hưởng bởi ngoại lệ do đầu ra bị giới hạn trong khoảng (0,1)
Độ chính xác	Phù hợp với dữ liệu có quan hệ tuyến tính	Phù hợp với bài toán phân loại và dữ liệu phi tuyến
Ưu điểm	- Dễ hiểu, dễ triển khai - Dự đoán giá trị liên tục - Phù hợp với dữ liệu tuyến tính	- Giải thích được bằng xác suất - Phù hợp với bài toán phân loại - Không yêu cầu dữ liệu có quan hệ tuyến tính
Nhược điểm	- Không phù hợp với dữ liệu phi tuyến - Dễ bị ảnh hưởng bởi ngoại lệ	- Không thể dự đoán giá trị cụ thể - Cần chuẩn hóa dữ liệu để tránh mất cân bằng lớp
Ví dụ	"Ngôi nhà này có giá khoảng 750 triệu đồng"	"Email này là spam hay không spam?"

Sau khi tổng hợp các đặc trưng của hai thuật toán, chúng em quyết định sử dụng thuật toán hồi quy tuyến tính với đầu ra của dự đoán là một giá trị cụ thể và dự đoán liên tục của biến phụ thuộc.

2.3. Công cụ thực hiện bài toán.

2.3.1. Python

Hình 2.8: Ngôn ngữ lập trình Python

Python là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay, thường được sử dụng để xây dựng trang web và phần mềm, tự động hoá các tác vụ và tiến hành phân tích dữ liệu. Với sự phát triển của khoa học dữ liệu hiện nay, Python lại càng được ứng dụng rộng rãi hơn trong ngành Data Analyst. Với thư viện đa dạng trong các lĩnh vực như khai thác dữ liệu (Scrapy, BeautifulSoup4, …), xử lý dữ liệu và mô hình hóa (Pandas, Scikit-learn, …), trực quan hóa dữ liệu (Matplotlib, Plotly,…) thì đây là một lựa chọn tuyệt vời để phân tích dữ liệu.

Tuy nhiên bên cạnh những ưu điểm về thư viện cũng như cộng đồng lập trình đông đảo, Python vẫn vướng phải một số nhược điểm, đó là bị giới hạn tốc độ, mức tiêu thụ bộ nhớ cao và không phải là một ngôn ngữ được hỗ trợ nhiều cho môi trường di động.

2.3.2. R

Hình 2.9: Ngôn ngữ lập trình R

Ngôn ngữ R là một ngôn ngữ lập trình và môi trường tính toán thống kê phổ biến trong lĩnh vực phân tích dữ liệu và thống kê. Nó cung cấp nền tảng mạnh mẽ cho việc thực hiện các phân tích thống kê, xử lý dữ liệu và tạo biểu đồ. R cũng là một cộng đồng mã nguồn mở lớn, điều này có nghĩa là người dùng có thể dễ dàng chia sẻ mã nguồn, gói phân tích và kiến thức với nhau. Vậy nên việc phân tích dữ liệu trên R cũng rất thuận tiện khi có đầy đủ các thư viện về phân tích dữ liệu và có khả năng tích hợp tốt với môi trường nghiên cứu khoa học.

Dù vậy, R vẫn có một vài nhược điểm nhất định. Phổ biến trong số đấy là sự phức tạp của ngôn ngữ khi lập trình viên mới bắt đầu tiếp xúc và sử dụng, xử lý dữ liệu lớn không tốt so với nhiều ngôn ngữ khác và hiệu suất không phải lúc nào cũng ổn định.

2.3.3. Lựa chọn công cụ

Cả Python và R đều là hai ngôn ngữ phổ biến được sử dụng cho phân tích dữ liệu và thống kê. Việc lựa chọn sử dụng ngôn ngữ nào phụ thuộc vào

nhiều yếu tố như mục tiêu, kinh nghiệm cá nhân, loại dữ liệu đang làm việc, và các thư viện hỗ trợ cần sử dụng.

Sau đây là bảng so sánh để đưa ra quyết định lựa chọn công cụ phục vụ giải quyết bài toán:

Bảng 2.x: So sánh giữa Python và R

Ngôn ngữ	**Python**	**R**
Ưu điểm	- **Đa năng:** Python không chỉ giới hạn trong phân tích dữ liệu, mà còn có thể sử dụng cho nhiều mục đích khác như phát triển ứng dụng, web, automation, và machine learning. - **Thư viện phong phú:** Có nhiều thư viện mạnh mẽ giúp thực hiện các tác vụ phân tích và xử lý dữ liệu một cách hiệu quả. - **Cộng đồng lớn:** Python có cộng đồng lớn giúp việc chia sẻ, học hỏi dễ dàng hơn.	- **Thống kê chuyên sâu**: R được thiết kế đặc biệt cho thống kê và phân tích dữ liệu, với nhiều gói như dplyr, ggplot2, tidyr, và lubridate giúp thực hiện các tác vụ phân tích chi tiết. - **Biểu đồ phức tạp:** Gói ggplot2 trong R cho phép tạo ra biểu đồ phức tạp và tùy chỉnh một cách dễ dàng.
Nhược điểm	**- Thống kê chuyên sâu:** Mặc dù Python có thư viện thống kê tốt, nhưng R vẫn là lựa chọn	- **Thiếu phổ biến:** R có tính chuyên môn hơn so vớiPython. **- Sử dụng bộ nhớ:** R có xu

	phổ biến hơn trong các nghiên cứu thống kê và phân tích dữ liệu chuyên sâu.	hướng sử dụng nhiều bộ nhớ hơn so với Python. **- Quản lý mã nguồn:** R không thể sử dụng mã nguồn, mở rộng và phân chia mã nguồn dễ dàng như Python. Việc quản lý và tái sử dụng mã có thể trở nên khó khăn hơn khi dự án phát triển.

Sau khi tổng hợp các ưu, nhược điểm của cả hai ngôn ngữ, chúng em quyết định sử dụng ngôn ngữ Python với sự đa năng, cộng đồng lớn và nhiều thư viện hỗ trợ.

Kết luận chương 2

Chương 2 đã trình bày các thuật toán hồi quy, cụ thể là thuật toán hồi quy tuyến tính và thuật toán hồi quy Logistic. Cùng với đó là các công cụ thực hiện bài toán. Đồng thời lựa chọn được thuật toán hồi quy tuyến tính đa biến và ngôn ngữ Python để thực hiện thực nghiệm.

CHƯƠNG 3: TRIỂN KHAI BÀI TOÁN

3.1. Giới thiệu bộ dữ liệu.

Average income of residents.	Average age of households.	Average room size.	Average playground size..	Population	House price
68559.57107	5.682861322	7.009188143	4.09	23086.8005	1059033.558
79248.64245	6.002899808	6.730821019	3.09	40173.07217	1505890.915
61287.06718	5.86588984	8.51272743		36882.1594	1058987.988
63345.24005	7.188236095	5.586728665	3.26	34310.24283	1260616.807
59982.19723	5.040554523	7.839387785	4.23	26354.10947	630943.4893
80175.75416	4.988407758	6.104512439	4.04	26748.42842	1068138.074
64698.46343	6.025335907	8.147759585		60828.24909	1502055.817
78394.33928	6.989779748	6.620477995		36516.35897	1573936.564
59927.66081	5.36212557	6.393120981	2.3	29387.396	798869.5328

Hình 3.1. Một số dòng dữ liệu đầu của bộ dữ liệu

3.2. Tiền xử lý dữ liệu.

3.2.1. Quá trình thu thập dữ liệu.

Sử dụng các thư viện của Python như BeautifulSoup, Cloudscraper để thu thập dữ liệu từ các trang web. Các công cụ này giúp tự động truy cập các trang web, tìm kiếm thông tin từ các phần tử HTML và lưu trữ chúng vào các định dạng như CSV hoặc cơ sở dữ liệu. Ngoài ra sử dụng thêm các thư viện khác như concurrent để tăng luồng xử lý và tăng tốc độ thu thập dữ liệu.

3.2.2. Tóm lược dữ liệu.

- Mô tả thuộc tính và kiểu dữ liệu tương ứng:

```
df = pd.read_csv('data.csv')
df.head(5)
```

	Average income of residents.	Average age of households.	Average room size.	Average playground size..	Population	House price
0	68559.57107	5.682861	7.009188	4.09	23086.80050	1.059034e+06
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06
2	61287.06718	5.865890	8.512727	NaN	36882.15940	1.058988e+06
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05

Hình 3.2: Đọc vào dữ liệu

- Mô tả độ lớn của bộ dữ liệu:

```
[ ]  # Độ lớn dữ liệu
     df.shape

     (5028, 6)
```

Hình 3.3: Độ lớn dữ liệu

-Thực hiện tóm lược dữ liệu với describe:

	Average income of residents.	Average age of households.	Average room size.	Average playground size..	Population	House price
count	4927.000000	4853.000000	4933.000000	4023.000000	4958.000000	5.028000e+03
mean	68539.386347	5.975659	6.990494	3.984893	36142.757236	1.232095e+06
std	10599.611017	0.979464	0.991573	1.226500	9568.855304	3.526324e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61503.652615	5.345859	6.317400	3.150000	29877.354227	9.987560e+05
50%	68606.870050	5.976767	6.990350	4.030000	36143.368790	1.232095e+06
75%	75672.049015	6.623682	7.654179	4.480000	42389.618025	1.470440e+06
max	107701.748400	9.519088	10.759588	6.500000	69575.449460	2.469066e+06

Hình 3.4: Tóm lược dữ liệu

-Thống kê dữ liệu khuyết:

```
# Thống kê dữ liệu khuyết.
df.isnull().sum()
```

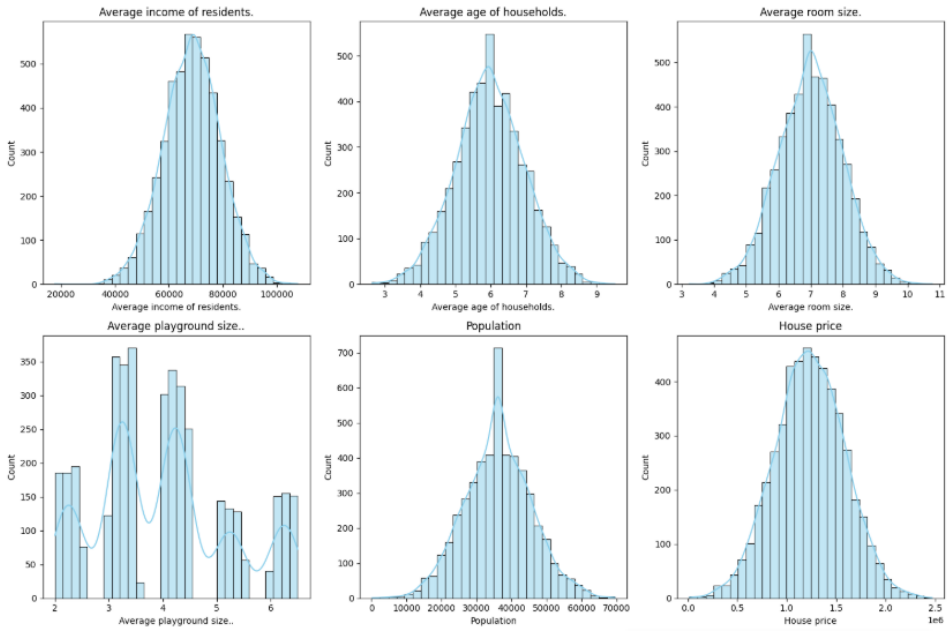
	0
Average income of residents.	101
Average age of households.	175
Average room size.	95
Average playground size..	1005
Population	70
House price	0

dtype: int64

Hình 3.5: Thống kê dữ liệu khuyết

=> Nhận xét: Dữ liệu bị khuyết chủ yếu ở "Average playground size" (1005 giá trị thiếu) và một số cột khác như thu nhập, diện tích phòng, dân số. Có thể xử lý bằng cách loại bỏ dòng thiếu hoặc điền giá trị thay thế (mean/median) để đảm bảo độ chính xác của mô hình.

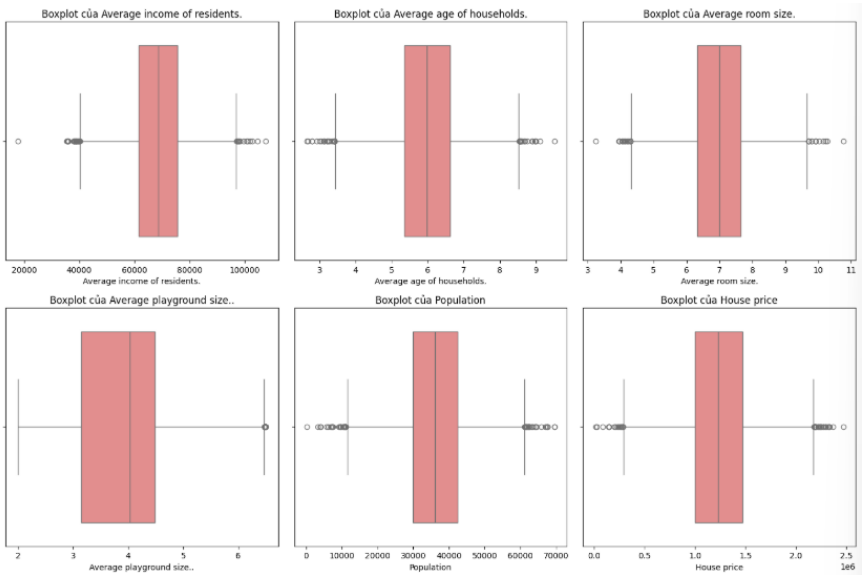
- Biểu đồ histogram với các thuộc tính:



Hình 3.6: Biểu đồ histogram các thuộc tính

=> Thông qua các biểu đồ ta thấy: Hầu hết các biến số có phân phối chuẩn, trừ "Average playground size" có phân phối đa đỉnh, có thể do dữ liệu đến từ nhiều nhóm khác nhau. Biến "Population" có một đỉnh bất thường quanh 40,000, cho thấy một khu vực có dân số cao hơn đáng kể.

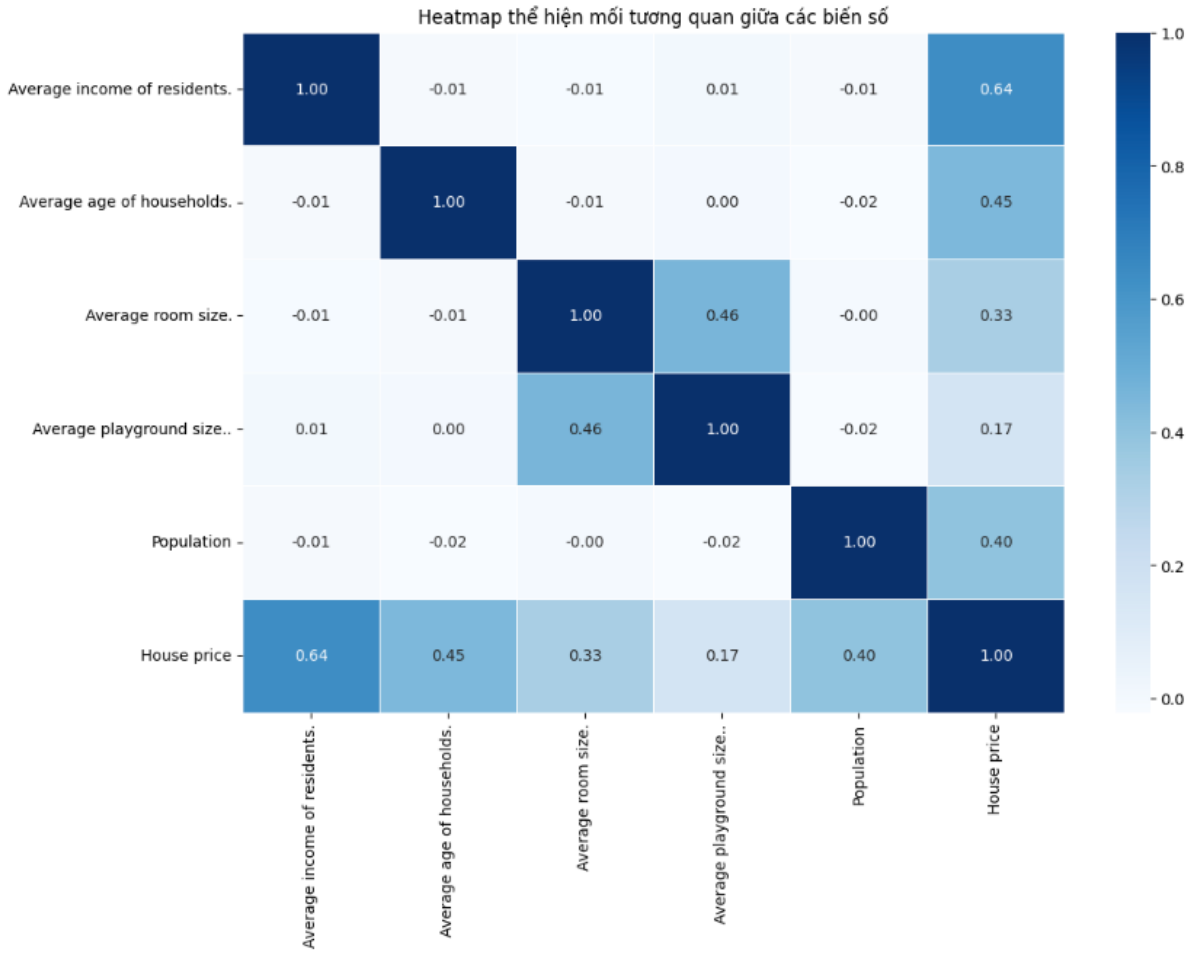
- Xem outliers của các thuộc tính thông qua biểu đồ Boxplot



Hình 3.7: Biểu đồ Boxplot các thuộc tính

=> Thông qua biểu đồ ta thấy: Các boxplot cho thấy hầu hết các biến số có phân phối khá tập trung nhưng đều xuất hiện outliers, đặc biệt là ở phía trên của phạm vi dữ liệu. Biến "Average playground size" và "Population" có nhiều ngoại lệ hơn, có thể cần xử lý outliers để tránh ảnh hưởng đến mô hình.

- Xem độ tương quan giữa các thuộc tính bằng heatmap



Hình 3.8: Biểu đồ heatmap giữa các thuộc tính

=> Thông qua biểu đồ ta thấy: Biểu đồ heatmap cho thấy "House price" có tương quan cao nhất với "Average income of residents" (0.64) và "Average age of households" (0.45), gợi ý rằng thu nhập và độ tuổi hộ gia đình ảnh hưởng đáng kể đến giá nhà. Các biến còn lại có tương quan yếu, ngoại trừ "Average room size" và "Average playground size" (0.46), cho thấy mối liên hệ nhẹ giữa không gian sống và khu vui chơi.

3.2.2. Làm sạch dữ liệu

3.2.2.1. Thực hiện xử lý dữ liệu khuyết

Với dữ liệu liên tục, ta thay thế bằng giá trị trung bình của thuộc tính.

```
# Thực hiện xử lý dữ liệu khuyết:
df['Average income of residents.'].fillna(df['Average income of residents.'].mean(), inplace=True)
df['Average age of households.'].fillna(df['Average age of households.'].mean(), inplace=True)
df['Average room size.'].fillna(df['Average room size.'].mean(), inplace=True)
df['Average playground size..'].fillna(df['Average playground size..'].mean(), inplace=True)
df['Population'].fillna(df['Population'].mean(), inplace=True)
df['House price'].fillna(df['House price'].mean(), inplace=True)
# Thống kê dữ liệu khuyết.
df.isnull().sum()
```

Hình 3.9: Bổ sung dữ liệu khuyết

3.2.2.2. Thực hiện xử lý giá trị ngoại lai

- Cách xác định ngoại lai: mục đích của phương pháp này đưa các điểm nằm ngoài ngưỡng cho phép (do người dùng tự đặt ra) về một giá trị nằm trong khoảng giá trị của 1 phân phối chuẩn. Cụ thể hơn là việc tìm các điểm các điểm với z-score lớn hơn giá trị tuyệt đối của ngưỡng (threshold)
- Cách tính z-score: z-score được tính bằng công thức: $z = (x - \mu) / \sigma$
    - Với X là mảng giá trị của thuộc tính
    - μ là giá trị trung bình của mảng giá trị đó
    - σ là giá trị độ lệch chuẩn với mảng giá trị của thuộc tính tương ứng
- Xử lý giá trị nằm ngoài ngưỡng: các giá trị nằm ngoài ngưỡng sẽ được thay thế bằng một giá trị cụ thể được tính như sau:
    - Với các giá trị nhỏ hơn –threshold thì chúng ta sẽ thay thế một giá trị là mean – threshold * standard deviation. Với mean là giá trị trung bình của mảng giá trị thuộc tính là chúng ta thực hiện thay thế và standard deviation là độ lệch chuẩn tương ứng với mảng giá trị của thuộc tính.
    - Với các giá trị lớn hơn threshold thì chúng ta thay thế bằng một giá trị là mean threshold * standard deviation
- Điều kiện thực hiện: các giá trị trong thuộc tính phải là các giá trị liên tục

- Lý do lựa chọn phương pháp z-score thay vì IQR: z-score giúp duy trì phân phối chuẩn của dữ liệu và giảm thiểu việc loại bỏ các điểm dữ liệu quan trọng. Trong khi đó, phương pháp IQR có thể loại bỏ những điểm có giá trị thực sự, đặc biệt trong trường hợp số lượng ngoại lai lớn. Việc thay đổi các điểm ngoại lai về giá trị cực đại (max) hoặc cực tiểu (min) trong IQR có thể làm biến dạng phân phối dữ liệu, dẫn đến việc mất đi thông tin quan trọng mà những điểm đó mang lại.

```
for column in df.columns:
    # Tính trung bình và độ lệch chuẩn của cột
    mean = df[column].mean()
    std = df[column].std()

    # Tính Z-score
    z_scores = (df[column] - mean) / std

    # Xác định các chỉ số vượt ngưỡng Z-score
    index_lower = np.where(z_scores < -z_threshold)[0]
    index_upper = np.where(z_scores > z_threshold)[0]

    # In số lượng outliers
    print(column + ":", len(index_lower), len(index_upper))

    # Xử lý outliers: Thay thế bằng ngưỡng
    lower_threshold = mean - z_threshold * std
    upper_threshold = mean + z_threshold * std
    df.loc[index_lower, column] = lower_threshold
    df.loc[index_upper, column] = upper_threshold
```

```
Average income of residents.: 0 0
Average age of households.: 0 0
Average room size.: 0 0
Average playground size..: 0 0
Population: 0 0
House price: 0 0
```

Hình 3.10: Xử lý ngoại lai

3.2.4. Kết quả sau khi tiền xử lý làm sạch dữ liệu.

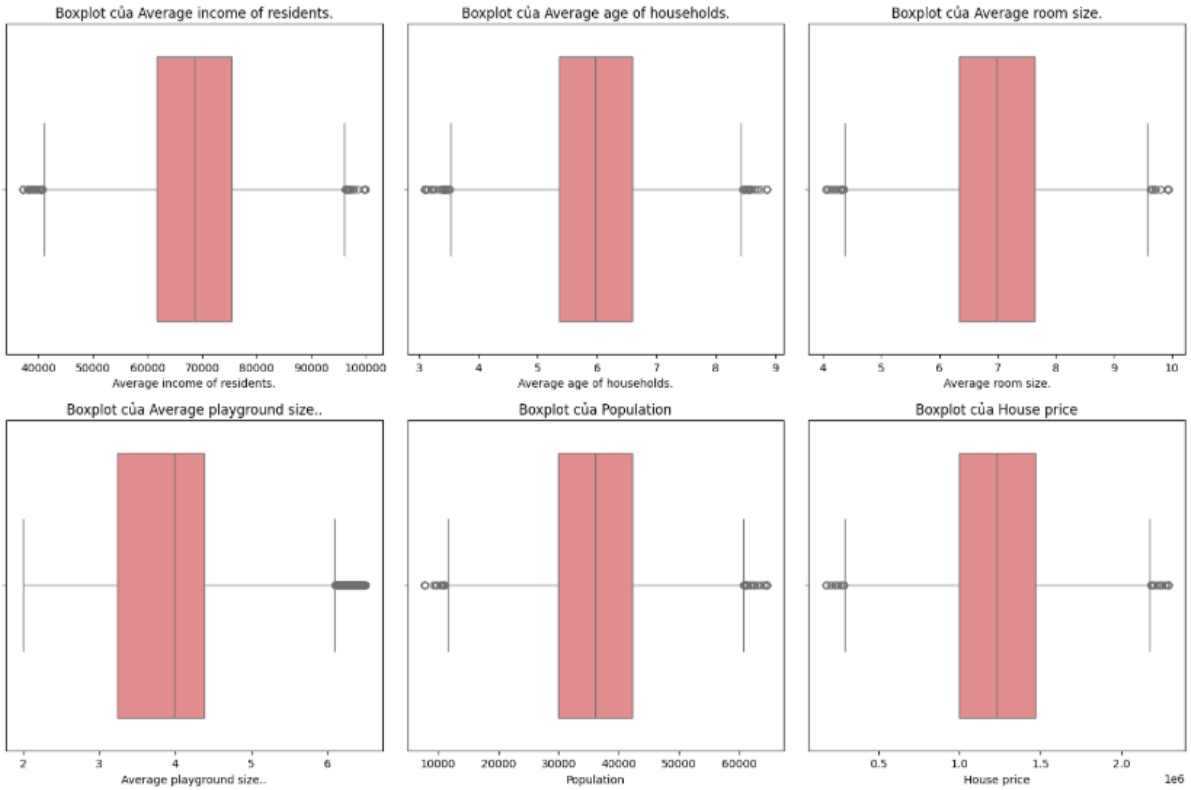
- Dữ liệu sau khi xử lý giá trị khuyết:

	0
Average income of residents.	0
Average age of households.	0
Average room size.	0
Average playground size..	0
Population	0
House price	0

dtype: int64

Hình 3.11: Dữ liệu sau khi xử lý giá trị khuyết

- Dữ liệu sau khi xử lý giá trị ngoại lai:



Hình 3.12: Dữ liệu sau khi xử lý giá trị ngoại lai

3.3. Phân tích hồi quy tuyến tính đa biến và dự đoán.

3.3.1. Phân tích hồi quy đa biến.

3.3.1.1. Thiết lập dữ liệu

- X là tập dữ liệu gồm các đặc trưng độc lập (dự báo).
- Y là biến mục tiêu (House price), biểu diễn giá nhà của từng căn chung cư.

```
# Gán dữ liệu
X = df.drop(columns=['House price'])
Y = df['House price']
```

Hình 3.13: Thiết lập dữ liệu

3.3.1.2. Thiết lập mô hình.

Đoạn mã sau sử dụng thư viện scikit-learn để huấn luyện mô hình hồi quy tuyến tính.

- from sklearn.linear_model import LinearRegression: Nhập lớp LinearRegression từ scikit-learn. Lớp này là một triển khai của mô hình hồi quy tuyến tính.
- model = LinearRegression(): Tạo một đối tượng mô hình hồi quy tuyến tính.
- model.fit(x_train, y_train):Huấn luyện mô hình với dữ liệu huấn luyện (x_train, y_train):
  x_train: Ma trận các biến độc lập
  y_train: Vectơ biến phụ thuộc.

```
# Huấn luyện mô hình hồi quy tuyến tính
model = LinearRegression()
model.fit(X_train, y_train)
```

Hình 3.14: Thiết lập mô hình hồi quy tuyến tính

3.3.1.3. Thiết lập Cross-Validation.

- Chia dữ liệu thành 10 tập con (fold). Mỗi lần lặp, một tập con được dùng làm tập kiểm tra, và 9 tập còn lại làm tập huấn luyện.

- StratifiedKFold đảm bảo tỷ lệ các lớp trong y được giữ nguyên ở từng fold.

```
# Chia dữ liệu thành k tập con
k = 10
kf = KFold(n_splits=k, shuffle=True, random_state=42)
```

Hình 3.15: Cross-Validation với StratifiedKFold

3.3.1.4. Huấn luyện và độ chính xác trong từng fold.

Trong vòng lặp qua từng fold:

- Chia dữ liệu: train_idx và test_idx được sử dụng để tạo tập huấn luyện (X_train, y_train) và kiểm tra (X_test, y_test).
- Huấn luyện mô hình: Huấn luyện mô hình trên tập huấn luyện bằng model.fit(X_train, y_train).
- Dự đoán: Dự đoán nhãn lớp (y_pred) và dữ liệu test(y_test).
- Tính sai số của mô hình

```
# Huấn luyện mô hình trên từng fold
# Danh sách lưu lỗi MSE của từng fold
mse_scores = {}
all_results = []
# Vòng lặp qua từng fold
for fold, (train_index, test_index) in enumerate(kf.split(X), 1):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = Y.iloc[train_index], Y.iloc[test_index]

    # Huấn luyện mô hình hồi quy tuyến tính
    model = LinearRegression()
    model.fit(X_train, y_train)
    # Dự đoán trên tập kiểm tra
    y_pred = model.predict(X_test)
    # Lưu dữ liệu mô hình
    fold_results = pd.DataFrame({'Fold': fold, 'actual': y_test.values, 'Predicted': y_pred.flatten()})
    all_results.append(fold_results)
    # Đánh giá độ chính xác

    # Đánh giá mô hình bằng Mean Squared Error (MSE)
    mse = mean_squared_error(y_test, y_pred)
    mse_scores[fold] = mse
```

Hình 3.16: Huấn luyệ mô hình trên từng fold

Để so sánh giá nhà thực tế và giá nhà của mô hình dự đoán ở thang đo được chuẩn hóa, do vậy chúng ta cần chuyển các giá trị này về thang đo gốc.

```
# In kết quả mô hình trong mỗi lần thử
final_results = pd.concat(all_results, ignore_index=True)
print(final_results)
```

```
      Fold        actual     Predicted
0        1  1.545155e+06  1.456812e+06
1        1  1.556787e+06  1.511657e+06
2        1  1.030591e+06  1.205079e+06
3        1  8.820572e+05  6.919458e+05
4        1  1.186689e+06  1.014627e+06
...    ...           ...           ...
4277    10  1.120943e+06  1.058454e+06
4278    10  1.518478e+06  1.421724e+06
4279    10  1.050224e+06  8.994865e+05
4280    10  1.491812e+06  1.538945e+06
4281    10  1.114902e+06  1.053370e+06

[4282 rows x 3 columns]
```

Hình 3.17: Các giá trị nhà thực tế và dự đoán của mô hình trên từng fold.

Mean Squared Error (MSE) là một phép đo đánh giá hiệu suất của mô hình hồi quy. Nó đo lường sự chênh lệch giữa giá trị dự đoán của mô hình và giá trị thực tế trong dữ liệu kiểm thử.

MSE tính trung bình của bình phương của độ chênh lệch giữa giá trị thực tế và giá trị dự đoán. Nếu MSE là 0, đồng nghĩa với việc mô hình dự đoán hoàn hảo và không có sự chênh lệch nào. Ngược lại, giá trị MSE lớn hơn thể hiện sự chênh lệch lớn giữa giá trị dự đoán và giá trị thực tế. MSE thường được sử dụng trong các bài toán hồi quy để đánh giá chất lượng của mô hình.

```
# Kiểm tra độ chính xác của thuật toán
MSE = pd.DataFrame(list(mse_scores.items()), columns=['Model', 'MSE'])
print(MSE.head(10))
average_mse = np.mean(list(mse_scores.values()))
print(f"Trung bình MSE: {average_mse:.4f}")
```

```
   Model          MSE
0      1  1.703237e+10
1      2  1.642596e+10
2      3  1.709138e+10
3      4  1.541223e+10
4      5  1.595469e+10
5      6  1.461659e+10
6      7  1.665936e+10
7      8  1.514389e+10
8      9  1.456620e+10
9     10  1.464620e+10
Trung bình MSE: 15754886988.4581
```

Hình 3.18: Tính sai số của mô hình trên từng fold

3.3.2. Dự báo.

3.3.2.1 Dự liệu đầu vào.

a) Lấy dữ liệu

```
# Nhập dữ liệu từ người dùng
x1 = input("Nhập thu nhập trung bình của người dân: x1 = ")
x2 = input("Nhập trung bình độ tuổi của người nhà: x2 = ")
x3 = input("Nhập diện tích các phòng: x3 = ")
x4 = input("Nhập diện tích sân: x4 = ")
x5 = input("Nhập dân số tại khu vực đó: x5 = ")
print("Giá trị của đầu vào:")
print (x1, x2, x3, x4, x5)
```

Hình 3.19: Nhập dữ liệu

b) Xử lý dữ liệu

Sau khi lấy được dữ liệu đầu vào, chúng ta cần xử lý dữ liệu đầu vào về kiểu dữ liệu phù hợp với yêu cầu mô hình.

```
# Chuyển dữ liệu nhập thành mảng số
numbers = [x1, x2, x3, x4, x5]
arr_number = np.array(numbers)
arr_number = arr_number.astype(int)
```

Hình 3.20: Chuyển đổi dữ liệu

3.3.2.2. Áp dụng mô hình dự báo.

a) Áp dụng mô hình

```
# Tính giá nhà dự đoán
giaNhaDuDoan_Y = model.predict(np.array(arr_number).reshape(1, -1))
# In kết quả
print("Giá nhà dự đoán là: " + str(giaNhaDuDoan_Y[0]))
```

Hình 3.21: Thự hiện dự đoán và đưa ra kết quả

Nhập lần lượt dữ liệu của căn nhà mà bạn muốn mô hình dự đoán là:

- Thu nhập trung bình của người dân: x1 = 56000
- Trung bình độ tuổi của ngôi nhà: x2 = 5
- Diện tích các phòng: x3 = 56

- Diện tích sân: x4 = 30
- Dân số tại khu vực đó : x5 = 13200

b) Kết quả thu được

Kết quả mô hình dự đoán là: “Giá nhà dự đoán là: 6370539.685455687”.

Như vậy chúng ta đã xây dựng xong mô hình dự báo giá nhà bằng thuật toán hồi quy tuyến tính hoàn chỉnh.

Kết luận chương 3

Chương 3 đã trình bày quy trình thực nghiệm theo các bước thao tác với dữ liệu bằng cách điền khuyết, làm sạch; huấn luyện mô hình hồi quy tuyến tính; dự báo sử dụng mô hình đã huấn luyện.

KẾT LUẬN

Trong nghiên cứu này, chúng tôi đã tập trung vào việc xây dựng một hệ thống dự đoán giá nhà bằng cách ứng dụng thuật toán hồi quy. Với sự phát triển không ngừng của thị trường bất động sản, việc dự đoán giá nhà chính xác đóng vai trò quan trọng trong quá trình ra quyết định của người mua, người bán, cũng như các nhà đầu tư. Do đó, mục tiêu chính của nghiên cứu là phát triển một mô hình dự đoán dựa trên các yếu tố quan trọng như thu nhập trung bình của cư dân, diện tích phòng, mật độ dân số và các thông số liên quan đến bất động sản.

Trong quá trình thực hiện, dữ liệu được thu thập từ nhiều nguồn đáng tin cậy, sau đó được tiền xử lý để loại bỏ giá trị ngoại lai, huấn luyện. Chúng tôi đã thử nghiệm nhiều phương pháp hồi quy, trong đó mô hình hồi quy tuyến tính đa biến cho kết quả khả quan nhất. Việc đánh giá mô hình trên tập dữ liệu kiểm tra cho thấy độ chính xác cao, chứng minh rằng phương pháp được đề xuất có thể ứng dụng hiệu quả trong thực tế.

Tuy nhiên, nghiên cứu này vẫn còn một số hạn chế cần khắc phục. Mặc dù mô hình đã được xây dựng và kiểm chứng, chúng tôi chưa triển khai một giao diện người dùng trực quan để giúp người dùng có thể dễ dàng nhập dữ liệu và nhận kết quả dự đoán một cách trực tiếp.

Trong tương lai, chúng tôi sẽ phát triển một giao diện thân thiện với người dùng, giúp hệ thống trở nên dễ sử dụng hơn, mở rộng khả năng tiếp cận với đối tượng người dùng phổ thông. Việc kết hợp một mô hình dự đoán mạnh mẽ với giao diện trực quan sẽ giúp ứng dụng trở thành một công cụ hữu ích trong lĩnh vực bất động sản, hỗ trợ người mua, người bán, và các nhà đầu tư ra quyết định tốt hơn.

TÀI LIỆU THAM KHẢO

[1] Giáo trình: Trí tuệ nhân tạo-Trường Đại Học Công Nghiệp Hà Nội (Nguyễn Phương Nga-Chủ biên, Trần Hùng Cường)

[2] Hồi quy tuyến tính và Ứng dụng https://www.ibm.com/think/topics/linear-regression

[3] https://meeyland.com/tin-tuc/hoi-quy-tuyen-tinh-la-gi-phan-loai-phuong-trinh-vi-du-va-cac-gia-dinh-378180470

[4] https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/overview

[5] https://github.com/thanhhff/AIVN-Machine-Learning/blob/master/Week%203/Linear-Regression-with-multiple-variables.ipynb