HỌC VIỆN HÀNG KHÔNG VIỆT NAM

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO

KHAI THÁC DỮ LIỆU

ỨNG DỤNG THUẬT TOÁN DỰ ĐOÁN GIÁ NHÀ

Giảng viên hướng dẫn: Trần Anh Tuấn

Sinh viên/ Nhóm sinh viên thực hiện: Nhóm 04……

Mã số sinh viên:…….

Lớp: 010100087302……

TP.Hồ Chí Minh, tháng 03/2025

HỌC VIỆN HÀNG KHÔNG VIỆT NAM

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO

KHAI THÁC DỮ LIỆU

ỨNG DỤNG THUẬT TOÁN DỰ ĐOÁN GIÁ NHÀ

Giảng viên hướng dẫn: Trần Anh Tuấn

Sinh viên/ Nhóm sinh viên thực hiện: Nhóm 04……

Mã số sinh viên:…….

Lớp: 010100087302……

Thành phố Hồ Chí Minh, tháng 03/2025

Danh sách Nhóm:

STT	Họ và tên	MSSV	Lớp	Ghi chú
1	Văn Trương Thùy Dung	2254810302	22ĐHTT06	Nhóm Trưởng
2	Nguyễn Văn Sơn	2254810168	22ĐHTT04	Thành Viên
3	Vũ Thành Đạt	2254810175	22ĐHTT04	Thành Viên
4	Nguyễn Minh Hiếu	2254810172	22ĐHTT04	Thành Viên
5	Văn Thế Tuấn	2254810199	22ĐHTT04	Thành Viên

Cán bộ chấm thi 1 *(ký và ghi rõ họ tên)*	**Cán bộ chấm thi 2** *(ký và ghi rõ họ tên)*
Cán bộ chấm thi phúc khảo 1 *(ký và ghi rõ họ tên)*	**Cán bộ chấm thi phúc khảo 2** *(ký và ghi rõ họ tên)*

DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Ký hiệu, chữ viết tắt	Chữ viết đầy đủ

MỤC LỤC

DANH MỤC HÌNH ẢNH

MỞ ĐẦU

Trong kỷ nguyên công nghệ số, dữ liệu không chỉ đơn thuần là thông tin mà còn là một tài nguyên quan trọng, giúp con người đưa ra những quyết định chính xác và kịp thời. Quá trình phân tích dữ liệu ngày càng trở nên thiết yếu, cho phép khám phá những xu hướng và mối quan hệ tiềm ẩn trong tập dữ liệu. Đồng thời, sự phát triển mạnh mẽ của ngôn ngữ lập trình Python đã mở ra nhiều cơ hội để thực hiện các phân tích này một cách hiệu quả và linh hoạt.

Trước sự biến động không ngừng của thị trường bất động sản, việc dự đoán giá nhà trở thành một yếu tố quan trọng đối với cả người mua và người bán. Đề tài này tập trung vào việc áp dụng thuật toán Random Forest – một mô hình học máy phổ biến – để phân tích dữ liệu và dự đoán giá trị bất động sản. Thông qua việc sử dụng Python, nhóm nghiên cứu mong muốn trình bày quá trình triển khai mô hình Random Forest để xử lý dữ liệu nhà đất, từ đó đưa ra những dự đoán có độ chính xác cao, hỗ trợ người dùng trong việc ra quyết định.

CHƯƠNG 1. GIỚI THIỆU

1.1. Lý do chọn đề tài

Phân tích giá nhà là một chủ đề có tính thực tiễn cao, đóng vai trò quan trọng trong lĩnh vực bất động sản và kinh tế. Việc hiểu rõ các yếu tố ảnh hưởng đến giá nhà giúp nhà đầu tư, người mua và người bán đưa ra quyết định chính xác, tối ưu hóa lợi nhuận và nắm bắt xu hướng thị trường.

Một trong những lý do quan trọng khi lựa chọn đề tài này là phạm vi ứng dụng rộng rãi của nó. Không chỉ giới hạn trong lĩnh vực bất động sản, phân tích giá nhà còn có ý nghĩa lớn đối với tài chính, kinh tế và chính sách công. Việc dự đoán chính xác giá nhà có thể góp phần hỗ trợ phát triển các chính sách kinh tế, đồng thời giúp quản lý tài chính hiệu quả hơn.

Giá nhà chịu tác động từ nhiều yếu tố như diện tích, vị trí, tiện ích xung quanh, tình trạng thị trường và các yếu tố kinh tế - xã hội. Để phân tích và dự đoán giá nhà, cần áp dụng các phương pháp xử lý dữ liệu phức tạp, kết hợp với các thuật toán học máy như mạng nơ-ron nhân tạo và Random Forest. Trong đó, mạng nơ-ron nhân tạo được sử dụng để phân loại nhà theo nhóm giá trị, trong khi Random Forest giúp xác định các quy tắc quan trọng trong tập dữ liệu để hỗ trợ dự đoán.

Bên cạnh đó, sự phát triển của công nghệ và nguồn dữ liệu phong phú đã tạo điều kiện thuận lợi cho việc triển khai các mô hình phân tích giá nhà chính xác hơn. Các dữ liệu liên quan như giá nhà lịch sử, thông tin địa lý, tình hình kinh tế - xã hội và tiện ích xung quanh ngày càng trở nên sẵn có, cho phép áp dụng các kỹ thuật khai thác dữ liệu và trí tuệ nhân tạo để tối ưu hóa dự đoán.

Cuối cùng, đề tài này không chỉ mang tính ứng dụng mà còn mở ra cơ hội nghiên cứu, phát triển các phương pháp mới nhằm nâng cao độ chính xác trong dự đoán giá nhà. Việc khám phá các yếu tố tác động, phân tích xu hướng và đánh giá hiệu suất mô hình không chỉ đóng góp vào lĩnh vực học thuật mà còn hỗ trợ thực tế cho thị trường bất động sản và tài chính.

1.2. Mục tiêu đề tài

Nghiên cứu và phân tích giá nhà không chỉ giúp hiểu rõ các yếu tố tác động đến giá trị bất động sản mà còn cung cấp thông tin quan trọng cho các bên liên quan như nhà đầu tư, người mua, người bán và các tổ chức tài chính. Việc nắm bắt xu hướng thị trường giúp họ đưa ra quyết định thông minh, tối ưu hóa lợi nhuận và giảm thiểu rủi ro khi tham gia vào thị trường bất động sản.

Một trong những mục tiêu chính của đề tài là cung cấp cơ sở khoa học để hỗ trợ quyết định đầu tư. Bằng cách phân tích dữ liệu giá nhà và các yếu tố ảnh hưởng, nhà đầu tư có thể xây dựng chiến lược phù hợp, tận dụng cơ hội và hạn chế tổn thất. Ngoài ra, nghiên cứu này còn giúp dự đoán xu hướng thị trường, hỗ trợ việc hoạch định chính sách kinh tế và quản lý tài chính một cách hiệu quả, từ đó góp phần điều tiết thị trường bất động sản.

Bên cạnh đó, đề tài hướng đến việc phát triển các mô hình dự đoán giá nhà chính xác bằng cách ứng dụng các thuật toán phân tích dữ liệu và học máy. Việc khai thác và xử lý dữ liệu từ nhiều nguồn khác nhau giúp cải thiện độ tin cậy của mô hình, mang lại công cụ hữu ích cho nhà đầu tư, các tổ chức bất động sản và các nhà quản lý thị trường.

Tóm lại, nghiên cứu về phân tích giá nhà không chỉ dừng lại ở việc dự đoán mà còn đóng góp vào việc phát triển các phương pháp khoa học trong lĩnh vực này. Việc tìm hiểu yếu tố ảnh hưởng, đánh giá hiệu suất mô hình và đề xuất phương pháp mới sẽ giúp nâng cao độ chính xác trong dự đoán giá nhà, mang lại giá trị thực tiễn cho thị trường bất động sản và nền kinh tế nói chung.

1.3. Phạm vi đề tài

Đề tài tập trung vào việc ứng dụng thuật toán Random Forest để phân tích dữ liệu và dự đoán giá nhà. Phạm vi nghiên cứu cụ thể bao gồm:

Không gian: Nghiên cứu dựa trên dữ liệu bất động sản của một hoặc nhiều khu vực cụ thể, có thể bao gồm dữ liệu từ các thành phố lớn hoặc khu vực có sự biến động mạnh về giá nhà.

Thời gian: Dữ liệu được thu thập trong khoảng thời gian nhất định, có thể là dữ liệu lịch sử trong vòng 5–10 năm gần đây để đảm bảo tính chính xác của mô hình dự đoán.

Lĩnh vực: Thuộc lĩnh vực khoa học dữ liệu, trí tuệ nhân tạo và bất động sản, tập trung vào phân tích và dự đoán giá nhà bằng cách sử dụng các thuật toán học máy.

Giới hạn nghiên cứu: Đề tài không đi sâu vào các yếu tố vĩ mô như chính sách nhà nước hay tác động của biến động kinh tế toàn cầu đến giá nhà, mà chỉ tập trung vào mô hình dự đoán dựa trên dữ liệu sẵn có.

1.4 Đối tượng nghiên cứu

Dữ liệu bất động sản: Gồm thông tin về giá nhà, vị trí, diện tích, số phòng ngủ, số phòng tắm, tiện ích xung quanh, tình trạng pháp lý, năm xây dựng, v.v.

Thuật toán dự đoán: Nghiên cứu và ứng dụng thuật toán Random Forest, phân tích ưu – nhược điểm của từng phương pháp và so sánh hiệu suất dự đoán.

Các yếu tố ảnh hưởng đến giá nhà: Xác định và phân tích các biến đầu vào quan trọng ảnh hưởng đến giá nhà, từ đó chọn lọc các thuộc tính phù hợp để tối ưu mô hình dự đoán.

Hiệu suất của mô hình: Đánh giá độ chính xác của mô hình thông qua các tiêu chí như độ chính xác (accuracy) và hàm mất mát (binary_crossentropy). Đối với mô hình Neural Network, accuracy được sử dụng để đo lường tỷ lệ dự đoán đúng trên tập dữ liệu kiểm tra, trong khi binary_crossentropy giúp đánh giá mức độ sai số của mô hình trong bài toán phân loại nhị phân.

1.5. Phương pháp nghiên cứu

1.5.1. Phương pháp thu thập thông tin

Thu thập dữ liệu từ nguồn mở: Dữ liệu bất động sản sẽ được lấy từ các nguồn uy tín như Kaggle, Zillow, hoặc từ các trang web chuyên về bất động sản.

Khảo sát thực tế: Nếu có điều kiện, có thể thu thập thêm dữ liệu thực tế từ các công ty bất động sản hoặc khảo sát giá nhà qua các nền tảng trực tuyến.

Nghiên cứu tài liệu: Tham khảo các nghiên cứu trước đây về dự đoán giá nhà và các thuật toán học máy trong lĩnh vực bất động sản.

1.5.2. Phương pháp xử lý thông tin

Tiền xử lý dữ liệu: Trong quá trình xử lý dữ liệu, trước tiên cần thực hiện các bước tiền xử lý như làm sạch dữ liệu, loại bỏ các giá trị bị thiếu hoặc không hợp lệ, đồng thời mã hóa các dữ liệu dạng phân loại như khu vực, loại nhà để mô hình có thể tiếp nhận và xử lý dễ dàng hơn. Tiếp đó, dữ liệu được chuẩn hóa để đưa về cùng một thang đo phù hợp, giúp tăng độ chính xác khi đưa vào mô hình dự đoán.

Phân tích dữ liệu: Sau khi dữ liệu đã được xử lý, bước phân tích dữ liệu được tiến hành nhằm nhận diện các xu hướng quan trọng thông qua phương pháp thống kê mô tả. Việc xây dựng các biểu đồ trực quan giúp làm rõ mối quan hệ giữa các yếu tố như diện tích, vị trí, số phòng và giá nhà, từ đó hỗ trợ trong quá trình lựa chọn biến đầu vào cho mô hình dự đoán.

Xây dựng mô hình dự đoán: Khi đã có dữ liệu đầy đủ, việc xây dựng mô hình dự đoán sẽ được thực hiện bằng cách áp dụng thuật toán Random Forest. Trong giai đoạn này, các tham số mô hình sẽ được tinh chỉnh (Hyperparameter Tuning) nhằm tối ưu hóa hiệu suất dự đoán, đảm bảo mô hình có thể hoạt động tốt trên tập dữ liệu mới.

Đánh giá mô hình: Để đánh giá hiệu quả của mô hình Neural Network, các chỉ số như accuracy và binary_crossentropy sẽ được sử dụng. Độ chính xác (accuracy) cho biết tỷ lệ dự đoán đúng trên tổng số mẫu thử nghiệm, trong khi binary_crossentropy phản ánh mức độ sai số của mô hình trong việc phân loại. Đối với Random Forest, độ chính xác cũng là tiêu chí quan trọng để so sánh hiệu suất giữa các thuật toán và xác định thuật toán nào phù hợp hơn với bài toán dự đoán giá nhà.

1.5.3. Phương pháp thực nghiệm

Chạy thử nghiệm mô hình với tập dữ liệu huấn luyện và kiểm tra.

Thực hiện kiểm thử với các dữ liệu mới để đánh giá độ chính xác của mô hình trong thực tế.

Điều chỉnh thuật toán hoặc bổ sung dữ liệu nếu cần thiết để cải thiện kết quả dự đoán.

1.6 Bố cục đề tài

Chương 2 trình bày về cơ sở lý thuyết liên quan đến các thuật toán dự đoán giá nhà, bao gồm Random Forest và Neural Network. Trước tiên, thuật toán Random Forest được giới thiệu như một phương pháp học máy mạnh mẽ dựa trên mô hình Cây quyết định, giúp giảm thiểu overfitting và nâng cao độ chính xác trong dự đoán. Phần này cũng phân tích cách hoạt động của Random Forest, từ quá trình lấy mẫu ngẫu nhiên, xây dựng cây quyết định đến việc tổng hợp kết quả dự đoán. Tiếp theo, chương này đi sâu vào mạng Neural Network, mô tả cấu trúc mạng với các tầng đầu vào, tầng ẩn và tầng đầu ra, cùng với vai trò của các hàm kích hoạt như ReLU và Sigmoid trong việc tối ưu hóa quá trình huấn luyện. Bên cạnh đó, chương 2 cũng đề cập đến thư viện Keras, một công cụ hỗ trợ quan trọng trong việc xây dựng và huấn luyện mô hình học sâu, giúp tối ưu hóa hiệu suất dự đoán giá nhà.

Chương 3 trình bày quá trình phân tích hệ thống và xây dựng sản phẩm, tập trung vào việc triển khai mô hình dự đoán giá nhà bằng hai thuật toán chính. Phần đầu của chương này phân tích dữ liệu bất động sản, thực hiện các bước tiền xử lý, chuẩn hóa dữ liệu và chia tập huấn luyện – kiểm tra. Tiếp theo, chương này mô tả cách huấn luyện mô hình Random Forest và Neural Network, so sánh hiệu suất giữa hai thuật toán, đồng thời kiểm tra độ chính xác trên tập dữ liệu thực tế. Cuối cùng, chương này trình bày về xây dựng giao diện sản phẩm, trong đó bao gồm giao diện nhập dữ liệu bất động sản, giao diện hiển thị kết quả dự đoán, và biểu đồ trực quan giúp so sánh hiệu suất giữa hai mô hình.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Thuật toán Random Forest:

Random Forest là một phương pháp tổng hợp, hoạt động theo nguyên tắc "chia để trị". Phương pháp này tạo ra nhiều cây quyết định trên các tập dữ liệu được chọn ngẫu nhiên và sau đó tổng hợp kết quả để đưa ra dự đoán chính xác hơn. Do đó, tập hợp các cây quyết định này được gọi là một "rừng" (Forest).

Mỗi cây trong rừng được xây dựng bằng cách sử dụng các chỉ số đánh giá mức độ quan trọng của thuộc tính, chẳng hạn như tăng thông tin (Information Gain), tỷ lệ tăng (Gain Ratio) và chỉ số Gini (Gini Index). Quá trình tạo cây diễn ra độc lập trên từng mẫu dữ liệu được chọn ngẫu nhiên, giúp giảm thiểu tình trạng overfitting và cải thiện tính tổng quát của mô hình.

Khi áp dụng vào bài toán phân loại, mỗi cây quyết định trong Random Forest sẽ thực hiện bỏ phiếu đa số (Voting), và nhãn phổ biến nhất trong số các cây sẽ được chọn làm kết quả cuối cùng. Điều này giúp thuật toán Random Forest trở thành một công cụ mạnh mẽ, giảm thiểu overfitting và mang lại độ chính xác cao trong bài toán phân loại.

2.1.1 Cách hoạt động của thuật toán Random Forest:

Random Forest là một thuật toán ensemble learning, nghĩa là nó kết hợp nhiều mô hình nhỏ (các cây quyết định) để tạo ra một mô hình mạnh hơn, giúp cải thiện độ chính xác và khả năng tổng quát hóa, nhờ vào:

Lấy mẫu ngẫu nhiên: Thuật toán sử dụng phương pháp Bootstrap Aggregating (Bagging) để lấy mẫu dữ liệu huấn luyện từ tập dữ liệu gốc. Mỗi tập con được chọn ngẫu nhiên, có kích thước bằng tập dữ liệu ban đầu nhưng có thể chứa một số mẫu trùng lặp. Điều này giúp tăng tính đa dạng của dữ liệu đầu vào cho từng cây quyết định.

Dự đoán từ từng cây: Khi mô hình đã được huấn luyện, một dữ liệu mới sẽ được đưa vào từng cây quyết định để dự đoán kết quả. Mỗi cây sẽ đưa ra một kết quả riêng biệt dựa trên cấu trúc của nó.

Tổng hợp kết quả dự đoán: Trong bài toán phân loại, thuật toán Random Forest sử dụng nguyên tắc bỏ phiếu đa số (majority voting) để xác định kết quả dự đoán cuối cùng. Nhờ cơ chế này, thuật toán có khả năng hạn chế overfitting tốt hơn so với một cây quyết định đơn lẻ, đồng thời mang lại độ chính xác cao hơn trong các bài toán phân loại.

2.1.2 Lý do chọn thuật toán Random Forest vào đề tài:

Một trong những lợi thế nổi bật của thuật toán Random Forest là khả năng xử lý khối lượng dữ liệu lớn và đa dạng. Trong lĩnh vực bất động sản, các yếu tố ảnh hưởng đến giá nhà rất phong phú, bao gồm diện tích, vị trí, tiện ích xung quanh và nhiều đặc điểm khác. Với sự biến động cao của dữ liệu, thuật toán này có thể tổng hợp thông tin một cách hiệu quả và giảm thiểu tác động của các biến ít quan trọng, giúp mô hình trở nên ổn định hơn.

Bên cạnh đó, Random Forest có ưu điểm vượt trội trong việc xử lý dữ liệu bị thiếu mà không cần điền giá trị trước. Trong quá trình xây dựng các cây quyết định, thuật toán có khả năng tự động xử lý những giá trị bị thiếu này, giúp đơn giản hóa công đoạn tiền xử lý dữ liệu, điều mà nhiều thuật toán khác yêu cầu thực hiện trước khi huấn luyện mô hình.

Một đặc điểm quan trọng khác của Random Forest là khả năng giảm thiểu tình trạng overfitting – hiện tượng mô hình quá mức ghi nhớ dữ liệu huấn luyện và không thể tổng quát hóa tốt với dữ liệu mới. Nhờ phương pháp Bootstrap Aggregating (Bagging), việc lấy mẫu ngẫu nhiên và chọn đặc trưng ngẫu nhiên khi xây dựng từng cây quyết định giúp thuật toán duy trì tính linh hoạt và ổn định hơn trong quá trình dự đoán.

Nhờ những đặc điểm trên, Random Forest trở thành một lựa chọn lý tưởng cho bài toán phân tích và dự đoán giá nhà. Không chỉ mạnh mẽ về mặt tính toán, thuật toán này còn có khả năng giải thích ý nghĩa của các đặc trưng quan trọng trong dữ liệu, giúp các bên liên quan trong thị trường bất động sản có cơ sở đáng tin cậy để đưa ra quyết định.

2.2. Mạng Nơ-ron:

Mạng nơ-ron nhân tạo (Neural Network) là một tập hợp các thuật toán được thiết kế để xác định và tìm kiếm mối quan hệ giữa các dữ liệu, mô phỏng theo cách thức hoạt động của não bộ con người. Hệ thống này bao gồm nhiều tế bào thần kinh nhân tạo liên kết với

nhau, tạo thành một mạng lưới có khả năng học hỏi và xử lý thông tin tương tự như cách mà bộ não sinh học hoạt động.

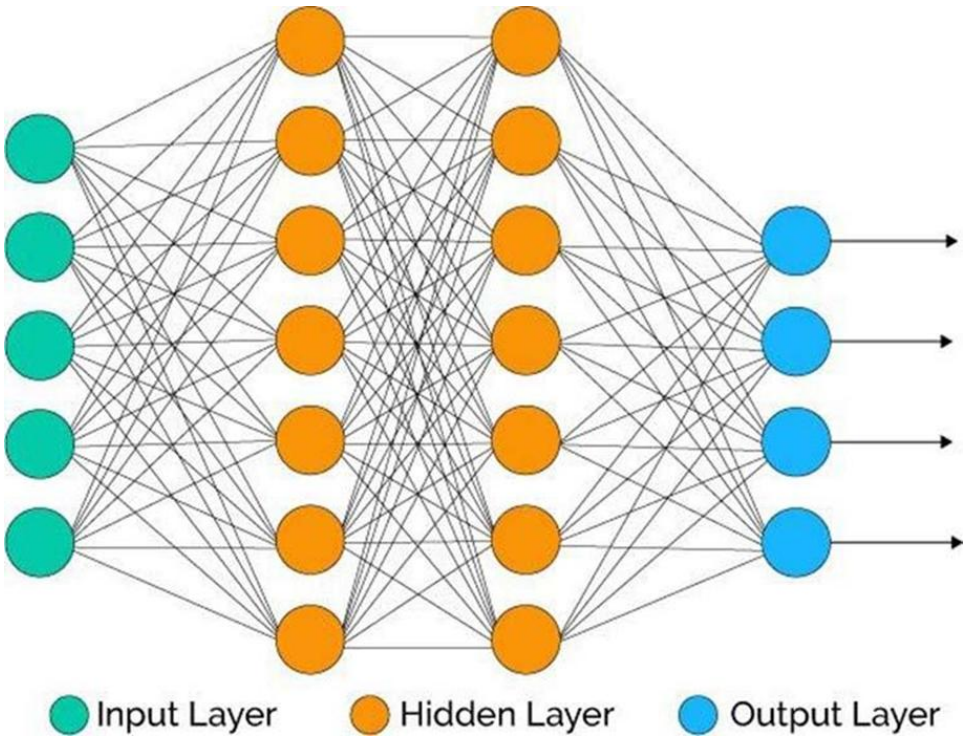
Trong mô hình này, mỗi nơ-ron nhân tạo có vai trò như một hàm toán học, thực hiện nhiệm vụ thu thập, phân tích và phân loại thông tin dựa trên cấu trúc đã được thiết lập. Nhờ đó, mạng nơ-ron có thể tự động học từ dữ liệu đầu vào, phát hiện quy luật tiềm ẩn và đưa ra dự đoán chính xác trong nhiều bài toán phức tạp.

2.2.1 Cấu trúc Mạng Nơ-ron:

Mạng Neural Network được xây dựng từ sự kết hợp của nhiều tầng Perceptron, còn được gọi là Perceptron đa tầng. Một mạng Neural Network thông thường bao gồm ba loại tầng chính.

Đầu tiên là tầng đầu vào (Input Layer), nằm ở phía ngoài cùng bên trái của mạng và đóng vai trò tiếp nhận dữ liệu đầu vào. Tiếp theo là tầng ẩn (Hidden Layer), nằm giữa tầng đầu vào và tầng đầu ra, nơi diễn ra quá trình xử lý dữ liệu và suy luận logic của mô hình. Cuối cùng là tầng đầu ra (Output Layer), nằm ở phía ngoài cùng bên phải, có nhiệm vụ đưa ra kết quả dự đoán sau khi dữ liệu đã được xử lý qua các tầng trước đó.

Sự kết hợp giữa các tầng này giúp Neural Network có khả năng học hỏi, nhận dạng mẫu và đưa ra các dự đoán chính xác dựa trên dữ liệu đầu vào.



Hình 2.1 Cấu trúc Mạng Nơ-ron

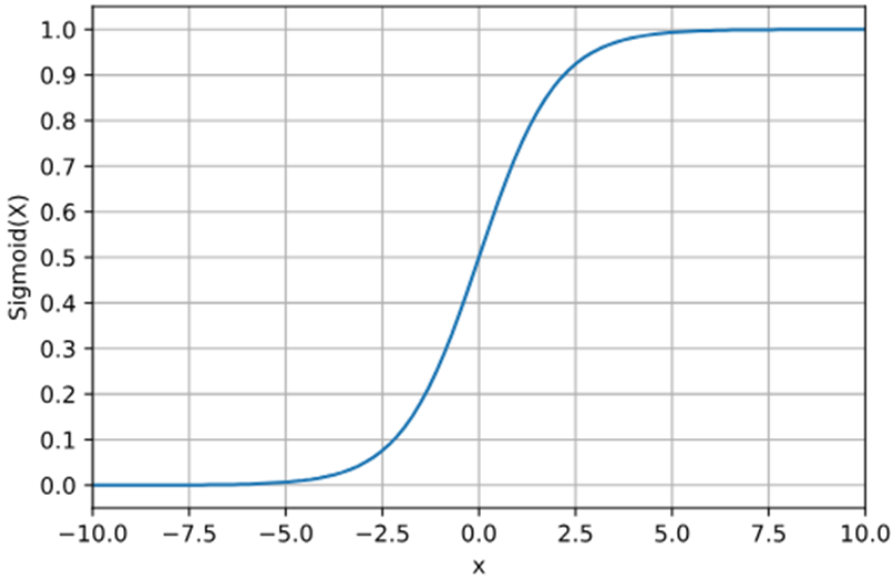
Trong mạng Neural Network, mỗi nút mạng đóng vai trò như một nơ-ron với hàm kích hoạt, trong đó phổ biến nhất là hàm sigmoid. Tuy nhiên, trong thực tế, các nơ-ron thường sử dụng cùng một loại hàm kích hoạt để tối ưu hóa quá trình tính toán.

Số lượng nơ-ron tại mỗi tầng có thể khác nhau tùy thuộc vào đặc điểm của bài toán và phương pháp giải quyết. Đặc biệt, các tầng ẩn thường có số lượng nơ-ron khác nhau, giúp tăng cường khả năng học và trích xuất đặc trưng từ dữ liệu. Ngoài ra, các nơ-ron trong cùng một tầng thường được liên kết với nhau theo mô hình mạng kết nối đầy đủ, đảm bảo thông tin được truyền tải và xử lý hiệu quả. Dựa trên số tầng và số lượng nơ-ron tại mỗi tầng, người dùng có thể tính toán kích thước tổng thể của mạng Neural Network.

2.2.2 Hàm kích hoạt:

Trong mạng nơ-ron nhân tạo, các đơn vị tính toán chuyển đổi giá trị đầu vào bằng một hàm vô hướng, được gọi là hàm kích hoạt. Kết quả đầu ra của hàm này là mức độ kích hoạt của nơ-ron, quyết định tín hiệu truyền sang các nơ-ron tiếp theo trong mạng.

Thông thường, các hàm kích hoạt giới hạn giá trị đầu ra trong một phạm vi xác định, giúp ổn định quá trình học và tránh các vấn đề như giá trị đầu ra quá lớn hoặc quá nhỏ, làm mất thông tin quan trọng. Chính vì vậy, chúng thường được gọi là các hàm bẹp (squashing functions). Một số hàm kích hoạt phổ biến trong mạng nơ-ron bao gồm sigmoid, ReLU (Rectified Linear Unit) và tanh, mỗi loại có đặc điểm riêng và phù hợp với từng loại bài toán cụ thể.



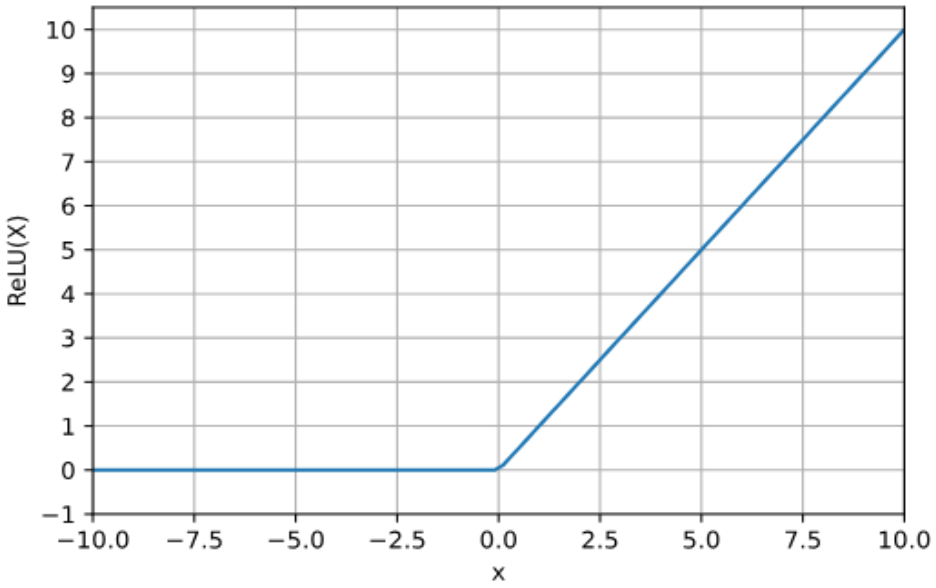
Hình 2.2 Hàm Sigmoid

Hàm Sigmoid nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (0;1) (xem đồ thị phía trên). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp. Tuy nhiên hiện nay hàm Sigmoid rất ít được dùng vì những nhược điểm sau:

Hàm Sigmoid bão hòa và triệt tiêu gradient:

Một nhược điểm dễ nhận thấy là khi đầu vào có trị tuyệt đối lớn (rất âm hoặc rất dương), gradient của hàm số này sẽ rất gần với 0. Điều này đồng nghĩa với việc các hệ số

tương ứng với unit đang xét sẽ gần như không được cập nhật (còn được gọi là vanishing gradient).



Hình 2.3 Hàm ReLU

Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng neuron. ReLU đơn giản lọc các giá trị < 0.

Ưu điểm của hàm Relu là tính đơn giản của nó và nó đã được chứng minh là giúp tăng tốc quá trình training. Tiếp theo là nó không bị chặn như hàm sigmoid nên nó không phải là nguyên nhân gây ra hiện tượng vanishing gradient. Mặc dù vậy thì tại những điểm có giá trị âm thì giá trị của Relu sẽ bằng 0 (dying relu) và theo lý thuyết nó sẽ không có đạo hàm tại các điểm 0 nhưng thực tế thì người ta thường bổ sung thêm đạo hàm của relu tại 0 bằng 0 và bằng thực nghiệm người ta cũng thấy rằng xác suất để input relu rơi vào điểm 0 là rất nhỏ. Và do nó ko được chặn trên nên cũng có một nhược điểm là gây ra hiện tượng exploding gradient nhưng thường sẽ relu sẽ hoạt động tốt trong thực tế.

2.3. Thư viện keras

Keras là một thư viện mã nguồn mở được sử dụng phổ biến trong lĩnh vực học sâu và mạng nơ-ron nhân tạo. Được thiết kế với mục tiêu mang lại sự linh hoạt và dễ sử dụng,

Keras giúp các nhà phát triển xây dựng và thử nghiệm mô hình học sâu mà không cần quan tâm quá nhiều đến các chi tiết kỹ thuật phức tạp.

Một trong những ưu điểm nổi bật của Keras là giao diện lập trình ứng dụng (API) trực quan, giúp đơn giản hóa quá trình phát triển mô hình. Thư viện này còn có khả năng tích hợp với các framework học sâu mạnh mẽ như TensorFlow và Theano, tận dụng sức mạnh tính toán của chúng để tối ưu hóa quá trình huấn luyện mô hình. Nhờ vào tính linh hoạt và hiệu suất cao, Keras trở thành một trong những lựa chọn hàng đầu cho nhiều ứng dụng học sâu, bao gồm phân loại ảnh, dự đoán chuỗi thời gian, xử lý ngôn ngữ tự nhiên và nhiều lĩnh vực khác.

2.3.1 Lợi ích của thư viện keras:

Thư viện Keras mang lại nhiều lợi ích đáng kể trong lĩnh vực học sâu nhờ tính đơn giản, linh hoạt và khả năng tích hợp mạnh mẽ. Trước hết, Keras được thiết kế nhằm đơn giản hóa quá trình xây dựng mô hình, cung cấp một API trực quan giúp người dùng tập trung vào thiết kế thay vì phải xử lý những chi tiết kỹ thuật phức tạp.

Bên cạnh đó, Keras có khả năng tích hợp tốt với các framework học sâu như TensorFlow, giúp tận dụng sức mạnh của nền tảng này mà vẫn giữ được sự đơn giản và dễ sử dụng. Một điểm mạnh khác của Keras là cộng đồng người dùng rộng lớn, cung cấp tài liệu phong phú, hướng dẫn chi tiết và hỗ trợ kịp thời, giúp người mới dễ dàng tiếp cận và giải quyết các vấn đề gặp phải trong quá trình phát triển mô hình.

Không chỉ phù hợp cho nghiên cứu và thử nghiệm nhanh chóng, Keras còn được ứng dụng trong các dự án thực tế và sản xuất, đáp ứng được yêu cầu về hiệu suất và khả năng triển khai. Hơn nữa, Keras hỗ trợ nhiều nền tảng, từ máy tính cá nhân đến các hệ thống điện toán đám mây, giúp tối ưu hóa việc triển khai mô hình trên nhiều thiết bị khác nhau.

2.3.2 Keras model:

Trong Keras, khái niệm "model" đề cập đến kiến trúc của một mô hình học sâu, mô tả cách các lớp (layers) được liên kết và hoạt động cùng nhau để giải quyết một nhiệm vụ

cụ thể. Keras cung cấp nhiều cách để xây dựng và tùy chỉnh mô hình, giúp người dùng linh hoạt trong việc thiết kế mạng nơ-ron.

Một trong những phương pháp phổ biến để tạo mô hình trong Keras là **Sequential Model**, trong đó các lớp được xếp chồng tuần tự, tạo thành một chuỗi tuyến tính. Chẳng hạn, bạn có thể khởi tạo một mô hình bằng cách thêm một lớp Dense, tiếp theo là một lớp Activation, và tiếp tục như vậy.

Ngoài ra, Keras còn cung cấp **Functional API**, một phương pháp mạnh mẽ hơn cho phép xây dựng các mô hình có cấu trúc phức tạp hơn, chẳng hạn như mô hình có nhiều đầu vào hoặc đầu ra, chia sẻ lớp giữa các phần của mạng, hoặc các kiến trúc phi tuần tự khác. Functional API giúp mở rộng khả năng thiết kế mô hình, đáp ứng nhu cầu của những bài toán phức tạp hơn trong học sâu.

CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM

3.1. Phân tích hệ thống

Khai báo các thư viện quan trọng như pandas, numpy, matplotlib cho việc phân tích và dự đoán giá nhà:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
```

Hình 3.1 Các thư viện import

Sau khi đã khai báo các thư viện cần thiết, ta tiến hành thực thi hai câu lệnh ‘read_csv’ và ‘info()’ trong pandas để đọc file CSV (Comma-separated values) và nắm được các thông tin cơ bản của các mẫu dữ liệu bên trong bộ dữ liệu chẳng hạn như: tên các đặc trưng, kiểu dữ liệu,...

```python
# Tải dữ liệu từ tệp Excel
data = pd.read_excel('real_estate_listings.xlsx')
```

```python
# In danh sách các cột trong tập dữ liệu để kiểm tra thông tin
print("Available columns in dataset:")
for col in data.columns:
    print(f"- {col}")
```

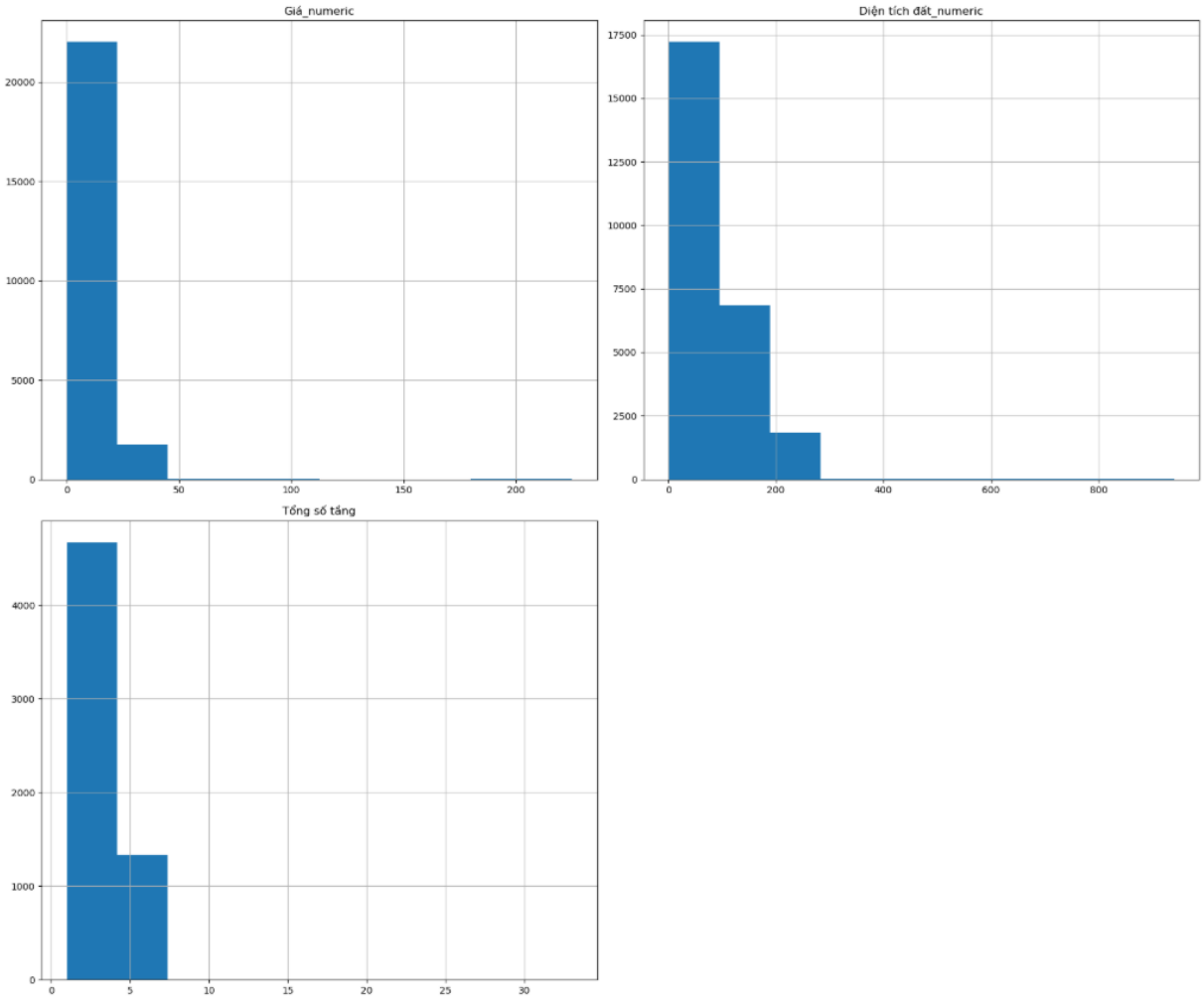
```
Available columns in dataset:
- Vị trí
- Giá
- Loại nhà
- Diện tích đất
- Số phòng ngủ
- Số toilet
- Tổng số tầng
- Hướng cửa chính
- Hướng ban công
- Giấy tờ pháp lý
```

Hình 3.2 Phân tích file dataset

Như ta thấy ở hình trên, bộ dữ liệu có tổng cộng 10 cột đặc trưng, 26048 mẫu dữ liệu và kiểu dữ liệu chính là số nguyên (int64).Trước khi đi vào phần tiền xử lý, ta cần khám phá xem các dữ liệu phân bố như thế nào trước khi đi vào phân tích thông qua câu lệnh ‘hist()’ trong matplotlib:

```
# Vẽ biểu đồ phân bố dữ liệu
plt.figure(figsize=(18,15))
data[numeric_columns].hist(figsize=(18,15))
plt.tight_layout()
plt.show()
```

Hình 3.3 Khám phá dữ liệu



Hình 3.4 Biểu đồ phân bố dữ liệu

Việc sử dụng ‘hist()’ trong matplotlib trước khi tiền xử lý dữ liệu giúp ta có cái nhìn rõ ràng về dữ liệu ban đầu, từ đó giúp định hình được quy trình tiền xử lý và phân tích dữ liệu sau này. Do thuật toán Random Forest không có quy định quá khắt khe về dữ liệu và bộ dữ liệu nhóm mình đang sử dụng không có các điểm nhiễu dữ liệu và bên cạnh đó tất cả các đặc

trưng đều cần thiết cho việc huấn luyện và đưa ra dự đoán do đó nhóm sẽ tạm bỏ qua bước tiền xử lý dữ liệu:

Sau khi quá trình xử lý dữ liệu hoàn tất, ta tiến hành sử dụng phương thức 'train_test_split' bên trong thư viện sklearn.model_selection cho việc chia bộ dữ liệu cho việc huấn luyện và kiểm thử.

```
# Chuẩn bị dữ liệu để huấn luyện mô hình
X = data[numeric_columns]  # Đầu vào
y = data['AboveMedianPrice']  # Nhãn
```

Hình 3.5 Chuẩn bị dữ liệu X, Y

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Hình 3.6 Chia dữ liệu thành tập huấn luyện và tập kiểm tra

Như trong ảnh chứa đoạn code trên, nhóm mình đã chọn các đặc trưng dạng số từ tập dữ liệu bằng cách sử dụng danh sách numeric_columns để làm đầu vào cho biến X. Cột 'AboveMedianPrice' được chọn làm nhãn đầu ra (y) để dự đoán giá trị của nó. Việc này giúp mô hình chỉ học từ các đặc trưng số có ý nghĩa mà không bao gồm các dữ liệu không phù hợp như ID hoặc các cột phân loại chưa được mã hóa. Nhóm sử dụng phương pháp train_test_split để chia dữ liệu với tỷ lệ 80% dành cho huấn luyện và 20% còn lại cho kiểm thử. Đồng thời, tham số random_state=42 giúp giữ nguyên kết quả phân chia dữ liệu trong các lần chạy tiếp theo, đảm bảo tính tái lập của mô hình.

Cuối cùng, ta sẽ khai báo mô hình Random Forest từ bên trong thư viện sklearn , đưa dữ liệu vào mô hình, tính toán độ chính xác và cuối cùng là dự đoán dựa trên bộ dữ liệu kiểm thử.

- Khai báo mô hình và tiến hành đưa dữ liệu vào mô hình:

```
from sklearn.ensemble import RandomForestClassifier

forest = RandomForestClassifier(n_estimators=20,random_state=42)
forest.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(n_estimators=20, random_state=42)
```

Hình 3.7 Huấn luyện mô hình Random Forest

- Dự đoán trên bộ dữ liệu kiểm tra:

```
# Dự đoán trên tập kiểm tra
predictions = forest.predict(X_test)
```

```
print(predictions[:20])  # In 20 kết quả đầu tiên
```

```
[0 0 0 1 0 0 1 1 0 1 0 1 1 0 0 0 1 1 0 0]
```

Hình 3.8 Dự đoán bộ dữ liệu kiểm tra

- Kiểm tra độ chính xác của mô hình:

```
# Đánh giá độ chính xác của mô hình
accuracy = forest.score(X_test, y_test)
print(f"Model accuracy: {accuracy}")
```

```
Model accuracy: 0.999616122840691
```

Hình 3.9 Kiểm tra độ chính xác của mô hình

3.2. Xây dựng giao diện sản phẩm

Quá trình khai báo thư viện và đọc dữ liệu CSV đều tương tự như phần trước nên nhóm mình sẽ không nói lại ở phần này mà chỉ tập trung vào các phần về sau:

- Chia dữ liệu thành 2 phần, phần đặc trưng X và nhãn y:

```
# Chuẩn bị dữ liệu để huấn luyện mô hình
X = data[numeric_columns]  # Đầu vào
y = data['AboveMedianPrice']  # Nhãn
```

Hình 3.10 Chia dữ liệu X, Y

- Do bộ dữ liệu có một vài đặc trưng phân bố không đồng đều từ hàng trăm đến hàng nghìn. Các yếu tố này sẽ ảnh hưởng trực tiếp đến thời gian huấn luyện mô hình nên ở đây, nhóm sẽ sử dụng phương pháp MinMaxScaler() có trong thư viện preprocessing của sklearn để biến đổi các giá trị nằm trong khoảng 0 đến 1 và từ đó tăng hiệu suất thời gian huấn luyện:

```
# Chuẩn hóa dữ liệu
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
```

Hình 3.11 Chuẩn hóa dữ liệu

- Sau khi hoàn thành việc Scaling dữ liệu, ta tiến hành chia bộ dữ liệu cho việc huấn luyện, xác thực và kiểm tra như sau:

```
# Chia tập dữ liệu thành tập huấn luyện, kiểm tra và xác thực
from sklearn.model_selection import train_test_split
X_train, X_val_test, y_train, y_val_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_val_test, y_val_test, test_size=0.5, random_state=42)
```

Hình 3.12 Chia tập dữ liệu thành tập huấn luyện, kiểm tra và xác thực

- Sau khi chia dữ liệu hoàn tất, ta thu được 70% dữ liệu cho việc huấn luyện, 15% cho xác thực và 15% còn lại cho kiểm tra. Tiếp đến, ta tiến hành khai báo mô hình Nơ-ron cần sử dụng, ở đây nhóm mình sẽ sử dụng mô hình Mạng Nơ-ron Nhân tạo

(Artificial Neural Networks) bao gồm 2 lớp ẩn sử dụng ReLu, hàm Sigmoid cho đầu ra và số lượng đầu vào sẽ là tổng số lượng đặc trưng của bộ dữ liệu:

```
# Xây dựng mô hình mạng nơ-ron bằng Keras
from keras.models import Sequential
from keras.layers import Dense

input_size = X_train.shape[1]
model = Sequential([
    Dense(32, activation='relu', input_shape=(input_size,)),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid'),
])
```

Hình 3.13 Xây dựng mô hình mạng nơ-ron bằng Keras

- Tiếp đến là quá trình biên dịch (Compile) mô hình trong Keras:

```
# Biên dịch mô hình
model.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
```

Hình 3.14 Biên dịch mô hình

Mô hình sẽ sử dụng Gradient Descent Stochastic cho việc tối ưu, Binary Cross Entropy cho hàm mất mát và độ đo lường là Accuracy.

- Sau khi đã thiết lập đầy đủ các thông số, nhóm sẽ tiến hành fit dữ liệu vào mô hình như sau:

```python
# Huấn luyện mô hình
history = model.fit(X_train, y_train, batch_size=32, epochs=100, validation_data=(X_val, y_val))
```

```
Epoch 1/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.5574 - loss: 0.6868 - val_accuracy: 0.5672 - val_loss: 0.6785
Epoch 2/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.5591 - loss: 0.6788 - val_accuracy: 0.5672 - val_loss: 0.6747
Epoch 3/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.5559 - loss: 0.6761 - val_accuracy: 0.5669 - val_loss: 0.6714
Epoch 4/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.5583 - loss: 0.6717 - val_accuracy: 0.5687 - val_loss: 0.6665
Epoch 5/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.5637 - loss: 0.6659 - val_accuracy: 0.6153 - val_loss: 0.6613
Epoch 6/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.6019 - loss: 0.6613 - val_accuracy: 0.6163 - val_loss: 0.6554
Epoch 7/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.6123 - loss: 0.6537 - val_accuracy: 0.6317 - val_loss: 0.6469
Epoch 8/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.6213 - loss: 0.6459 - val_accuracy: 0.6335 - val_loss: 0.6363
Epoch 9/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.6301 - loss: 0.6339 - val_accuracy: 0.6337 - val_loss: 0.6232
Epoch 10/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.6721 - loss: 0.6205 - val_accuracy: 0.7530 - val_loss: 0.6076
Epoch 11/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.7260 - loss: 0.6046 - val_accuracy: 0.7530 - val_loss: 0.5877
Epoch 12/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.7930 - loss: 0.5839 - val_accuracy: 0.8805 - val_loss: 0.5654
Epoch 13/100
...
Epoch 99/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.9354 - loss: 0.1478 - val_accuracy: 0.9365 - val_loss: 0.1665
Epoch 100/100
570/570 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.9368 - loss: 0.1436 - val_accuracy: 0.9363 - val_loss: 0.1408
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Hình 3.15 Huấn luyện mô hình

Nhóm sẽ lấy 32 bộ dữ liệu và fit vào mô hình ở mỗi lần lặp và sử dụng các bộ dữ liệu xác thực (Validation) đã khai báo trước đó cho việc xác thực trong giới hạn là 100 lần lặp.

- Sau khi đã huấn luyện xong, nhóm tiến hành kiểm tra độ chính xác dựa trên bộ dữ liệu kiểm tra:

```
# Đánh giá mô hình trên tập kiểm tra
test_accuracy = model.evaluate(X_test, y_test)[1]
print(f"Độ chính xác trên tập kiểm tra: {test_accuracy}")
```

```
123/123 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9319 - loss: 0.1425
Độ chính xác trên tập kiểm tra: 0.9324462413787842
```

Hình 3.16 Đánh giá mô hình trên tập kiểm tra

3.3. So sánh kết quả

3.3.1 Random Forest:

Trong quá trình nghiên cứu, chúng em đã áp dụng mô hình Random Forest vào bài toán dự đoán giá nhà và thu được một kết quả đáng chú ý. Kết quả của quá trình đánh giá mô hình trên tập dữ liệu kiểm tra được mô tả bằng điểm số.

Kết quả của mô hình Machine Learning cho việc dự đoán giá nhà đã đạt được độ chính xác đáng kinh ngạc. Với điểm số forest.score(X_test, y_test) = 0.999616122840691 trên tập dữ liệu kiểm tra. Điểm số này là một chỉ số quan trọng, cho thấy mô hình Random Forest của chúng em có khả năng dự đoán giá nhà với độ chính xác 100% gần như tuyệt đối trên tập dữ liệu kiểm tra.

Mô hình Random Forest được chọn làm một lựa chọn lý tưởng trong bài toán này vì khả năng làm việc hiệu quả với dữ liệu có nhiều biến đầu vào và giảm thiểu được nguy cơ overfitting. Điều này có thể giải thích sự hiệu quả của mô hình trong việc dự đoán giá nhà với độ chính xác cao.

3.3.2 Neural Networks:

Kết quả của quá trình đánh giá mô hình Neural Network trên tập dữ liệu kiểm tra đã cho thấy một độ chính xác đáng chú ý. Mô hình đã đạt được tỷ lệ độ chính xác (Accuracy) lên đến 93.2%, một con số ấn tượng trong việc dự đoán giá nhà.

Quá trình đánh giá đã hoàn thành với hiệu suất cao và ổn định, được thực hiện trong 7 bước một cách nhanh chóng, mỗi bước chỉ mất 0 giây để thực hiện. Điều này chỉ ra sự hiệu quả và khả năng tối ưu hóa của mô hình Neural Network.

Mặc dù độ chính xác của mô hình là rất cao, cần phải lưu ý rằng mất mát (Loss) được ước lượng là 0.1425. Điều này có thể gợi ý về sự phức tạp của mô hình hoặc sự hiện diện của overfitting, mặc dù độ chính xác vẫn được duy trì ở mức cao.

Qua quá trình áp dụng các thuật toán Machine Learning vào việc dự đoán giá nhà, chúng em đã thu được một mô hình có hiệu suất khá ấn tượng. Kết quả của quá trình đánh giá mô hình trên tập dữ liệu kiểm tra đã cho thấy tỷ lệ độ chính xác xấp xỉ 93.2%, cùng với các chỉ số đánh giá khác như độ chính xác và mất mát.

KẾT LUẬN

1. Kết quả đạt được

Trong đề tài này, nhóm đã nghiên cứu và ứng dụng hai thuật toán Random Forest và Neural Network để dự đoán giá nhà dựa trên dữ liệu bất động sản. Quá trình thực hiện bao gồm thu thập, tiền xử lý dữ liệu, xây dựng mô hình, huấn luyện và đánh giá hiệu suất của hai thuật toán.

Thuật toán Random Forest đã chứng tỏ hiệu quả cao trong việc dự đoán giá nhà với độ chính xác lên đến 99.9% trên tập kiểm tra. Phương pháp này giúp giảm thiểu overfitting và cung cấp dự đoán ổn định nhờ vào cơ chế tổng hợp từ nhiều cây quyết định.

Trong khi đó, mô hình Neural Network với kiến trúc nhiều lớp và dùng các hàm kích hoạt như ReLU và Sigmoid cũng đạt độ chính xác 93.2%. Mạng nơ-ron cho thấy khả năng học sâu và phát hiện mối quan hệ phức tạp qua các đặc trưng của dữ liệu bất động sản.

Mặc dù đạt được kết quả khả quan, đề tài vẫn còn một số hạn chế. Trước hết, mô hình Neural Network có thể gặp vấn đề overfitting khi số lượng tham số quá lớn hoặc dữ liệu chưa được tối ưu. Điều này có thể làm giảm khả năng tổng quát hóa của mô hình khi áp dụng trên dữ liệu thực tế

2. Hướng phát triển

Tối ưu hóa mô hình: Sử dụng các kỹ thuật như Regularization (L1, L2), Dropout để giảm thiểu overfitting trong mô hình Neural Network. Ngoài ra, có thể thử nghiệm các kiến trúc tiên tiến hơn như CNN (Convolutional Neural Networks) hoặc LSTM (Long Short-Term Memory) để dự đoán giá nhà theo chuỗi thời gian.

Mở rộng phạm vi nghiên cứu: Áp dụng mô hình trên các tập dữ liệu lớn hơn và đa dạng hơn, có thể mở rộng sang các thành phố khác hoặc tích hợp thêm các dữ liệu về kinh tế, xu hướng thị trường.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] M. Yazdani, "Machine Learning, Deep Learning, and Hedonic Methods for Real Estate Price Prediction," arXiv preprint arXiv:2110.07151, 2021.

[2] A. K. Das, S. Ghosh, và S. Ghosh, "Predicting House Prices with Machine Learning Algorithms: A Comparative Study," International Journal of Advanced Computer Science and Applications, vol. 11, no. 2, pp. 193-200, 2020. DOI: 10.14569/IJACSA.2020.0110225

[3] H. Samarakoon, M. Wijewickrema, và J. Wickramasinghe, "House Price Prediction Using Machine Learning Algorithms: A Case Study of Melbourne City, Australia," Asian Journal of Computer Science and Technology, vol. 8, no. 1, pp. 1-6, 2019.

[4] S. S. Devi và S. K. Singh, "House Price Prediction Using Random Forest Machine Learning Technique," International Journal of Recent Technology and Engineering, vol. 8, no. 3, pp. 227-230, 2019. DOI: 10.35940/ijrte.C4243.098319

[5] N. A. Rahadi, D. K. Wiryono, D. A. Koesrindartoto, và I. Syamwil, "Factors influencing the price of housing in Indonesia," International Journal of Housing Markets and Analysis, vol. 8, no. 2, pp. 169-188, 2015. DOI: 10.1108/IJHMA-04-2014-0011

[6] Dataset on Kaggle: https://www.kaggle.com/datasets/lammike/vietnam-housing-hcm/data