**UỶ BAN NHÂN DÂN TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**----------------------------------------**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN CUỐI KỲ**

***KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG***

***Dự đoán giá nhà ở Hoa Kỳ***

**Sinh viên thực hiện:**

Họ và tên MSSV Mã Lớp

**NGUYỄN TRỌNG TRIẾT 3120410554 DCT1203**

GIẢNG VIÊN HƯỚNG DẪN : **ThS.NGUYỄN THANH PHƯỚC**

**TP. Hồ Chí Minh, tháng 5 năm 2024**

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc165823142)

[**CHƯƠNG 1 : TỔNG QUAN VỀ ĐỀ TÀI** 2](#_Toc165823143)

[1.1 Đặt vấn đề 2](#_Toc165823144)

[1.2 Mục tiêu của đề tài 2](#_Toc165823145)

[1.3 Đối tượng nghiên cứu 2](#_Toc165823146)

[1.4 Bố cục đề tài 2](#_Toc165823147)

[**CHƯƠNG 2: GIỚI THIỆU VỀ HỌC MÁY VÀ KHAI THÁC DỮ LIỆU** 4](#_Toc165823148)

[2.1 Giới thiệu về học máy và khai thác dữ liệu 4](#_Toc165823149)

[Khái niệm về học máy và khai thác dữ liệu 4](#_Toc165823150)

[Ứng dụng của học máy 5](#_Toc165823151)

[Ứng dụng của khai thác dữ liệu 5](#_Toc165823152)

[2.2 Thuật toán hồi quy tuyến tính trong học máy 6](#_Toc165823153)

[Khái niệm 6](#_Toc165823154)

[Dạng của Linear Regression 6](#_Toc165823155)

[Hàm mất mát 7](#_Toc165823156)

[2.3 Mô hình Decision Tree 8](#_Toc165823157)

[Khái niệm 8](#_Toc165823158)

[Cấu trúc 8](#_Toc165823159)

[Quy trình xây dựng 9](#_Toc165823160)

[Ứng dụng 10](#_Toc165823161)

[2.4 Mô hình Random Forest 10](#_Toc165823162)

[Khái niệm 10](#_Toc165823163)

[Cấu trúc 10](#_Toc165823164)

[Quy trình xây dựng 11](#_Toc165823165)

[Ưu Điểm 11](#_Toc165823166)

[Ứng Dụng 11](#_Toc165823167)

[**CHƯƠNG 3 : CÀI ĐẶT CHƯƠNG TRÌNH THỬ NGHIỆM** 13](#_Toc165823168)

[3.1 Giới thiệu bài toán 13](#_Toc165823169)

[3.2 Giải pháp hiện tại của bài toán 13](#_Toc165823170)

[3.3 Giải pháp CNTT cho bài toán 13](#_Toc165823171)

[3.4 Mô tả dữ liệu 14](#_Toc165823172)

[3.5 Trong hoàn cảnh nào có thể thu thập dữ liệu 14](#_Toc165823173)

[3.6 Trong hoàn cảnh nào không thể thu thập được dữ liệu 14](#_Toc165823174)

[3.7 Xử lý dữ liệu 15](#_Toc165823175)

[Import thư viện cần thiết cho đồ án 15](#_Toc165823176)

[Xác định và xử lý dữ liệu bị bác bỏ 16](#_Toc165823177)

[Gom nhóm các đặc trưng (theo kiểu dữ liệu) 16](#_Toc165823178)

[Thống kê dữ liệu 16](#_Toc165823179)

[Phân tích đa biến 17](#_Toc165823180)

[Phân tích đơn biến 18](#_Toc165823181)

[Phân tích tương quan 20](#_Toc165823182)

[3.8 Đặc trưng 21](#_Toc165823183)

[Trích xuất đặc trưng 21](#_Toc165823184)

[Chọn lọc đặc trưng 21](#_Toc165823185)

[3.9 Xây dựng mô hình dự đoán bằng thư viện Scikit – Learn 22](#_Toc165823186)

[Phân tách dữ liệu thành train và test 22](#_Toc165823187)

[Tạo model và training 23](#_Toc165823188)

[Tinh chỉnh mô hình và đánh giá mô hình tốt nhất 25](#_Toc165823189)

[Features Importances 26](#_Toc165823190)

[Dự đoán và đánh giá mô hình 27](#_Toc165823191)

[**CHƯƠNG 4 : KẾT LUẬN** 32](#_Toc165823192)

[4.1. Tổng quan dữ liệu 32](#_Toc165823193)

[4.2. Mô Hình Linenear Regression , Decision Tree , Random Forest 32](#_Toc165823194)

[4.3. Đánh giá và kết luận 32](#_Toc165823195)

[**TÀI LIỆU THAM KHẢO** 33](#_Toc165823196)

# LỜI MỞ ĐẦU

**1. Giới thiệu**

Trong những năm gần đây, sự bùng nổ của dữ liệu đã mở ra không gian lớn cho nhiều lĩnh vực nghiên cứu và ứng dụng máy học. Trong bối cảnh này, việc dự đoán giá nhà không chỉ là một thách thức quan trọng mà còn là một lĩnh vực nghiên cứu đáng chú ý, đặc biệt trong ngữ cảnh thị trường bất động sản ngày càng biến động.

Bài báo cáo này tập trung vào việc áp dụng mô hình hồi quy tuyến tính (Linenear Regression) cùng với mô hình Random Forest và Decision Tree để dự đoán giá nhà tại Hoa Kỳ, sử dụng một bộ dữ liệu đa dạng với nhiều đặc trưng kinh tế - xã hội quan trọng. Chúng tôi sẽ đi sâu vào quá trình phân tích dữ liệu, xử lý và tiền xử lý dữ liệu, cũng như xây dựng và đánh giá các mô hình.

**2. Ngữ cảnh nghiên cứu**

Bất động sản không chỉ là nơi ở, mà còn là một phần quan trọng của cơ sở hạ tầng kinh tế và xã hội. Dự đoán giá nhà không chỉ quan trọng với mỗi cá nhân mà còn ảnh hưởng đến quyết định đầu tư, phát triển đô thị, và quản lý tài nguyên đất đai.

Mô hình hồi quy tuyến tính, mặc dù đơn giản, nhưng lại mạnh mẽ và linh hoạt, là công cụ lý tưởng để xác định mối quan hệ giữa các yếu tố kinh tế và giá nhà. Bên cạnh đó, sự bổ sung của mô hình Random Forest và Decision Tree cũng mang lại sự đa dạng và hiệu quả trong dự đoán.

Chúng ta sẽ kiểm tra mối liên kết này và đưa ra những nhận định sâu sắc về sự biến động của giá nhà.

**3. Cấu trúc bài báo cáo**

Bài báo cáo được tổ chức một cách có tổ chức, bắt đầu từ xử lý dữ liệu, tiến qua mô hình hóa, và kết thúc bằng đánh giá mô hình. Mỗi phần đều được thiết kế để cung cấp cái nhìn chi tiết và toàn diện nhất về quá trình nghiên cứu.

Chúng ta hy vọng rằng bài báo cáo này không chỉ là một tài liệu nghiên cứu mà còn là một nguồn cảm hứng và thảo luận xây dựng cầu nối giữa lý thuyết và thực tế trong lĩnh vực dự đoán giá nhà.

# CHƯƠNG 1 : TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong bối cảnh thị trường bất động sản ngày càng phát triển và biến động, việc dự đoán giá nhà trở nên cực kỳ quan trọng. Người mua, người bán, và những người làm chuyên gia bất động sản đều đặt ra câu hỏi về giá trị tài sản dựa trên nhiều yếu tố. Điều này tạo ra nhu cầu xây dựng mô hình dự đoán giá nhà, và phương pháp Linear Regression cùng với mô hình Random Forest và Decision Tree là những công cụ mạnh mẽ để giải quyết vấn đề này.

## 1.2 Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một mô hình dự đoán giá nhà sử dụng phương pháp Linear Regression kết hợp với mô hình Random Forest và Decision Tree. Mô hình sẽ dựa trên các yếu tố như thu nhập trung bình khu vực, tuổi nhà, số lượng phòng, số lượng phòng ngủ và dân số khu vực. Mô hình này không chỉ cung cấp cái nhìn chính xác về giá nhà mà còn hỗ trợ quyết định đầu tư và giao dịch bất động sản.

## 1.3 Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm cả cộng đồng liên quan đến thị trường bất động sản, đặc biệt là người mua và người bán nhà, môi giới bất động sản, và những chuyên gia quan tâm đến dự đoán giá nhà. Mô hình dự đoán có thể cung cấp thông tin hữu ích và đáng tin cậy giúp họ đưa ra quyết định thông minh..

## 1.4 Bố cục đề tài

**Phần 1: Giới thiệu tổng quan về đề tài**

Mô tả về tầm quan trọng của việc dự đoán giá nhà và ứng dụng của nó trong lĩnh vực bất động sản. Cung cấp lý do chọn phương pháp hồi quy tuyến tính(Linenear Regression) cùng với mô hình Random Forest và Decision Tree và giới thiệu các biến đầu vào.

**Phần 2: Giới thiệu về học máy & khai thác dữ liệu và mô hình Linear Regression,Random Forest,Decision Tree**

Mô tả học máy và các lý thuyết, công thức toán liên quan đến mô hình Linear Regression cũng như mô hình Random Forest và Decision Tree.

**Phần 3: Xử lý dữ liệu & đặc trưng & xây dựng mô hình Linear Regression,Random Forest,Decision Tree & Đánh giá mô hình**

Thể hiện rõ các bước xử lý dữ liệu như : import thư viện , xác định và xử lý dữ liệu bị bác bỏ , gom nhóm đặc trưng , thống kê dữ liệu , phân phối các đặc trưng số học,mối tương quan của các đặc trưng .

Thể hiện rõ các bước xử lý đặc trưng như ; Trích xuất đặc trưng và chọn lọc đặc trưng .Trình bày các bảng biểu và biểu đồ về đặc trưng và trích xuất các đặc trưng của tập dữ liệu.

Thể hiện rõ các bước để xây dựng mô hình : Phân tách dữ liệu , Tạo model và training , hệ số coeff , dự đoán và đánh giá mô hình

**Phần 4: Kết luận và hướng phát triển & tài liệu tham khảo**

Tóm tắt tổng quan dữ liệu và mô hình hồi quy. Đề xuất hướng phát triển tiếp theo và cải thiện mô hình. Liệt kê các tài liệu tham khảo và nguồn dữ liệu quan trọng được sử dụng trong quá trình nghiên cứu.

# CHƯƠNG 2: GIỚI THIỆU VỀ HỌC MÁY VÀ KHAI THÁC DỮ LIỆU

## 2.1 Giới thiệu về học máy và khai thác dữ liệu

### Khái niệm về học máy và khai thác dữ liệu

**Học máy (Machine Learning)**

Học máy là một lĩnh vực con của Trí Tuệ Nhân Tạo, nơi máy tính được lập trình để tự động học từ dữ liệu và cải thiện hiệu suất mà không cần phải được lập trình một cách tường minh. Mục tiêu của Học m áy là phát triển các mô hình và thuật toán cho phép máy tính "học" từ dữ liệu, nhận biết mẫu và đưa ra dự đoán hoặc quyết định dựa trên những thông tin đã học được. Học Máy tập trung vào việc phát triển phần mềm và chương trình máy tính có khả năng tự học từ dữ liệu.

Tuy nhiên, Học máy vẫn cần sự can thiệp của con người để đánh giá dữ liệu và lựa chọn các kỹ thuật phù hợp để phân tích. Đồng thời, trước khi sử dụng, dữ liệu cần được làm sạch và không có sai lệch.

**Khai thác dữ liệu (Data Mining)**

Khai thác dữ liệu là quá trình khám phá thông tin có ý nghĩa từ dữ liệu lớn bằng cách sử dụng các kỹ thuật phân tích và thống kê. Mục tiêu của Khai Thác Dữ Liệu là tìm ra các mẫu, quy luật và sự kết hợp trong dữ liệu để phát hiện ra những thông tin ẩn sau các tập dữ liệu phức tạp.

Các kỹ thuật phổ biến trong khai thác dữ liệu bao gồm: phân tích cụm (cluster analysis), phân tích ghi chú (association analysis) và phân tích chuỗi thời gian (time series analysis).

Mặc dù có sự chồng chéo giữa Học máy và Khai thác dữ liệu trong một số kỹ thuật và phương pháp, nhưng mỗi lĩnh vực vẫn tập trung vào các mục tiêu và ứng dụng khác nhau trong việc phân tích và sử dụng dữ liệu.

### Ứng dụng của học máy

Nhiều hoạt động hàng ngày của chúng ta được trợ giúp bởi các thuật toán machine learning, bao gồm:

* Trong y tế: xác định bệnh lý của người bệnh mới dựa trên dữ liệu lịch sử của các bệnh nhân có cùng bệnh lý có cùng các đặc điểm đã được chữa khỏi trước đây, hay xác định loại thuốc phù hợp [1]
* Trong lĩnh vực ngân hàng: xác định khả năng khách hàng chậm trả các khoản vay hoặc rủi ro tín dụng do nợ xấu dựa trên phân tích Credit score; xác định xem liệu các giao dịch có hành vi phạm tội, lừa đảo hay không. [2]
* Trong giáo dục: phân loại các học sinh theo hoàn cảnh, học lực để xem xem cần hỗ trợ gì cho những học sinh ví dụ như hoàn cảnh sống khó khăn nhưng học lực lại tốt. [3]
* Trong thương mại điện tử: phân loại khách hàng theo sở thích cụ thể để hỗ trợ personalized marketing hay xây dựng hệ thống khuyến nghị, dựa trên dữ liệu từ website, social media. [4]

### Ứng dụng của khai thác dữ liệu

Ứng dụng của Khai Thác Dữ Liệu cũng rất đa dạng và quan trọng trong nhiều lĩnh vực. Dưới đây là một số ví dụ về cách mà khai thác dữ liệu được sử dụng trong cuộc sống hàng ngày và các lĩnh vực khác nhau .

**Y tế**

Xác định bệnh lý: Khai thác dữ liệu có thể được sử dụng để phân tích lịch sử bệnh lý của các bệnh nhân và xác định các biểu hiện chung của một bệnh để giúp trong việc chẩn đoán bệnh mới cho các bệnh nhân. [5]

Dự đoán phản ứng với liệu pháp: Dữ liệu từ các trường hợp điều trị trước đây có thể được sử dụng để dự đoán phản ứng với liệu pháp của các bệnh nhân mới.[5]

**Ngân hàng**

Đánh giá rủi ro tín dụng: Khai thác dữ liệu được sử dụng để phân tích hành vi tài chính của khách hàng và dự đoán khả năng trả nợ, đồng thời xác định nguy cơ nợ xấu và gian lận tài chính. [6]

Phát hiện giao dịch gian lận: Dữ liệu từ các giao dịch trước đó được sử dụng để xác định các mẫu và biểu hiện của giao dịch gian lận trong tương lai. [6]

**Giáo dục**

Phân loại học sinh: Khai thác dữ liệu giúp phân loại học sinh theo nhiều tiêu chí như hoàn cảnh, học lực để có thể cung cấp hỗ trợ phù hợp và cá nhân hóa chương trình giáo dục. [7]

Dự đoán hiệu suất học tập: Dữ liệu về hành vi học tập và kết quả của học sinh có thể được sử dụng để dự đoán hiệu suất học tập của họ và xác định các yếu tố ảnh hưởng. [7]

**Thương mại điện tử**

Personalized marketing: Dữ liệu từ hành vi mua hàng của khách hàng có thể được sử dụng để tạo ra các chiến lược quảng cáo cá nhân hóa và tùy chỉnh sản phẩm dựa trên sở thích cụ thể của từng khách hàng. [8]

Hệ thống khuyến nghị: Khai thác dữ liệu giúp xây dựng các hệ thống khuyến nghị sản phẩm dựa trên lịch sử mua hàng và các thông tin phản hồi từ khách hàng trên các nền tảng thương mại điện tử và mạng xã hội. [8]

## 2.2 Thuật toán hồi quy tuyến tính trong học máy

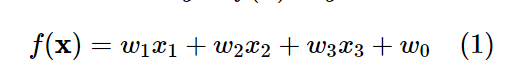
### Khái niệm

Hồi quy tuyến tính là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Hồi quy tuyến tính là một trong hai dạng lớn của học có giám sát (supervised learning) dựa trên tập dữ liệu mẫu.

Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó .

### Dạng của Linear Regression

Hồi quy tuyến tính có phương trình đạng :

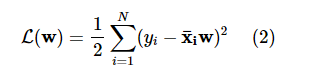


Trong đó, w1, w2, wn, w0 là các hằng số, w0 còn được gọi là bias hay sai số. Mối quan hệ giữa y f(x) bên trên là một mối quan hệ tuyến tính (linear). Bài toán chúng ta đang làm là một bài toán thuộc loại regression. Bài toán đi tìm các hệ số tối ưu { w1, w2, wn, w0 } chính vì vậy được gọi là bài toán Linear Regression (Hồi quy tuyến tính).Trong phương trình (1) nếu chúng ta đặt **w = [**w0, w1, w2, wn**]**T là một vecter (cột) hệ số cần phải tối ưu và = [1, x1, x2, xn] (đọc là x bar trong tiếng Anh) là vector (hàng) dữ liệu đầu vào mở rộng. Số 1 ở đầu được thêm vào để phép tính đơn giản hơn và thuận tiện cho việc tính toán. Khi đó, phương trình (1) có thể được viết lại dưới dạng:

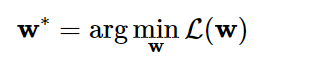


### Hàm mất mát

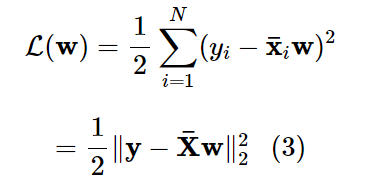
Máy học từ giá trị trung bình của một hàm mất mát. Đây là một phương pháp đánh giá độ hiệu quả của một thuật toán nào đó trên bộ dữ liệu cho trước. Nếu kết quả dự đoán chênh lệch quá nhiều so với kết quả thực tế, hàm mất mát sẽ là một số rất lớn. Điều tương tự xảy ra với tất cả các cặp (xi, yi), i = 1, 2, 3,…, N với N là số lượng dữ liệu quan sát được. Để hàm mất mát nhỏ nhất khi đó tổng sai số là nhỏ nhất tương đương với việc tìm **w** để hàm số sau đạt giá trị nhỏ nhất:



Hàm số L(w)được gọi là hàm mất mát (loss function) của bài toán Linear Regression. Chúng ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số **w** sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của **w** làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là *điểm tối ưu* (optimal point), ký hiệu:



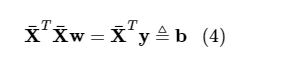
Trước khi đi tìm lời giải, chúng ta đơn giản hóa phép toán trong phương trình hàm mất mát (2). Đặt y = [y1;y2;…;yn] là một vector cột chứa tất cả các output của training data; X=[¯x1;¯x2;…;¯xN] là ma trận dữ liệu đầu vào (mở rộng) mà mỗi hàng của nó là một điểm dữ liệu. Khi đó hàm số mất mát **L(w)** được viết dưới dạng ma trận đơn giản hơn:



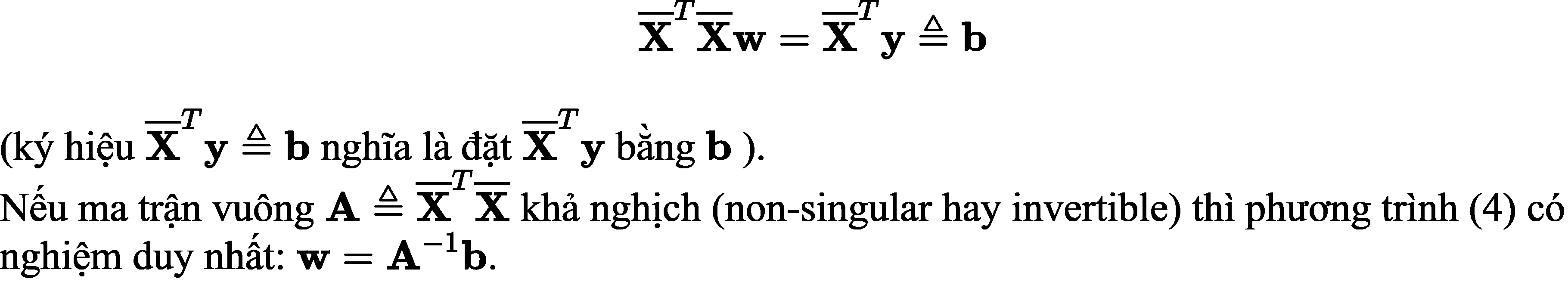
**Tìm nghiệm của mô hình hồi quy tuyến tính**

Để tìm nghiệm cho một bài toán tối ưu chúng ta thường giải phương trình đạo hàm J(w)=0.

Đạo hàm theo **w** của hàm mất mát là:



Phương trình đạo hàm bằng 0 tương đương với:



## 2.3 Mô hình Decision Tree

### Khái niệm

Mô hình Decision Tree là một công cụ học máy phổ biến được sử dụng cho các tác vụ phân loại và dự đoán. Nó hoạt động bằng cách phân tách dữ liệu thành các nhóm con dựa trên các quy tắc quyết định đơn giản. Mỗi nút trong cây đại diện cho một biến đầu vào và mỗi nhánh đại diện cho một lựa chọn hoặc quyết định. Quá trình phân tách tiếp tục cho đến khi đạt được một điều kiện dừng được xác định trước hoặc không còn thêm thông tin chia tách được.

### Cấu trúc

Cây quyết định bao gồm các thành phần sau:

* Nút gốc (Root Node): Là nút đầu tiên của cây, chứa thuộc tính được sử dụng để phân loại hoặc dự đoán.
* Nút Nội Bộ (Internal Nodes): Là các nút không phải là lá, chứa các quy tắc quyết định.
* Nhánh (Branches): Biểu diễn quyết định hoặc kết quả của mỗi quy tắc.
* Nút Lá (Leaf Nodes): Là nút cuối cùng của cây, chứa kết quả dự đoán hoặc lớp của dữ liệu đầu vào.



### Quy trình xây dựng

Quy trình xây dựng cây quyết định bao gồm các bước sau:

* Chọn thuộc tính: Chọn thuộc tính tốt nhất để phân chia tập dữ liệu. Các thuộc tính được chọn dựa trên các tiêu chí như độ thuần khiết (purity) hoặc giảm độ không chắc chắn (uncertainty).
* Phân chia dữ liệu: Phân chia tập dữ liệu thành các tập con dựa trên giá trị của thuộc tính được chọn.
* Lặp lại hoặc dừng: Nếu một điều kiện dừng được đạt được (ví dụ: tất cả các mẫu trong một nhóm thuộc cùng một lớp), dừng quá trình. Nếu không, lặp lại quá trình trên mỗi tập con.
* Tạo các nhánh: Tạo các nút nội bộ và nhánh dựa trên quyết định của thuộc tính được chọn.
* Lá (Leaf): Gán nhãn cho các lá bằng cách đưa ra dự đoán hoặc lớp thường xuất hiện nhất trong tập dữ liệu.

### Ứng dụng

Mô hình Decision Tree được sử dụng rộng rãi trong các lĩnh vực như:

* Dự đoán: Dự đoán giá cổ phiếu, dự đoán doanh số bán hàng, v.v.
* Phân loại: Phân loại email spam, phát hiện gian lận tín dụng, v.v.
* Hỗ trợ quyết định: Hỗ trợ quyết định y tế, hỗ trợ quyết định tài chính, v.v.

Mô hình Decision Tree là một công cụ mạnh mẽ và dễ hiểu, giúp giải quyết nhiều vấn đề trong phân tích dữ liệu và dự đoán. Tuy nhiên, nó cũng có thể dễ dàng bị quá mức và không ổn định nếu không được điều chỉnh đúng cách.

## 2.4 Mô hình Random Forest

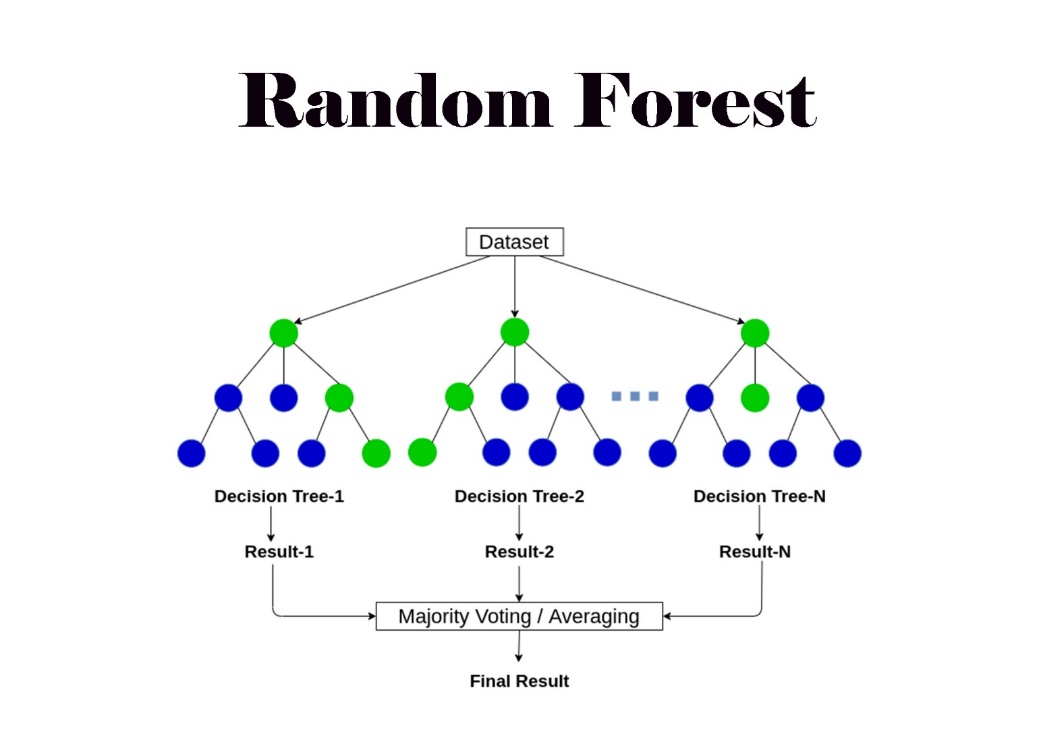
### Khái niệm

Mô hình Random Forest là một phương pháp kết hợp nhiều cây Decision Tree để tạo ra một mô hình dự đoán. Quy trình này thực hiện bằng cách tạo ra nhiều cây quyết định độc lập nhau từ các tập dữ liệu con được lấy mẫu ngẫu nhiên từ tập dữ liệu huấn luyện. Sau đó, mỗi cây sẽ đưa ra dự đoán riêng và kết quả cuối cùng sẽ là kết quả của việc bầu chọn hoặc trung bình các dự đoán từ các cây con. Mô hình Random Forest thường có hiệu suất tốt hơn so với một cây quyết định đơn lẻ vì nó giảm thiểu được hiện tượng overfitting và có khả năng xử lý dữ liệu nhiễu tốt hơn.

### Cấu trúc

Mô hình Random Forest bao gồm các thành phần sau:

* Các Cây Quyết Định (Decision Trees): Là các cây quyết định độc lập với nhau, mỗi cây được xây dựng trên một tập dữ liệu con khác nhau được lấy mẫu từ tập dữ liệu huấn luyện gốc.
* Phương Pháp Bầu Chọn (Voting Method): Kết quả cuối cùng của mô hình được tính dựa trên bầu chọn hoặc trung bình các dự đoán của từng cây quyết định.



### Quy trình xây dựng

Quy trình xây dựng một mô hình Random Forest bao gồm các bước sau:

* Lấy Mẫu Ngẫu Nhiên: Lấy mẫu ngẫu nhiên từ tập dữ liệu huấn luyện để tạo ra các tập dữ liệu con cho mỗi cây quyết định.
* Xây Dựng Cây Quyết Định: Xây dựng một cây quyết định trên mỗi tập dữ liệu con bằng cách sử dụng các thuật toán xây dựng cây quyết định như CART (Classification and Regression Trees).
* Bầu Chọn hoặc Trung Bình: Kết quả dự đoán cuối cùng được tính dựa trên bầu chọn hoặc trung bình các dự đoán của từng cây quyết định.

### Ưu Điểm

Mô hình Random Forest có các ưu điểm sau:

* Ưu Việt Trong Dự Đoán: Có khả năng dự đoán tốt và ổn định hơn so với các mô hình học máy khác.
* Giảm Overfitting: Bằng cách sử dụng nhiều cây quyết định, mô hình giảm thiểu được hiện tượng overfitting.
* Xử Lý Dữ Liệu Nhiễu: Mô hình Random Forest có khả năng xử lý dữ liệu nhiễu tốt.

### Ứng Dụng

Mô hình Random Forest được sử dụng rộng rãi trong nhiều lĩnh vực như:

* Dự đoán: Dự đoán giá cổ phiếu, dự đoán doanh số bán hàng, v.v.
* Phân loại: Phân loại email spam, phát hiện gian lận tín dụng, v.v.
* Hỗ trợ quyết định: Hỗ trợ quyết định y tế, hỗ trợ quyết định tài chính, v.v.

Mô hình Random Forest là một công cụ mạnh mẽ và linh hoạt, thường được ưa chuộng trong các ứng dụng thực tế nhờ vào khả năng dự đoán chính xác và khả năng xử lý dữ liệu nhiễu tốt.

# CHƯƠNG 3 : CÀI ĐẶT CHƯƠNG TRÌNH THỬ NGHIỆM

## 3.1 Giới thiệu bài toán

Lấy bối cảnh ở nước Mỹ, chúng ta sẽ đóng vai trò như một đại lý nhà nước để dự đoán giá nhà cho các khu vực. Với tập dữ liệu đã được chuẩn bị sẵn, nhiệm vụ bây giờ đó là sử dụng mô hình hồi quy tuyến tính để có thể ước tính ngôi nhà sẽ được bán với giá bao nhiêu.

## 3.2 Giải pháp hiện tại của bài toán

Hiện nay, Linear Regression cùng với Decision Tree và Random Forest được sử dụng rộng rãi trong dự đoán giá nhà.

Linear Regression giả định mối quan hệ tuyến tính giữa các biến độc lập và biến phụ thuộc, và dựa trên việc tìm ra đường thẳng tốt nhất phù hợp với dữ liệu để dự đoán giá nhà. Tuy nhiên, mô hình Linear Regression có thể đối mặt với các hạn chế khi xử lý các biến phức tạp và tương tác không tuyến tính giữa các yếu tố.

Decision Tree và Random Forest cung cấp các phương pháp linh hoạt hơn để xử lý các tương tác phức tạp và phi tuyến tính. Decision Tree phân tách dữ liệu thành các nhóm con dựa trên quy tắc quyết định đơn giản, trong khi Random Forest kết hợp nhiều cây quyết định để tạo ra một mô hình dự đoán mạnh mẽ và ổn định hơn.

## 3.3 Giải pháp CNTT cho bài toán

Sử dụng Machine Learning và Công nghệ Đám mây

* Để cải thiện hiệu suất dự đoán, có thể tích hợp các giải pháp CNTT như Machine Learning và công nghệ đám mây. Sử dụng các mô hình phức tạp hơn như Gradient Boosting có thể xử lý tốt hơn sự phức tạp của dữ liệu. Đồng thời, việc triển khai mô hình trên nền tảng đám mây giúp tối ưu hóa quá trình tính toán và quản lý tài nguyên.

Big Data và Analytics

* Việc sử dụng Big Data cho phép xử lý lượng lớn dữ liệu từ nhiều nguồn khác nhau, giúp tăng cường độ chính xác của mô hình. Các phương pháp Analytics cũng có thể được tích hợp để hiểu rõ hơn về xu hướng thị trường và yếu tố ảnh hưởng đến giá nhà.

## 3.4 Mô tả dữ liệu

Tệp dữ liệu là file mở rộng CSV. Trong tệp dữ liệu này có 7 cột và 5000 hàng

|  |  |
| --- | --- |
| Avg. Area Income | Thu nhập trung bình tại khu vực ngôi nhà đã bán |
| Avg. Area House Age | Trung bình tuổi của một ngôi nhà đã bán |
| Avg. Area Number of Rooms | Trung bình diện tích các phòng |
| Avg. Area Number of Bedrooms | Trung bình diện tích phòng ngủ |
| Area Population | Dân số tại khu vực bán nhà |
| Price | Giá ngôi nhà đã bán |
| Address | Địa chỉ ngôi nhà bán |

## 3.5 Trong hoàn cảnh nào có thể thu thập dữ liệu

Dữ liệu có thể được thu thập từ các nguồn như trang web bất động sản, cơ sở dữ liệu chính phủ, hay thông qua các cuộc khảo sát trực tuyến. Công nghệ tự động hóa có thể được áp dụng để liên tục cập nhật dữ liệu từ các nguồn này.

## 3.6 Trong hoàn cảnh nào không thể thu thập được dữ liệu

Trong trường hợp thông tin nhà đất được giữ kín bởi các tổ chức hoặc không có sẵn công cộng, việc thu thập dữ liệu có thể trở nên khó khăn. Các biện pháp pháp lý và đạo đức cần được xem xét khi thu thập thông tin nhạy cảm.

## 3.7 Xử lý dữ liệu

### Import thư viện cần thiết cho đồ án

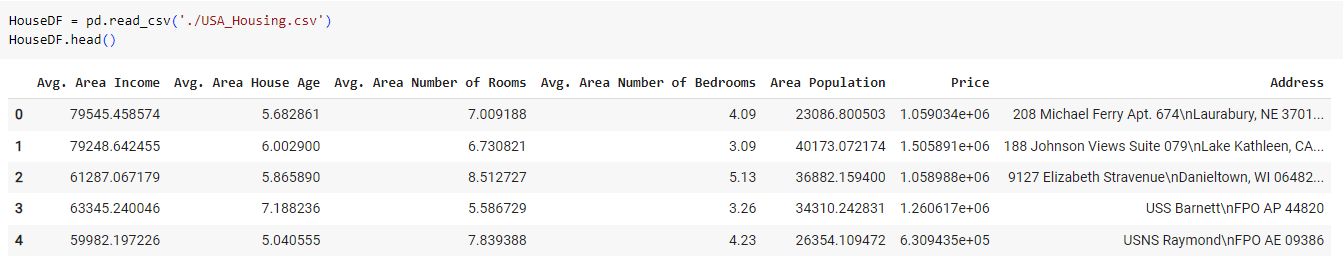
Chúng ta sẽ sử dụng 1 số thư viện cần thiết để xử lí và phân tích trực quan dữ liệu:

import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.ensemble import RandomForestRegressor  
from sklearn.tree import DecisionTreeRegressor  
from sklearn.preprocessing import StandardScaler  
from sklearn import metrics  
%matplotlib inline

Tiếp theo chúng ta sẽ load tệp dữ liệu có đuôi .csv:

HouseDF = pd.read\_csv('./USA\_Housing.csv')

HouseDF.head()



Như đã nói ở trên dataset này gồm có các thông tin cơ bản đó là :

* Avg. Area Income: Thu nhập trung bình tại khu vực ngôi nhà đã bán
* Avg. Area House Age: Trung bình tuổi của một ngôi nhà đã bán
* Avg. Area Number of Rooms: Trung bình diện tích các phòng
* Avg. Area Number of Bedrooms: Trung bình diện tích phòng ngủ
* Area Population: Dân số tại khu vực bán nhà
* Price: Giá ngôi nhà đã bán
* Address: Địa chỉ ngôi nhà bán

### Xác định và xử lý dữ liệu bị bác bỏ

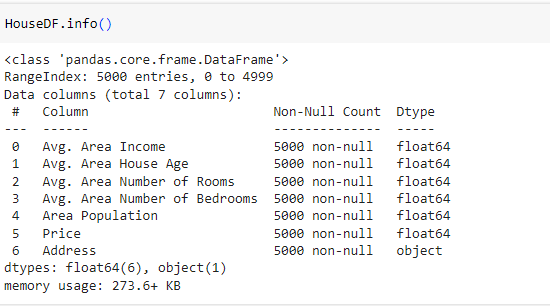
Chúng ta sẽ kiểm tra xem dữ liệu có hợp lệ không nếu không hợp lệ thì loại bỏ nó .

HouseDF.dropna(inplace=True)  
x = HouseDF[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population']]  
y = HouseDF['Price'] *# Extracting target variable directly as a Series*print("Số lượng dữ liệu sau khi xử lý bác bỏ:", len(HouseDF))



Để có thể xem tổng quát về dataset này ta có thể sử dụng và ta có thể thấy không có dữ liệu nào bị bác bỏ

HouseDF.info()



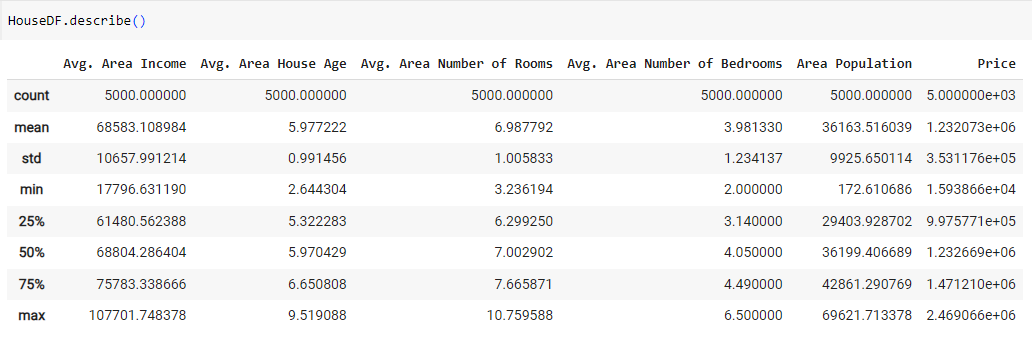
### Gom nhóm các đặc trưng (theo kiểu dữ liệu)

Như chúng ta có thể thấy chỉ có 2 kiểu dữ liệu chính là float và object

### Thống kê dữ liệu

Dựa trên dữ liệu này ta có 5000 dòng tương ứng với 5000 ngôi nhà đã được bán. Lượng dữ liệu này đủ để có thể xây dựng một mô hình học máy. Ngoài ra ta còn có thể thống kê dữ liệu này bằng describe():

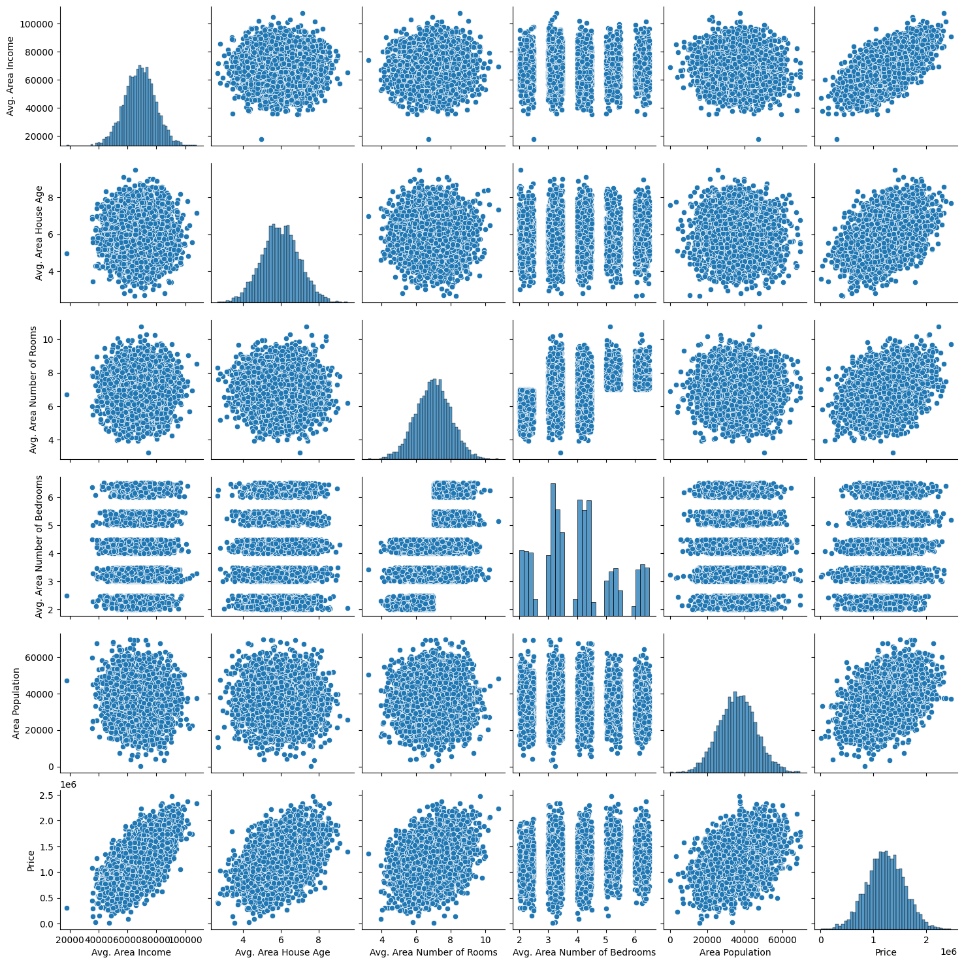
HouseDF.describe()



Nhờ hàm describe() giúp trả về dataframe mới với số hàng được hiển thị ra các thông số như số hàng, giá trị trung bình, độ lệch chuẩn, min, max, tỉ lệ phần trăm của các cột.

### Phân tích đa biến

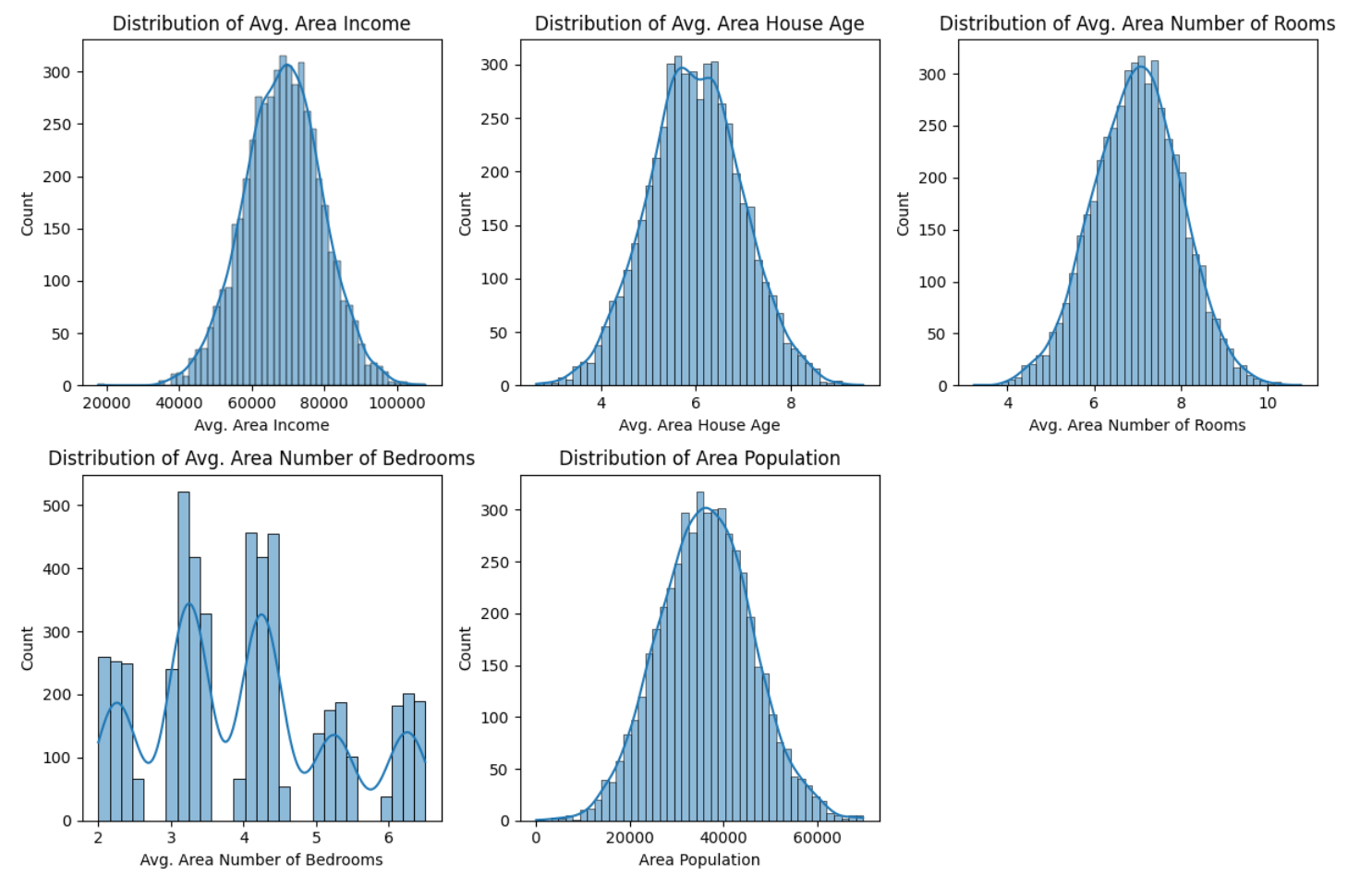
sns.pairplot(HouseDF)  
plt.show()



### Phân tích đơn biến

Chúng ta sẽ vẽ đồ thị để thấy được phân phối của đặc trưng số học

plt.figure(figsize=(12, 8))  
for i, column in enumerate(x.columns, 1):  
 plt.subplot(2, 3, i)  
 sns.histplot(x[column], kde=True)  
 plt.title(f'Distribution of {column}')  
  
plt.tight\_layout()  
plt.show()



Dữ liệu từ hình ảnh bao gồm năm biểu đồ:

**Thu nhập bình quân :** Biểu đồ này mô tả phân bố thu nhập trung bình, cho thấy thu nhập chủ yếu nằm ở mức 55k – 80k.

**Độ tuổi trung bình của nhà :** Biểu đồ này mô tả phân bố tuổi trung bình tuổi của nhà, cho thấy trung bình tuổi của nhà nằm từ 5 – 7 là nhiều nhất .

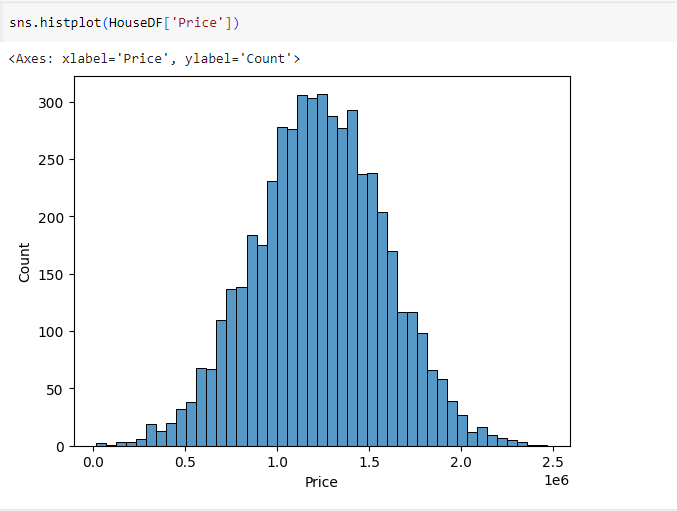
**Số phòng trung bình :** Biểu đồ này mô tả phân bố số phòng ngủ trung bình, đa số các căn nhà có từ 6 – 8 phòng.

**Số phòng ngủ trung bình :** Biểu đồ này mô tả số phòng ngủ trung bình, với các ngôi nhà có 3 -4 phòng ngủ chiếm số lượng đa số.

Về tương quan giữa các cột, ta thấy Cột Price có kiểu phân tán theo mô hình tuyến tính, dựa trên thông tin này, ta xây dựng mô hình máy học hồi quy tuyến tính để dự đoán nó dựa trên giá trị các cột khác, trừ cột địa chỉ (Address) ngôi nhà.

Bây giờ chúng ta sẽ sử dụng histplot() để vẽ biểu đồ giá nhà :

sns.histplot(HouseDF['Price'])

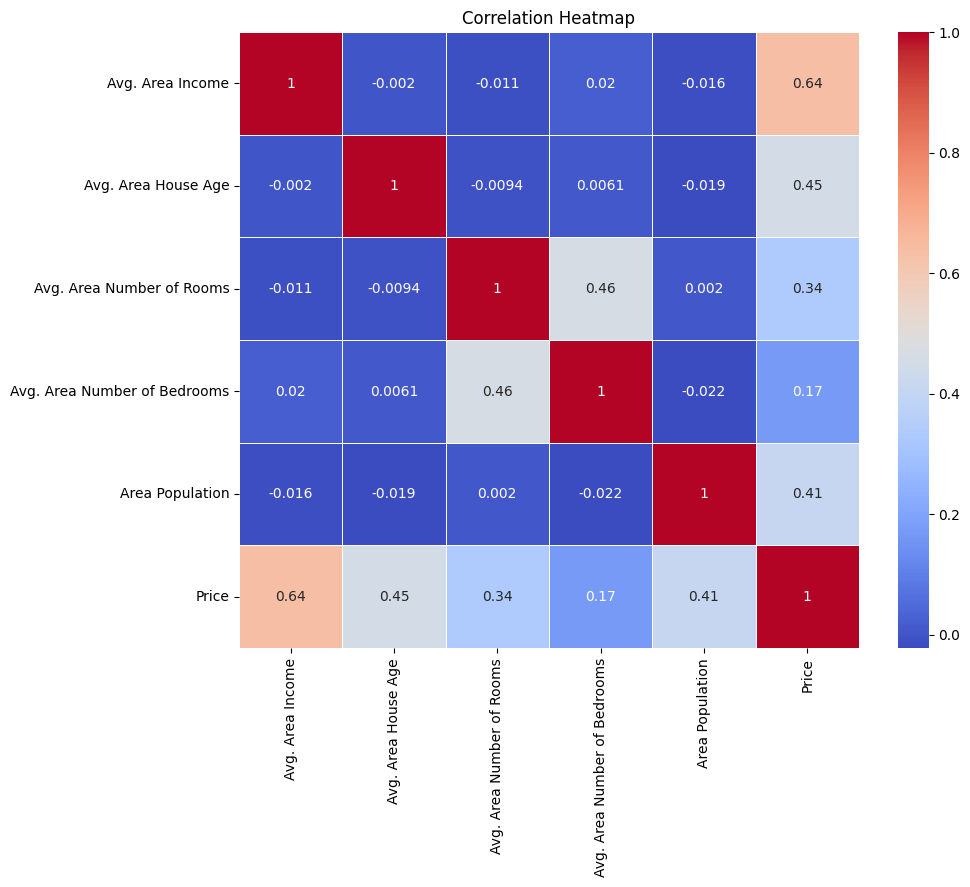


Ta thấy giá các ngôi nhà đã bán thường tập trung ở mức giá 0.5 đến 2.0 , và nhiều nhất là 0.8 đến 1.7

### Phân tích tương quan

Chúng ta sẽ sử dụng bản đồ nhiệt để kiểm tra độ tương quan giữa các cột:

*#Loại bỏ cột địa chỉ*HouseDF.drop(columns=['Address'], inplace=True)  
*# Phân tích tương quan.*plt.figure(figsize=(10, 8))  
sns.heatmap(HouseDF.corr(), annot=True, cmap='coolwarm', linewidths=0.5)  
plt.title('Correlation Heatmap')  
plt.show()

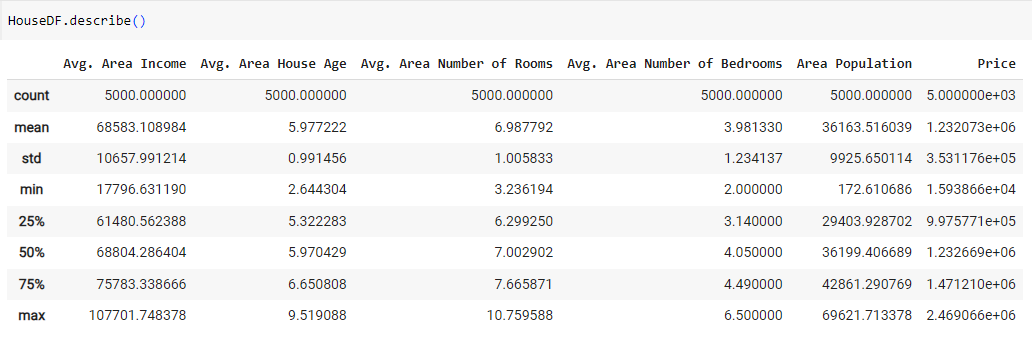


Qua đó, ta phân tích được các cột có giá trị tương quan như như thế nào với nhau. Về cơ bản, cột giá (price) có chút tương quan với các cột còn lại nhiều nhất, chứng tỏ các yếu tố đó có tác động ít nhiều lên giá nhà.

## 3.8 Đặc trưng

### Trích xuất đặc trưng

Đặc trưng số học :



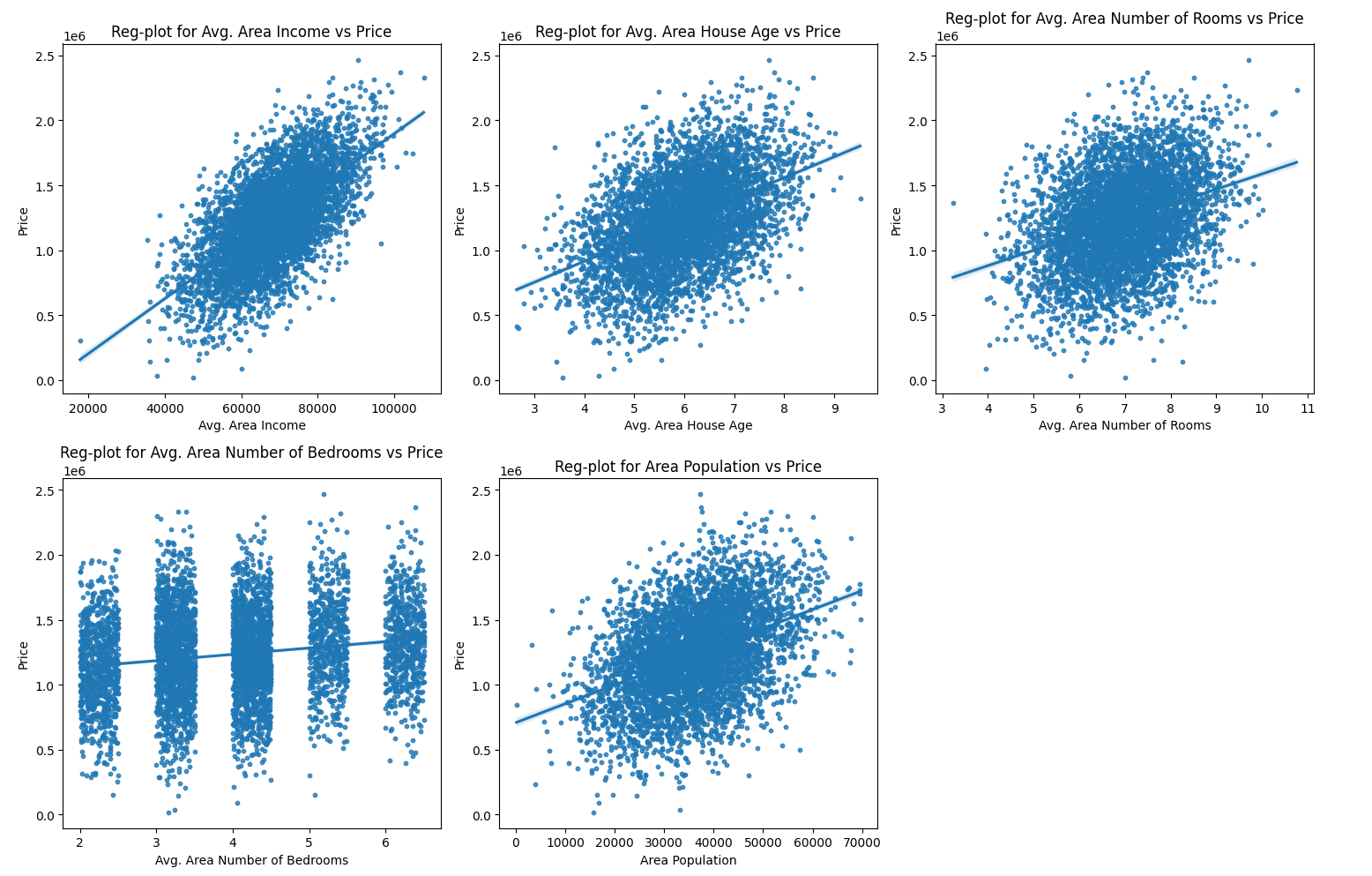
* Avg. Area Income: Thu nhập trung bình tại khu vực ngôi nhà đã bán là 68583
* Avg. Area House Age: Trung bình tuổi của một ngôi nhà đã bán 5.977
* Avg. Area Number of Rooms: Trung bình diện tích các phòng 6.98
* Avg. Area Number of Bedrooms: Trung bình diện tích phòng ngủ 3.98
* Area Population: Trung bình dân số tại khu vực bán nhà 36163.5
* Price: Trung bình giá ngôi nhà đã bán là l1232073

### Chọn lọc đặc trưng

plt.figure(figsize=(15, 10))  
for i, column in enumerate(x.columns, 1):  
 plt.subplot(2, 3, i)  
 sns.regplot(x=x[column], y=y, scatter\_kws={'s': 10})  
 plt.title(f'Reg-plot for {column} vs Price')  
plt.tight\_layout()  
plt.show()

Khi bạn chạy đoạn code để vẽ các reg-plot cho mỗi đặc trưng số học so với biến mục tiêu 'Price', bạn có thể quan sát mối quan hệ giữa từng đặc trưng và giá nhà. Dưới đây là một số điều mà bạn có thể quan sát được:

1. **Mối quan hệ tuyến tính:** Nếu reg-plot cho mỗi đặc trưng có hình dạng gần với đường thẳng, đó là một dấu hiệu của mối quan hệ tuyến tính giữa đặc trưng đó và giá nhà. Nếu đường hồi quy nghiêng lên, có thể có một tăng đáng kể giữa đặc trưng và giá nhà.
2. **Phân tán của dữ liệu:** Kích thước của phân tán dữ liệu quanh đường hồi quy có thể cho bạn biết về mức độ biến động của giá nhà dựa trên từng đặc trưng. Nếu phân tán lớn, có thể có nhiều yếu tố khác nhau đóng góp vào giá nhà.
3. **Outliers:** Reg-plot cũng có thể giúp bạn phát hiện các outliers, tức là các điểm dữ liệu không tuân theo xu hướng chung. Những điểm này có thể là những trường hợp đặc biệt cần được kiểm tra kỹ lưỡng.
4. **Mối quan hệ không tuyến tính:** Nếu đồ thị không tuân theo một đường thẳng, có thể có mối quan hệ không tuyến tính giữa đặc trưng và giá nhà.



## 3.9 Xây dựng mô hình dự đoán bằng thư viện Scikit – Learn

### Phân tách dữ liệu thành train và test

Bây giờ chúng ta hãy bắt đầu đào tạo mô hình hồi quy. Trước tiên, chúng ta sẽ cần tách dữ liệu của mình thành một mảng X chứa các tính năng cần đà tạo (các biến độc lập) và một mảng y với biến mục tiêu (biến phụ thuộc), trong trường hợp này là cột Giá. Chúng ta sẽ loại bỏ cột Địa chỉ vì nó chỉ có thông tin văn bản mà mô hình hồi quy tuyến tính không thể sử dụng

x = HouseDF[['Avg. Area Income','Avg. Area House Age','Avg. Area Number of Rooms','Avg. Area Number of Bedrooms','Area Population']]  
y = HouseDF[['Price']]  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.4, random\_state=101)

Giờ ta đã có hai biến x, y theo yêu cầu của mô hình, hai biến này dựa trên dữ liệu là dataset ta có được để đào tạo mô hình. Giờ là ta tách các biến trên thành giá trị train và test, hai giá trị này chúng ta sẽ luôn gặp và sử dụng trong qua trình xây dựng mô hình máy học.

Đầu tiên, ta từ thư viện Scikti – Learn model\_selection ta import train\_test\_split, phương thức này giúp ta tạo mô hình hồi quy

Sau đó ta tạo 4 biến, gồm X\_train, y\_train và X\_test, y\_test. Với đối số truyền vào là giá trị X, y ta đã lấy từ dữ liệu bên trên, test\_size trả về cho ta phần trăm dữ liệu được chia, ví dụ 0.4 tương ứng với dữ liệu được chia thành 40% giá trị là test, còn lại là dữ liệu train. random\_state bằng một số tương ứng nào đó để đảm bảo mỗi lần ta chạy lại mô hình, giá trị phân tách ngẫu nhiên nhận được là giống nhau, bạn có thể cho số nào bất kỳ.

Chúng ta tiếp tục chuẩn hóa đặc trưng của dữ liệu

scaler = StandardScaler()  
x\_train\_scaled = scaler.fit\_transform(x\_train)  
x\_test\_scaled = scaler.transform(x\_test)

### Tạo model và training

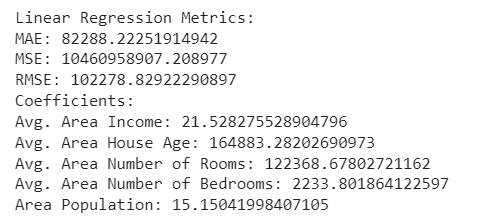
**Linear Regression**

Từ thư viện Scikit – Learn , linear\_model import module LinearRegression

*# Linear Regression*lm = LinearRegression()  
lm.fit(x\_train, y\_train)  
lm\_predictions = lm.predict(x\_test)  
lm\_coefficients = lm.coef\_  
print("Linear Regression Metrics:")  
print('MAE:', metrics.mean\_absolute\_error(y\_test, lm\_predictions))  
print('MSE:', metrics.mean\_squared\_error(y\_test, lm\_predictions))  
print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, lm\_predictions)))  
print('Coefficients:')  
for i, column in enumerate(x.columns):  
 print(f'{column}: {lm\_coefficients[i]}')

**Hệ số coeff**

Để đánh giá sực tác động của các tính năng ( các biến độc lập) lên kết quả đầu ra (biến phụ thuộc), ta sử dụng hệ số Coeff. Hệ số này cho ta biết khi giá trị biến độc lập thay đổi 1 đơn vị, thì giá trị đầu ra sẽ thay đổi như thế nào.



Diễn giải các hệ số trên:

* Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Avg. Area Income thì sẽ tăng $ 21, 52 trong giá nhà
* Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Avg. Area House Age thì sẽ tăng $164883.28 trong giá nhà
* Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Avg. Area Number of Rooms thì sẽ tăng $122368.67 trong giá nhà
* Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Avg. Area Number of Bedrooms thì sẽ tăng $2233.80 trong giá nhà
* Giữ tất cả các tính năng khác không thay đổi, khi tăng 1 đơn vị trong Cột: Area Population thì sẽ tăng $15.15 trong giá nhà

**Decision Tree**

*# Decision Tree*dt\_regressor = DecisionTreeRegressor(random\_state=42)  
dt\_regressor.fit(x\_train, y\_train)  
dt\_predictions = dt\_regressor.predict(x\_test)  
dt\_feature\_importances = dt\_regressor.feature\_importances\_  
  
print("\nDecision Tree Metrics:")  
print('MAE:', metrics.mean\_absolute\_error(y\_test, dt\_predictions))  
print('MSE:', metrics.mean\_squared\_error(y\_test, dt\_predictions))  
print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, dt\_predictions)))  
print('Feature Importances:')  
for i, column in enumerate(x.columns):  
 print(f'{column}: {dt\_feature\_importances[i]}')

**Random Forest**

*# Random Forest*rf\_regressor = RandomForestRegressor(n\_estimators=100, random\_state=42)  
rf\_regressor.fit(x\_train, y\_train)  
rf\_predictions = rf\_regressor.predict(x\_test)  
rf\_feature\_importances = rf\_regressor.feature\_importances\_  
  
print("\nRandom Forest Metrics:")  
print('MAE:', metrics.mean\_absolute\_error(y\_test, rf\_predictions))  
print('MSE:', metrics.mean\_squared\_error(y\_test, rf\_predictions))  
print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, rf\_predictions)))  
print('Feature Importances:')  
for i, column in enumerate(x.columns):  
 print(f'{column}: {rf\_feature\_importances[i]}')

### Tinh chỉnh mô hình và đánh giá mô hình tốt nhất

Sử dụng GridSearchCV để tinh chỉnh siêu tham số của mô hình Random Forest.

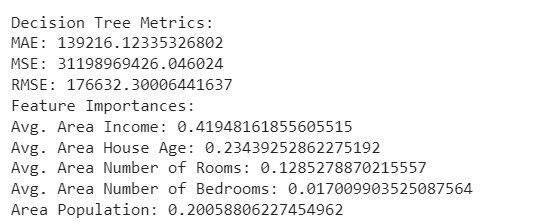
Đánh giá mô hình tốt nhất được tìm thấy từ GridSearchCV sử dụng các độ đo như MAE, MSE và RMSE.

In ra đặc tính quan trọng của các biến đầu vào trong mô hình tốt nhất.

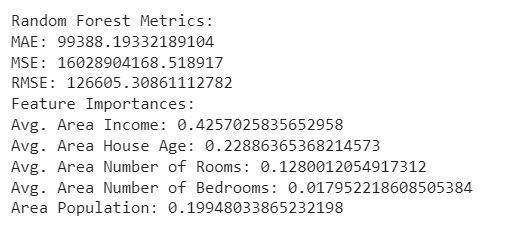
from sklearn.model\_selection import cross\_val\_score, GridSearchCV  
from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error  
from sklearn.pipeline import Pipeline  
from sklearn.compose import ColumnTransformer  
from sklearn.impute import SimpleImputer  
from sklearn.preprocessing import StandardScaler, OneHotEncoder  
from sklearn.ensemble import RandomForestRegressor  
  
*# Pipeline for preprocessing numerical features*numeric\_features = ['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population']  
numeric\_transformer = Pipeline(steps=[  
 ('imputer', SimpleImputer(strategy='mean')),  
 ('scaler', StandardScaler())  
])  
  
*# Pipeline for preprocessing categorical features if any*categorical\_features = [] *# Add categorical features if any*categorical\_transformer = Pipeline(steps=[  
 ('imputer', SimpleImputer(strategy='constant', fill\_value='missing')),  
 ('onehot', OneHotEncoder(handle\_unknown='ignore'))  
])  
  
*# Preprocessor combining the numeric and categorical transformers*preprocessor = ColumnTransformer(  
 transformers=[  
 ('num', numeric\_transformer, numeric\_features),  
 ('cat', categorical\_transformer, categorical\_features)  
 ])  
  
*# Append the preprocessor to the model pipeline*pipeline = Pipeline(steps=[('preprocessor', preprocessor),  
 ('model', RandomForestRegressor())])  
  
*# Define parameters grid for tuning*param\_grid = {  
 'model\_\_n\_estimators': [100, 200, 300],  
 'model\_\_max\_depth': [None, 5, 10, 15],  
 *# Add more parameters to search if necessary*}  
  
*# Perform Grid Search Cross Validation*grid\_search = GridSearchCV(pipeline, param\_grid, cv=5, scoring='neg\_mean\_squared\_error')  
grid\_search.fit(x\_train, y\_train)  
  
*# Get the best model from Grid Search CV*best\_model = grid\_search.best\_estimator\_  
  
*# Evaluate the best model*y\_pred = best\_model.predict(x\_test)  
print('Best Model Evaluation:')  
print('MAE:', mean\_absolute\_error(y\_test, y\_pred))  
print('MSE:', mean\_squared\_error(y\_test, y\_pred))  
print('RMSE:', np.sqrt(mean\_squared\_error(y\_test, y\_pred)))  
  
*# Get feature importances from the best model*feature\_importances = best\_model.named\_steps['model'].feature\_importances\_  
for i, column in enumerate(x.columns):  
 print(f'{column}: {feature\_importances[i]}')

### Features Importances

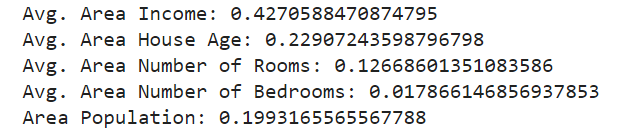
Feature importances đo lường mức độ quan trọng của mỗi biến đầu vào đối với việc đưa ra dự đoán. Cụ thể, nó cho biết biến nào đóng góp nhiều nhất vào quá trình quyết định của mô hình.



*Decision Tree*



*Random Forest*



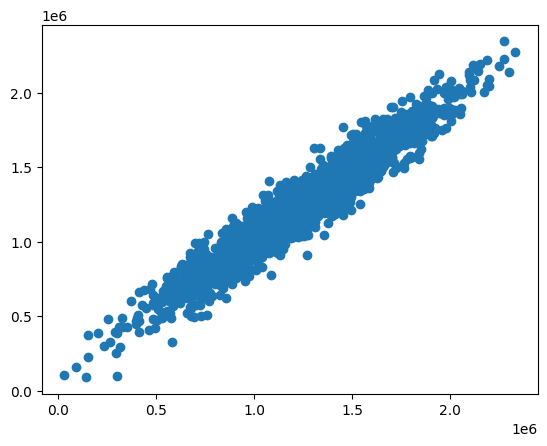
*Best Model Evaluation*

### Dự đoán và đánh giá mô hình

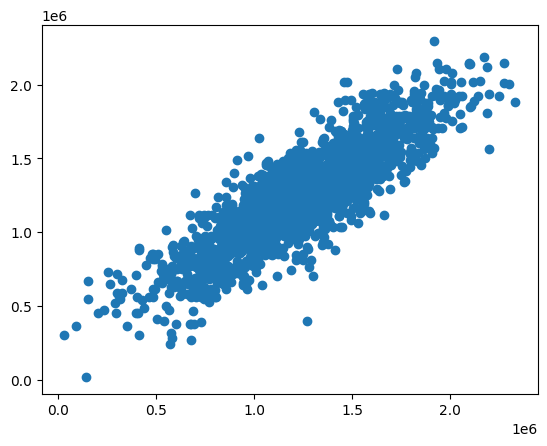
Để dự đoán và kiểm tra mô hình, ta sử dụng dữ liệu test bên trên mà ta đã tách ra. Trong đó, X\_test là các tính năng mà mô hình chưa biết, y\_test là kết quả biết trước để ta so sánh với kết quả dự đoán từ X\_test.

Lấy kết quả dự đoán từ X\_test, ta dùng phương thức predict() truyền đối số X\_test vào

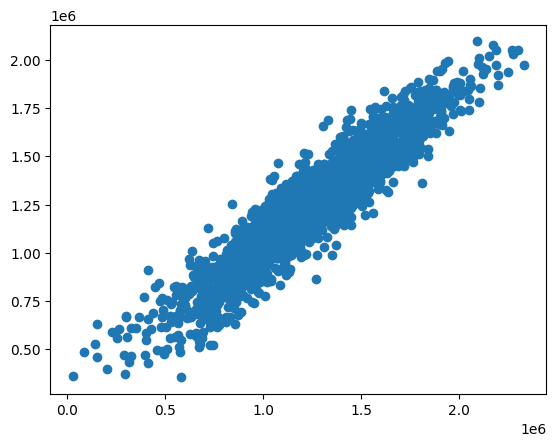
lm\_predictions = lm.predict(x\_test)  
plt.scatter(y\_test,lm\_predictions)



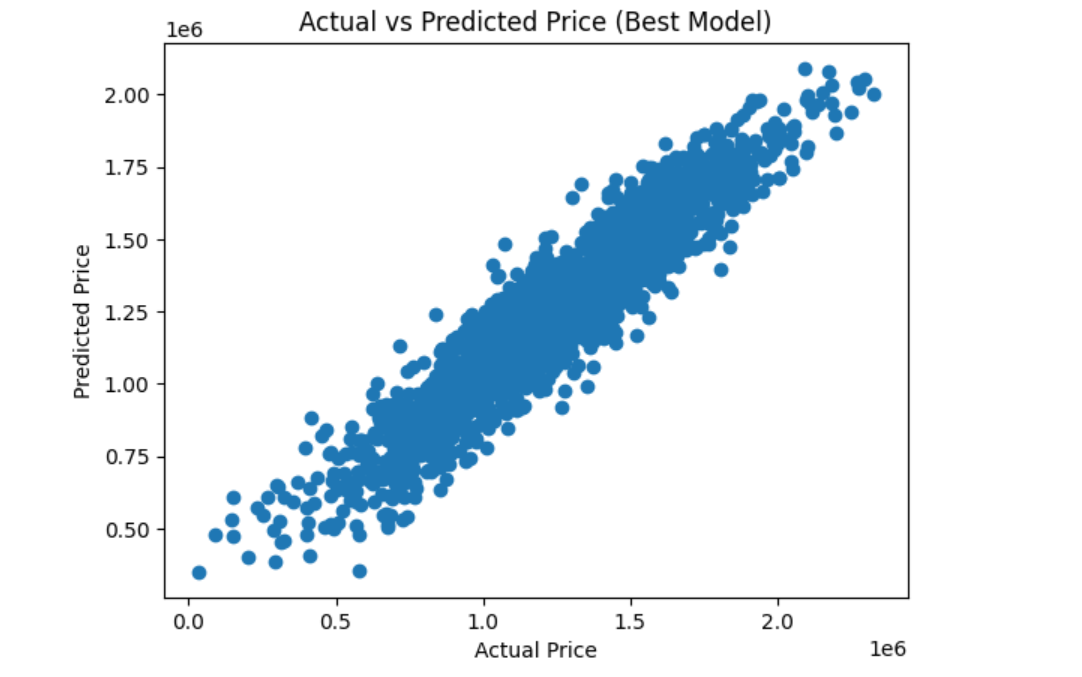
dt\_predictions = dt\_regressor.predict(x\_test)  
plt.scatter(y\_test,dt\_predictions)



rf\_predictions = rf\_regressor.predict(x\_test)  
plt.scatter(y\_test,rf\_predictions)



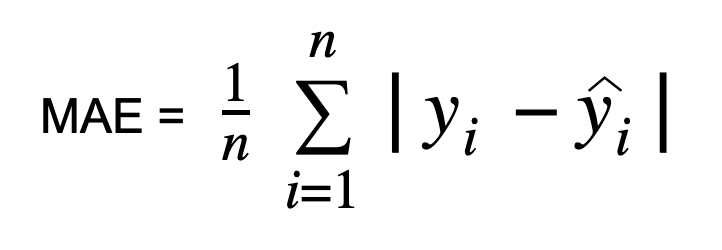
*# Visualize scatter plot for the best model predictions*plt.scatter(y\_test, y\_pred)  
plt.xlabel('Actual Price')  
plt.ylabel('Predicted Price')  
plt.title('Actual vs Predicted Price (Best Model)')  
plt.show()



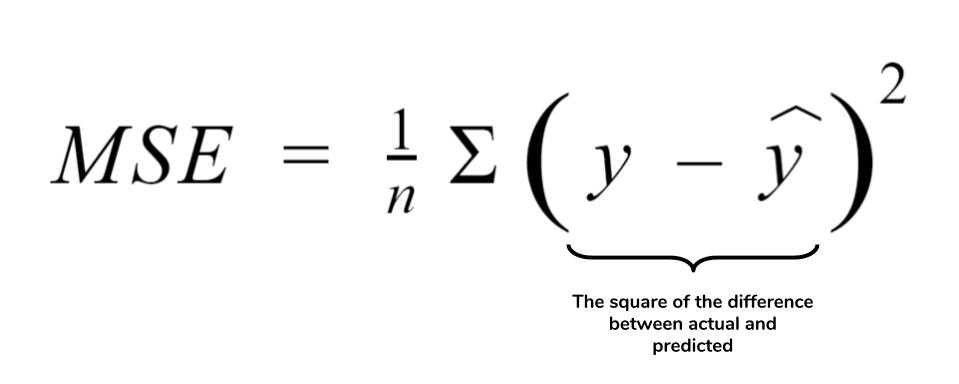
Bây giờ, ta hãy tìm số liệu cụ thể để có cái nhìn chính xác hơn dựa trên số liệu phân tích. Ta dựa vào chỉ số đánh giá hồi quy

Chúng ta có ba chỉ số đánh giá hồi quy để có số liệu chính xác:

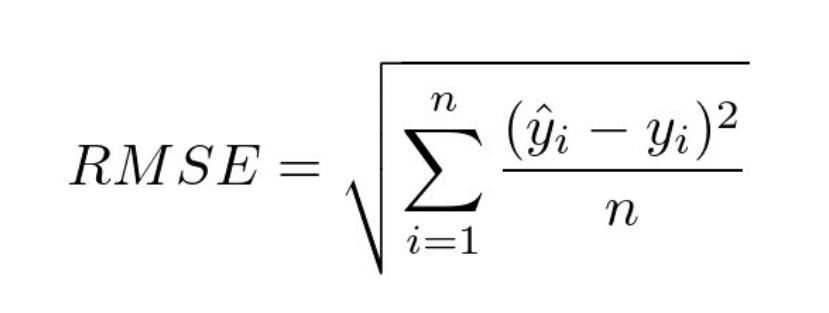
**Mean Absolute Error** (MAE): MAE là một phương pháp đo lường sự khác biệt (độ chênh lệch giá trị) giữa hai biến liên tục. Giả sử rằng *X* và *Y* là hai biến liên tục thể hiện kết quả dự đoán của mô hình và kết quả thực tế, đây là chỉ số dễ hiểu nhất, vì đó là giá trị chênh lệch trung bình và được xác định bằng công thức:



**Mean Squared Error** (MSE): là giá trị trung bình của bình phương sai số (Hàm mất mát), là sự khác biệt giữa các giá trị được mô hình dự đoán và giá trị thực. MSE cũng được gọi là một hàm rủi ro, tương ứng với giá trị kỳ vọng của sự mất mát sai số bình phương hoặc mất mát bậc hai chỉ số này phổ biến hơn chỉ số MAE bên trên, được xác định bằng công thức:

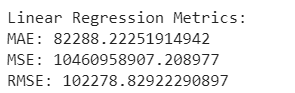


**Root Mean Squared Error** (RMSE): là căn bậc hai của giá trị trung bình của các sai số bình phương (MSE). Thông thường, ta thường dùng chỉ số này để xác định giá trị chênh lệch trung bình giữa giá dự đoán và giá trị test ban đầu, được xác định bằng công thức:

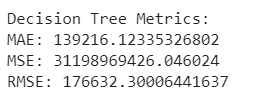


Đầu tiên, chúng ta sẽ lấy tất cả các chỉ số hồi quy đã nói ở trên

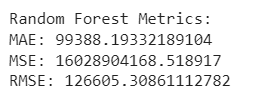
print("Linear Regression Metrics:")  
print('MAE:', metrics.mean\_absolute\_error(y\_test, lm\_predictions))  
print('MSE:', metrics.mean\_squared\_error(y\_test, lm\_predictions))  
print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, lm\_predictions)))



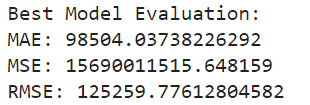
print("Decision Tree Metrics:")  
print('MAE:', metrics.mean\_absolute\_error(y\_test, dt\_predictions))  
print('MSE:', metrics.mean\_squared\_error(y\_test, dt\_predictions))  
print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, dt\_predictions)))



print("Random Forest Metrics:")  
print('MAE:', metrics.mean\_absolute\_error(y\_test, rf\_predictions))  
print('MSE:', metrics.mean\_squared\_error(y\_test, rf\_predictions))  
print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, rf\_predictions)))



*# Evaluate the best model*y\_pred = best\_model.predict(x\_test)  
print('Best Model Evaluation:')  
print('MAE:', mean\_absolute\_error(y\_test, y\_pred))  
print('MSE:', mean\_squared\_error(y\_test, y\_pred))  
print('RMSE:', np.sqrt(mean\_squared\_error(y\_test, y\_pred)))



Ta thấy, sử dụng chỉ số RMSE, cho thấy giá trị chênh lệch trung bình của giá dự đoán từ mô hình và giá trị thực tế :

* Linenear Regression : 102.278 USD
* Decision Tree : 176.632 USD
* Random Forest : 126.605 USD
* Best Model Evaluation : 125.259 USD

# CHƯƠNG 4 : KẾT LUẬN

## 4.1. Tổng quan dữ liệu

**Tiền xử lý dữ liệu:** Quá trình xử lý dữ liệu bao gồm việc loại bỏ các dòng có giá trị bị bác bỏ, giúp đảm bảo tính chính xác và độ tin cậy của kết quả phân tích.

**Phân tích dữ liệu** : Quá trình phân tích dữ liệu bao gồm các bước phân tích đa biến , phân tích đơn biến , phân tích tương quan

**Phân phối đặc trưng:** Đồ thị phân phối đã giúp chúng ta hiểu rõ về phân phối của từng đặc trưng, cung cấp thông tin về biến động và xu hướng của dữ liệu.

**Mối quan hệ đặc trưng – giá nhà:** Các reg-plot đã thể hiện mối quan hệ giữa từng đặc trưng số học và giá nhà. Việc này giúp xác định những đặc trưng có ảnh hưởng đáng kể đến giá nhà.

## 4.2. Mô hình Linenear Regression , Decision Tree , Random Forest

**Chia dữ liệu:** Dữ liệu đã được chia thành tập train, validation, và test để đảm bảo kiểm tra hiệu suất mô hình trên các tập dữ liệu độc lập.

**Chuẩn hóa đặc trưng:** Đặc trưng đã được chuẩn hóa bằng StandardScaler để đảm bảo đồng nhất và hiệu quả trong quá trình huấn luyện mô hình.

**Huấn luyện và đánh giá mô hình:** Mô hình hồi quy tuyến tính đã được huấn luyện trên tập train và đánh giá hiệu suất trên tập validation và test bằng cách sử dụng độ đo R-squared.

**Hiểu biết về mối quan hệ:** Các hệ số của đặc trưng trong mô hình đã được giải thích, giúp hiểu rõ hơn về cách từng đặc trưng ảnh hưởng đến giá nhà.

## 4.3. Đánh giá và kết luận

**Đánh giá mô hình:** Các mô hình đã được đánh giá dựa trên các chỉ số đánh giá như MAE, MSE và RMSE. Linear Regression có hiệu suất tốt nhất với RMSE thấp nhất, trong khi Decision Tree cho kết quả tệ nhất với RMSE cao nhất. Hệ số của Linear Regression và độ quan trọng của các đặc trưng trong Decision Tree và Random Forest cũng được cung cấp để hiểu rõ hơn về mối quan hệ giữa các biến và giá nhà.

**Phân tích lỗi:** Phân tích lỗi trên tập test giúp xác định sự chênh lệch giữa giá nhà thực tế và dự đoán, giúp cải thiện mô hình trong tương lai.

**Hạn chế và hướng phát triển:** Bài báo đã đề cập đến những hạn chế có thể xuất phát từ dữ liệu và mô hình, đồng thời đề xuất các hướng phát triển tiềm năng như sử dụng mô hình phức tạp hơn, bổ sung dữ liệu mới, hoặc tối ưu hóa siêu tham số.

Bài báo này mang lại cái nhìn tổng quan về việc áp dụng mô hình hồi quy tuyến tính (Linenear Regression) , Decision Tree , Random Forest cho dữ liệu giá nhà, đồng thời đề xuất cơ hội nghiên cứu và phát triển trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] M. L. Giger, "Machine Learning in Medical Imaging," \*Radiographics,\* vol. 37, no. 1, pp. 130-145, 2017.

Link: <https://doi.org/10.1148/rg.2017160130>

[2] M. Leo, S. Sharma, and K. Maddulety, "Machine Learning in Banking Risk Management: A Literature Review," Journal of Risk and Financial Management, vol. 7, no. 1, p. 29, 2019.

Link : <https://www.mdpi.com/2227-9091/7/1/29>

[3] A. Ahramovich, "Machine Learning in Education: 10 Use Cases, Examples, and Benefits," .

Link : [https://www.itransition.com/machine-learning/education.](https://www.itransition.com/machine-learning/education.%20)

[4] "Ecommerce Machine Learning Brings the Future to the Present of Online Shopping," .

Link : [https://www.bigcommerce.com/articles/ecommerce/machine-learning/.](https://www.bigcommerce.com/articles/ecommerce/machine-learning/)

[5] Siuly, Siuly, Zhang, Yanchun, & Xu, Yue, "Medical Big Data: Promise and Challenges," IEEE Access, vol. 5, pp. 26592-26633, 2017.

[6] Thomas, Lyn C., "Machine Learning and Credit Scoring: A Review," Proceedings of the IEEE, vol. 104, pp. 1915-1932, 2016.

[7] Romero, Cristóbal, & Ventura, Sebastián, "Educational Data Mining: A Review of the State of the Art," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, pp. 1992-2012, 2018.

[8] Tsiptsis, Konstantinos, & Chorianopoulos, Konstantinos, "Data Mining Techniques in Customer Relationship Management: A Literature Review and Classification," Expert Systems with Applications, vol. 39, pp. 7420-7432, 2012.