**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỒ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC**

**TRÍ TUỆ NHÂN TẠO**

**THỰC HIỆN GIẢI THUẬT HỒI QUY TUYẾN TÍNH VỚI HAI TẬP DỮ LIỆU THERMOSTAT SALES VÀ AIR PASSENGERS**

Giảng viên hướng dẫn: ThS**.Nguyễn Đức Vinh**

Sinh viên thực hiện:

Nguyễn Anh Huy ---------------

***Thành phố Hồ Chí Minh, tháng 07 năm 2025***

**LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Nguyễn Đức Vinh đã tận tình giảng dạy, truyền đạt những kiến thức và kinh nghiệm quý báu cho chúng em trong suốt quá trình học tập môn trí tuệ nhân tạo.

Trong suốt thời gian chúng em làm học tập những kiến thức về môn này và thực hiện bài báo cáo, thầy đã hết lòng giúp đỡ chúng em, chỉ ra cho hướng đi để chúng em có thể hoàn thành tốt nhất bài báo cáo này.

Trong quá trình thực hiện đề tài, dù chúng em đã cố gắng hoàn thiện đề tài một cách tốt nhất nhưng do thời gian và kiến thức còn hạn chế nên sẽ không tránh khỏi những thiếu sót nhất định, chúng em rất mong nhận được sự cảm thông, chia sẽ đến từ những đóng góp ý kiến chân thành từ thầy.

Chúng em xin chân thành cảm ơn thầy!

Mục lục

[DANH MỤC HÌNH i](#_Toc203918781)

[DANH MỤC BẢNG ii](#_Toc203918782)

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc203918783)

[1.1. Giới thiệu đề tài 1](#_Toc203918784)

[1.2. Mục Tiêu 1](#_Toc203918785)

[1.3. Nội dung đề tài 2](#_Toc203918786)

[1.4. Giới hạn đề tài 2](#_Toc203918787)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 3](#_Toc203918788)

[2.1. Định nghĩa về Hồi quy tuyến tính 3](#_Toc203918789)

[2.1.1. Định Nghĩa 3](#_Toc203918790)

[2.1.2. Phân loại Linear Regression 3](#_Toc203918791)

[2.1.3. Mục tiêu của Linear Regression 4](#_Toc203918792)

[2.1.4. Các phương pháp tìm nghiệm 5](#_Toc203918793)

[2.1.5. Các giả định cơ bản của Linear Regression 5](#_Toc203918794)

[2.1.6. Các chỉ số đánh giá mô hình 7](#_Toc203918795)

[2.1.7. Regularization (Chống overfitting) 7](#_Toc203918796)

[2.1.8. Ứng dụng thực tế của Linear Regression 7](#_Toc203918797)

[2.2. Hàm mất mát trong bài toán hồi quy tuyến tính 8](#_Toc203918798)

[2.2.1. Tầm quan trọng của Loss Function 8](#_Toc203918799)

[2.2.2. Mean Squared Error (MSE) 9](#_Toc203918800)

[2.2.3. Root-Mean-Square Error(RMSE) 9](#_Toc203918801)

[2.2.4. Hình dung RMSE 10](#_Toc203918802)

[2.2.5. R-squared 11](#_Toc203918803)

[2.2.6. R-squared được tính như thế nào? 11](#_Toc203918804)

[2.2.7. R-squared độ phù hợp 12](#_Toc203918805)

[2.2.8. Hạn chế của việc sử dụng phương pháp R-squared 13](#_Toc203918806)

[2.3. Tập dữ liệu 13](#_Toc203918807)

[2.3.1. Nguồn dữ liệu 13](#_Toc203918808)

[2.3.2. Mô tả dữ liệu 14](#_Toc203918809)

[2.3.3. Tiền xử lý dữ liệu 15](#_Toc203918810)

[2.3.4. Trực quan dữ liệu 15](#_Toc203918811)

[CHƯƠNG 3. Phương pháp thực hiện 22](#_Toc203918812)

[3.1. Quá trình thực hiện 22](#_Toc203918813)

[3.2. Thực hiện với tập dữ liệu Thermostat Sales 23](#_Toc203918814)

[3.2.1. Thực hiện với hồi quy tuyến tính thủ công 23](#_Toc203918815)

[3.2.2. Thực hiện với hồi quy tuyến tính dùng thư viện 25](#_Toc203918816)

[3.2.3. Đánh giá mô hình với tập dữ liệu trên hai hướng xây dựng mô hình: 28](#_Toc203918817)

[3.3. Thực hiện với tập dữ liệu Air Passengers 31](#_Toc203918818)

[3.3.1. Thực hiện với hồi quy tuyến tính thủ công 31](#_Toc203918819)

[3.3.2. Thực hiện với hồi quy tuyến tính dùng thư viện 33](#_Toc203918820)

[3.3.3. Đánh giá mô hình với tập dữ liệu trên hai hướng xây dựng mô hình: 36](#_Toc203918821)

[3.4. Đánh giá kết quả của mô hình trên hai tập dữ liệu 39](#_Toc203918822)

[CHƯƠNG 4. KẾT LUẬN 41](#_Toc203918823)

[4.1. Kết quả đạt được 41](#_Toc203918824)

[4.2. Ứng dụng thực tế 41](#_Toc203918825)

[4.3. Ưu và nhược điểm 41](#_Toc203918826)

[4.3.1. Ưu điểm 41](#_Toc203918827)

[4.3.2. Nhược điểm 42](#_Toc203918828)

[4.4. Hướng mở rộng trong tương lai 42](#_Toc203918829)

[CHƯƠNG 5. Mã nguồn 42](#_Toc203918830)

[5.1. Cài đặt các thư viện cần thiết 43](#_Toc203918831)

[5.2. Đọc và tiền xử lý dữ liệu 44](#_Toc203918832)

[5.3. Chia dữ liệu thành train và test 44](#_Toc203918833)

[5.4. Trực quan dữ liệu 45](#_Toc203918834)

[5.4.1. Cho tập dữ liệu Air Passengers 45](#_Toc203918835)

[5.4.2. Cho dữ liệu Thermostat Sales 47](#_Toc203918836)

[5.5. Xây dựng mô hình hồi quy thủ công 49](#_Toc203918837)

[5.5.1. Xây dựng mô hình 49](#_Toc203918838)

[5.5.2. Huấn luyện 50](#_Toc203918839)

[5.6. Kết quả 51](#_Toc203918840)

[5.6.1. Hiển thị các giá trị mất mát: 51](#_Toc203918841)

[5.6.2. Trực quan đường hồi quy trên tập huấn luyện 52](#_Toc203918842)

[5.6.3. Trực quan đường hồi quy trên tập kiểm tra 53](#_Toc203918843)

[5.7. Xây dựng mô hình hồi quy với thư viện 54](#_Toc203918844)

[5.7.1. Khởi tạo và huấn luyện mô hình: 54](#_Toc203918845)

[5.7.2. Tính toán hàm mất mát và hiển thị 54](#_Toc203918846)

[5.7.3. Trực quan đường hồi quy 55](#_Toc203918847)

[5.8. So sánh kết quả từ mô hình thủ công và mô hình từ thư viện 56](#_Toc203918848)

[5.8.1. So sánh trên biểu đồ trực quan của đường hồi quy 56](#_Toc203918849)

[5.8.2. So sánh kết quả dựa trên hàm mất mát 57](#_Toc203918850)

[CHƯƠNG 6. Tài liệu tham khảo 59](#_Toc203918851)

# DANH MỤC HÌNH

[Hình 2.1: RMSE 10](#_Toc203917410)

[Hình 2.2: R-Squared 12](#_Toc203917411)

[Hình 2.3: R-Squared cho res 13](#_Toc203917412)

[Hình 2.4: Biểu đồ phân bố cho Thermostat Sales 16](#_Toc203917413)

[Hình 2.5: Biều đồ cột cho Thermostat Sales 17](#_Toc203917414)

[Hình 2.6: Biểu đồ hộp cho Thermostat sales 18](#_Toc203917415)

[Hình 2.7: Phân bố dữ liệu cho Air Passengers 19](#_Toc203917416)

[Hình 2.8: Phân bố dạng cột dữ liệu cho Air Passengers 20](#_Toc203917417)

[Hình 2.9: Biểu đồ hộp cho Air Passengers 21](#_Toc203917418)

[Hình 3.1: Sơ đồ hoạt động 22](#_Toc203917419)

[Hình 3.2: Thermostat Sales thủ công hồi quy tuyến tính trên tập huấn luyện 24](#_Toc203917420)

[Hình 3.3: Thermostat Sales thủ công hồi quy tuyến tính trên tập kiểm tra 25](#_Toc203917421)

[Hình 3.4: Thermostat Sales thư viện hồi quy tuyến tính trên tập huấn luyện 26](#_Toc203917422)

[Hình 3.5: Thermostat Sales thư viện hồi quy tuyến tính trên tập kiểm tra 27](#_Toc203917423)

[Hình 3.6: Thermostat Sales so sánh hồi quy tuyến tính trên tập huấn luyện 29](#_Toc203917424)

[Hình 3.7: Thermostat Sales so sánh hồi quy tuyến tính trên tập kiểm tra 30](#_Toc203917425)

[Hình 3.8: Air Passengers thủ công hồi quy tuyến tính trên tập huấn luyện 32](#_Toc203917426)

[Hình 3.9: Air Passengers thủ công hồi quy tuyến tính trên tập kiểm tra 33](#_Toc203917427)

[Hình 3.10: Air Passengers thư viện hồi quy tuyến tính trên tập huấn luyện 34](#_Toc203917428)

[Hình 3.11: Air Passengers thư viện hồi quy tuyến tính trên tập kiểm tra 35](#_Toc203917429)

[Hình 3.12: Air Passengers so sánh hồi quy tuyến tính trên tập huấn luyện 37](#_Toc203917430)

[Hình 3.13: Air Passengers so sánh hồi quy tuyến tính trên tập kiểm tra 38](#_Toc203917431)

# DANH MỤC BẢNG

[Bảng 2.1: Mô tả dữ liệu 14](#_Toc203918006)

# GIỚI THIỆU

## Giới thiệu đề tài

Trong bối cảnh công nghệ phát triển nhanh chóng, trí tuệ nhân tạo (AI) đang từng bước len lỏi vào mọi mặt của đời sống hiện đại. Từ những công việc thường ngày đến các hoạt động chuyên môn phức tạp, con người ngày càng dựa vào sự hỗ trợ mạnh mẽ và hiệu quả của các thuật toán học máy và các mô hình thông minh. Những tiến bộ trong lĩnh vực AI không chỉ giúp giảm tải công việc thủ công mà còn mở ra nhiều hướng phát triển mới, đưa hiệu suất lao động và chất lượng sống lên một tầm cao mới.

Giữa làn sóng phát triển mạnh mẽ ấy, việc nắm bắt và hiểu rõ các thuật toán nền tảng là điều kiện tiên quyết để tiếp cận và ứng dụng hiệu quả trí tuệ nhân tạo. Trong khuôn khổ học phần Trí tuệ nhân tạo, nhóm chúng em lựa chọn tìm hiểu một trong những thuật toán căn bản và có ảnh hưởng sâu rộng nhất – Hồi quy tuyến tính (Linear Regression).

Hồi quy tuyến tính không chỉ là một công cụ thống kê đơn thuần mà còn là nền tảng quan trọng cho nhiều mô hình học máy hiện đại. Với cấu trúc đơn giản nhưng mang tính ứng dụng cao, thuật toán này cho phép mô hình hóa mối quan hệ giữa các biến số, từ đó phục vụ cho việc dự đoán, phân tích xu hướng, và hỗ trợ ra quyết định trong thực tế.

Trong phần trình bày này, nhóm sẽ đi vào lí thuyết hoạt của hồi quy tuyến tính, làm rõ các khái niệm cốt lõi thông qua lý thuyết và các ví dụ cụ thể. Bên cạnh đó, mô hình sẽ được triển khai trên các tập dữ liệu thực tế để kiểm nghiệm khả năng áp dụng vào các bài toán đời sống.

Cả 2 bài toán đều là bài toán học có giám sát (Supervised Learning)

* Input: Tập dữ liệu đầu vào đã được chuẩn hóa
* Output: Thực hiện đường hồi quy trên tập dữ liệu

## Mục Tiêu

Trình bày lý thuyết cơ bản về hồi quy tuyến tính, bao gồm: công thức mô hình, cách ước lượng các tham số, các giả định cần thiết, cũng như ý nghĩa thống kê của từng thành phần.

Phân tích các phương pháp đánh giá hiệu quả mô hình như sai số trung bình bình phương (MSE), sai số căn bậc hai (RMSE), và hệ số xác định (R²).

Triển khai mô hình hồi quy tuyến tính trên một số tập dữ liệu thực tế nhằm kiểm chứng khả năng dự đoán và giải thích dữ liệu.

So sánh kết quả giữa việc tính toán thủ công và sử dụng thư viện Python (như scikit-learn) để từ đó làm rõ quy trình triển khai mô hình trong thực tế.

Bài toán áp dụng thuật toán Hồi quy tuyến tính (Linear Regression) để dự báo giá trị dựa trên chuỗi dữ liệu đã cho:

* Thermostat Sales: Dự báo doanh số bán máy điều hòa theo tuần
* Air Passengers: Dự báo số lượng hàng khách đi máy bay theo tháng

Cả 2 bài toán đều là bài toán học có giám sát (Supervised Learning)

* Input: Tập dữ liệu đầu vào đã được chuẩn hóa
* Output: Thực hiện đường hồi quy trên tập dữ liệu

## Nội dung đề tài

* Tổng quan về dữ liệu:
* Phân tích, đánh giá cụ thể hai tập dữ liệu được cung
* Thu thập và xử lí dữ liệu:
* Thực hiện các bước chuẩn hóa và tiền xử lý dữ liệu
* Xây dựng mô hình dự đoán:
* Tiến hành thực hiện xây dựng mô hình Hồi quy tuyến tính thủ công
* Tiến hành thực hiện mô hình Hồi quy tuyến tính bằng thư viện
* Thực hiện áp dụng hai tập dữ liệu được cung cấp và đã tiền xử lí cho cả hai mô hình được xây dựng

## Giới hạn đề tài

* Nhóm chỉ dùng những dữ liệu được cung cấp không mở rộng thêm dữ liệu bên ngoài
* Không tiến hành mở rộng với các thuật toán hồi quy tuyến tính liên quan
* Không đào sâu vào các công thức thuần của mô hình

# CƠ SỞ LÝ THUYẾT

## Định nghĩa về Hồi quy tuyến tính

### Định Nghĩa

Linear Regression (Hồi quy tuyến tính) là một trong những thuật toán cơ bản và phổ biến nhất trong lĩnh vực học máy (machine learning) và thống kê. Nó thuộc nhóm học có giám sát (supervised learning), với mục tiêu chính là dự đoán một giá trị đầu ra (biến phụ thuộc y) liên tục dựa trên một hoặc nhiều biến đầu vào (biến độc lập x).

Công thức tổng quát của mô hình hồi quy tuyến tính nhiều biến:

ŷ = w₀ + w₁x₁ + w₂x₂ + ... + wₙxₙ

Trong đó:

- ŷ: giá trị dự đoán (predicted output)

- xⱼ: biến đầu vào thứ j (feature)

- wⱼ: hệ số hồi quy tương ứng với xⱼ (slope/weight)

- w₀: hệ số chệch (intercept/bias), đại diện cho giá trị dự đoán khi các xⱼ = 0

Linear Regression có thể được áp dụng trong nhiều bài toán thực tế khi có mối quan hệ tuyến tính giữa các biến, và nó cũng được dùng như một baseline đơn giản trong các bài toán phức tạp hơn.

### Phân loại Linear Regression

Linear Regression có thể được chia thành nhiều loại khác nhau dựa trên số lượng biến đầu vào, hình thức của mối quan hệ, và sự xuất hiện của các kỹ thuật regularization:

• Simple Linear Regression (Hồi quy tuyến tính đơn):

- Mô hình có đúng một biến đầu vào x và một biến đầu ra y.

- Dạng phương trình: ŷ = w₀ + w₁x

- Mục tiêu là tìm đường thẳng tốt nhất biểu diễn mối quan hệ giữa x và y trên mặt phẳng 2D.

• Multiple Linear Regression (Hồi quy tuyến tính đa biến):

- Mô hình có nhiều biến đầu vào x₁, x₂, ..., xₙ.

- Phương trình: ŷ = w₀ + w₁x₁ + w₂x₂ + ... + wₙxₙ

- Được sử dụng khi nhiều yếu tố ảnh hưởng đến kết quả cần dự đoán.

• Polynomial Regression (Hồi quy đa thức):

- Mở rộng của linear regression bằng cách thêm vào các đặc trưng phi tuyến như x², x³, ..., xᵈ.

- Dù đặc trưng phi tuyến, mô hình vẫn tuyến tính theo hệ số w.

- Dùng để mô hình hóa quan hệ cong giữa đầu vào và đầu ra.

• Regularized Linear Regression (Hồi quy có phạt):

- Bao gồm Ridge Regression (L2), Lasso Regression (L1), và Elastic Net.

- Thêm vào hàm mất mát một thành phần phạt nhằm giảm overfitting và cải thiện khả năng tổng quát.

- Đặc biệt hữu ích khi số lượng biến lớn hoặc có đa cộng tuyến giữa các biến đầu vào.

### Mục tiêu của Linear Regression

Mục tiêu chính của hồi quy tuyến tính là xây dựng một mô hình có thể dự đoán chính xác giá trị của biến phụ thuộc y dựa vào các biến độc lập x. Để làm điều đó, ta cần tìm bộ trọng số w sao cho mô hình dự đoán ŷ gần với y thực tế nhất.

Mức độ sai lệch giữa ŷ và y được đo bằng một hàm mất mát (loss function). Hàm phổ biến nhất là:

Mean Squared Error (MSE):

L(w) = (1/2N) \* Σ (yᵢ - xᵢᵀw)²

Giải thích:

- N: số lượng mẫu dữ liệu

- yᵢ: giá trị thực tế

- xᵢᵀw: giá trị dự đoán

- L(w): tổng sai số bình phương, chia trung bình để chuẩn hó

Mục tiêu học là tối thiểu hóa L(w), tức tìm w sao cho sai số giữa dự đoán và thực tế là nhỏ nhất trên toàn bộ tập huấn luyện.

Ngoài MSE, một số hàm mất mát khác có thể dùng là MAE (mean absolute error) hoặc Huber loss (để giảm ảnh hưởng của outliers).

### Các phương pháp tìm nghiệm

Có nhiều cách để tìm nghiệm (bộ trọng số w) cho mô hình Linear Regression:

• Phương pháp giải tích (Analytical Solution – Normal Equation):

- Giải trực tiếp phương trình đạo hàm bằng 0 để tìm nghiệm tối ưu:

w = (XᵀX)⁻¹Xᵀy

- Ưu điểm: nhanh, chính xác với dữ liệu nhỏ.

- Nhược điểm: không dùng được nếu XᵀX không khả nghịch hoặc dữ liệu rất lớn.

• Gradient Descent (GD):

- Thuật toán tối ưu số học lặp, cập nhật w theo hướng giảm dần của đạo hàm:

w ← w - η \* ∇L(w)

- Phù hợp khi số chiều lớn hoặc dữ liệu lớn.

- Cần chọn learning rate η phù hợp để đảm bảo hội tụ.

• Stochastic Gradient Descent (SGD):

- Cập nhật w sau mỗi mẫu hoặc mini-batch thay vì toàn bộ dữ liệu.

- Giúp giảm chi phí tính toán, tránh mắc kẹt local minima.

- Phù hợp với online learning hoặc dữ liệu rất lớn.

### Các giả định cơ bản của Linear Regression

Linear Regression hoạt động dựa trên một số giả định quan trọng. Nếu các giả định này không được đáp ứng, kết quả mô hình có thể bị sai lệch và không còn đáng tin cậy. Các giả định bao gồm:

**1. Mối quan hệ tuyến tính (Linearity)**

Mô hình giả định rằng mối quan hệ giữa biến phụ thuộc (y) và các biến độc lập (x) là tuyến tính.

Có nghĩa là sự thay đổi trong y có thể được biểu diễn như là một tổ hợp tuyến tính của x₁, x₂,...,xₙ.

Có thể kiểm tra giả định này bằng biểu đồ scatter hoặc residual plots (biểu đồ phần dư).

Nếu quan hệ là phi tuyến, bạn nên chuyển sang Polynomial Regression hoặc mô hình phi tuyến khác như Decision Tree, SVM, v.v.

**2. Độc lập của phần dư (Independence of Errors)**

Các phần dư (residuals) nên độc lập với nhau.

Vi phạm giả định này thường xảy ra với dữ liệu chuỗi thời gian (time series), nơi mà một quan sát bị ảnh hưởng bởi những quan sát trước đó.

Có thể kiểm tra bằng kiểm định Durbin-Watson.

Nếu vi phạm, nên dùng các mô hình hồi quy chuỗi thời gian như ARIMA, hoặc thêm biến trễ (.

**3. Phương sai không đổi (Homoscedasticity)**

Sai số của mô hình nên có phương sai không đổi trên tất cả các mức giá trị của biến độc lập.

Nếu phần dư có phương sai thay đổi (tăng/giảm theo x), ta gọi là heteroscedasticity – điều này làm giảm hiệu quả của ước lượng OLS.

Có thể kiểm tra bằng biểu đồ phần dư , hoặc kiểm định Breusch–Pagan, White test.

Cách xử lý: biến đổi biến đầu ra (log, sqrt), hoặc dùng kỹ thuật hồi quy có trọng số (WLS).

**4. Phân phối chuẩn của phần dư (Normality of Errors)**

Phần dư ( nên tuân theo phân phối chuẩn, đặc biệt quan trọng nếu bạn muốn áp dụng kiểm định thống kê (t-test, F-test) hoặc xây dựng khoảng tin cậy.

Kiểm tra bằng biểu đồ histogram của residuals, Q-Q plot hoặc kiểm định Shapiro-Wilk, Kolmogorov–Smirnov.

Vi phạm giả định này không quá nghiêm trọng nếu mục tiêu chỉ là dự đoán, nhưng sẽ ảnh hưởng đến việc suy luận thống kê.

**5. Không có đa cộng tuyến (No Multicollinearity)**

Các biến độc lập không nên có quan hệ tuyến tính quá cao với nhau.

Nếu tồn tại đa cộng tuyến , hệ số ước lượng sẽ không ổn định, có thể thay đổi lớn chỉ với thay đổi nhỏ trong dữ liệu.

Kiểm tra bằng hệ số phóng đại phương sai . Nếu VIF > 5 hoặc 10, nên xem xét lại.

Cách xử lý: loại bỏ một số biến tương quan mạnh, gộp biến, hoặc sử dụng Ridge/Lasso Regression để xử lý qua regularization.

### Các chỉ số đánh giá mô hình

• MSE (Mean Squared Error): trung bình bình phương sai số.

• RMSE (Root MSE): căn bậc hai của MSE.

• MAE (Mean Absolute Error): trung bình trị tuyệt đối sai số.

• R² (R-squared): tỉ lệ phương sai của y được mô hình giải thích.

• Adjusted R²: điều chỉnh R² khi thêm biến vào mô hình.

### Regularization (Chống overfitting)

• Ridge Regression (L2):

L = MSE + λ \* Σ(wⱼ²)

• Lasso Regression (L1):

L = MSE + λ \* Σ|wⱼ|

• Elastic Net:

L = MSE + λ₁ \* Σ|wⱼ| + λ₂ \* Σ(wⱼ²)

Mục tiêu của regularization là giảm độ phức tạp của mô hình và tránh overfitting.

### Ứng dụng thực tế của Linear Regression

Linear Regression có hàng loạt ứng dụng trong nhiều lĩnh vực khác nhau nhờ tính đơn giản và khả năng giải thích tốt:

• Dự đoán giá cả:

- Dự đoán giá nhà dựa trên diện tích, vị trí, số phòng...

- Dự đoán giá xe ô tô, thiết bị điện tử theo thông số kỹ thuật.

• Phân tích dữ liệu tài chính:

- Dự đoán lợi nhuận doanh nghiệp theo chi phí, doanh số, đầu tư...

- Ước lượng xu hướng thị trường chứng khoán, ngoại tệ, vàng.

• Kinh tế – xã hội:

- Dự báo tỷ lệ thất nghiệp, lạm phát, thu nhập cá nhân.

- Ước tính tác động của chính sách, chi tiêu công, mức sống.

• Y học và khoa học sức khỏe:

- Dự đoán nguy cơ mắc bệnh dựa trên tuổi, BMI, chỉ số sinh hóa.

- Phân tích hiệu quả điều trị, mối quan hệ giữa liều thuốc và kết quả.

• Giáo dục và tâm lý học:

- Ước lượng điểm số học sinh dựa trên thời gian học, điều kiện kinh tế.

- Phân tích mối quan hệ giữa mức độ stress và hiệu suất làm việc.

Tóm lại, Linear Regression là công cụ mạnh để phân tích, dự đoán và giải thích mối quan hệ trong dữ liệu thực tế.

## Hàm mất mát trong bài toán hồi quy tuyến tính

Loss Function là một phương pháp toán học để đo lường mức độ tốt hay xấu của dự đoán từ một mô hình so với kết quả thực tế. Nó cung cấp một con số duy nhất cho biết độ lệch của dự đoán. Con số càng nhỏ thì mô hình càng tốt. Loss Function được sử dụng để huấn luyện mô hình.

### Tầm quan trọng của Loss Function

Loss function là một phần không thể thiếu trong quá trình huấn luyện các mô hình học máy . Chúng cung cấp thước đo mức độ phù hợp giữa dự đoán của mô hình với dữ liệu thực tế. Bằng cách giảm thiểu tổn thất này , các mô hình học cách đưa ra dự đoán chính xác hơn. Việc lựa chọn hàm mất mát có thể ảnh hưởng đáng kể đến hiệu suất của mô hình, do đó việc lựa chọn hàm phù hợp dựa trên nhiệm vụ cụ thể là rất quan trọng.

### Mean Squared Error (MSE)

Đây là Trung bình bình phương của phần dư cho tất cả các điểm dữ liệu trong tập dữ liệu. Phần dư là sự khác biệt giữa dự đoán thực tế và dự đoán của mô hình.

Trong học máy , việc bình phương các phần dư là rất quan trọng để xử lý hiệu quả cả lỗi dương và lỗi âm. Vì các lỗi chuẩn có thể dương hoặc âm, việc cộng chúng lại có thể dẫn đến sai số ròng bằng 0, khiến mô hình lầm tưởng rằng nó đang hoạt động tốt, ngay cả khi thực tế không phải vậy . Để tránh điều này, chúng ta bình phương các phần dư, chuyển đổi tất cả các giá trị thành dương, giúp thể hiện đúng hiệu suất của mô hình.

Bình phương cũng có lợi ích bổ sung là gán nhiều trọng số hơn cho các lỗi lớn hơn, nghĩa là khi cost function cách xa giá trị tối thiểu của nó, mô hình sẽ bị phạt nặng hơn đối với các lỗi lớn hơn, giúp mô hình hội tụ đến giá trị tối thiểu nhanh hơn.

The Mean Squared Error (MSE) là một Loss Function phổ biến trong học máy, trong đó giá trị trung bình của các phần dư bình phương được lấy thay vì chỉ tổng. Điều này đảm bảo rằng Loss Function không phụ thuộc vào số điểm dữ liệu trong tập huấn luyện, giúp phép đo này đáng tin cậy hơn trên các tập dữ liệu có kích thước khác nhau. Tuy nhiên, MSE nhạy cảm với các giá trị ngoại lai, vì các sai số lớn có tác động không cân xứng đến kết quả cuối cùng.

Quá trình bình phương này rất cần thiết cho hầu hết các regression loss functions, đảm bảo rằng các mô hình có thể giảm thiểu lỗi và cải thiện hiệu suất. Công thức là:

* yi: là giá trị thực tế cho điểm dữ liệu thứ i.
* n: là tổng số điểm dữ liệu
* i: là giá trị dự đoán cho điểm dữ liệu thứ i.

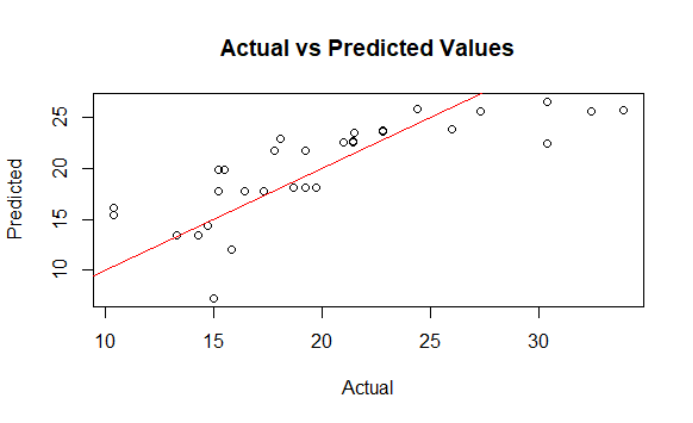
### Root-Mean-Square Error(RMSE)

Root Mean Squared Error (RMSE) là căn bậc hai của giá trị trung bình bình phương của các sai số. Đây là một thước đo sai số hữu ích cho các dự đoán số, chủ yếu để so sánh các sai số dự đoán của các mô hình hoặc cấu hình khác nhau cho cùng một biến, vì nó phụ thuộc vào quy mô. RMSE đo lường mức độ phù hợp của đường hồi quy với dữ liệu.

* **Phụ thuộc thang đo** : RMSE có cùng đơn vị với biến mục tiêu. RMSE càng thấp thì hiệu suất mô hình càng tốt, nhưng giá trị này phải được so sánh với thang đo của biến mục tiêu để có ý nghĩa.
* **Nhạy cảm với các giá trị ngoại lệ** : Vì RMSE bình phương các số hạng lỗi nên các lỗi lớn hơn có tác động lớn không cân xứng, khiến RMSE nhạy cảm với các giá trị ngoại lệ.
* **So sánh các mô hình** : RMSE có thể được sử dụng để so sánh các mô hình. Mô hình có giá trị RMSE thấp hơn thường được coi là dự đoán biến mục tiêu tốt hơn.

### Hình dung RMSE

Việc trực quan hóa hiệu suất của mô hình có thể giúp hiểu được mô hình đang hoạt động kém hiệu quả ở đâu. Biểu đồ phân tán so sánh giá trị thực tế và giá trị dự đoán có thể cung cấp thông tin chi tiết về mức độ phù hợp của mô hình với dữ liệu.



**Hình 2.1: RMSE**

Các điểm càng gần đường màu đỏ (thực tế = dự đoán) thì dự đoán của mô hình càng tốt.

Công thức RMSE là:

### R-squared

R-squared là một thước đo thống kê biểu thị độ phù hợp của mô hình hồi quy. Giá trị R-squared nằm trong khoảng từ 0 đến 1. Trong đó, R-squared bằng 1 khi mô hình hoàn toàn khớp với dữ liệu và không có sự khác biệt giữa giá trị dự đoán và giá trị thực tế. Tuy nhiên, R-squared bằng 0 khi mô hình không dự đoán bất kỳ biến thiên nào trong mô hình và không tìm thấy bất kỳ mối quan hệ nào giữa các biến phụ thuộc và biến độc lập.

### R-squared được tính như thế nào?

R bình phương hay còn gọi là hệ số xác định đo lường sự thay đổi của biến phụ thuộc Y được giải thích bởi các biến độc lập X i trong mô hình hồi quy.

* Đầu tiên, tính giá trị trung bình của biến mục tiêu/biến phụ thuộc y và chúng ta ký hiệu nó là y̅
* Tính tổng bình phương bằng cách trừ mỗi quan sát y i khỏi y̅, sau đó bình phương nó và cộng các hiệu số bình phương này trên tất cả các giá trị. Nó được ký hiệu là
* Chúng tôi ước tính tham số mô hình bằng cách sử dụng mô hình hồi quy phù hợp như Hồi quy tuyến tính.
* Chúng tôi tính Tổng bình phương do hồi quy, được ký hiệu là SSR. Giá trị này được tính bằng cách trừ mỗi giá trị dự đoán của y, được ký hiệu là y\_pred i , cho y i , bình phương các chênh lệch này, rồi cộng tất cả n số hạng. Công thức là :
* Chúng tôi tính tổng bình phương (SS res ) . Nó giải thích sự biến thiên chưa được tính đến của biến phụ thuộc y sau khi dự đoán các giá trị này từ một biến độc lập trong mô hình. Công thức là:
* Sau đó chúng ta có thể sử dụng một trong hai

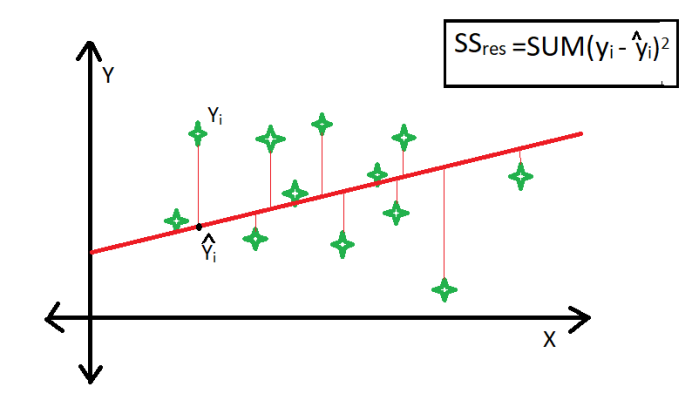
### R-squared độ phù hợp

R-square là phép so sánh giữa tổng bình phương còn lại (SS res ) với tổng bình phương tổng thể (SS tot ) . Tổng bình phương tổng thể được tính bằng cách cộng các bình phương của khoảng cách vuông góc giữa các điểm dữ liệu và đường trung bình.



**Hình 2.2: R-Squared**

Tổng bình phương còn lại được tính bằng tổng bình phương của khoảng cách vuông góc giữa các điểm dữ liệu và đường thẳng phù hợp nhất.



**Hình 2.3: R-Squared cho res**

### Hạn chế của việc sử dụng phương pháp R-squared

* Giá trị của r-square luôn tăng hoặc giữ nguyên khi các biến mới được thêm vào mô hình, mà không phát hiện ra ý nghĩa của biến mới này (tức là giá trị của r-square không bao giờ giảm khi thêm các thuộc tính mới vào mô hình). Do đó, các thuộc tính không có ý nghĩa cũng có thể được thêm vào mô hình với sự gia tăng giá trị r-square.
* Điều này là do SS tot luôn không đổi và mô hình hồi quy cố gắng giảm giá trị của SS res bằng cách tìm một số mối tương quan với thuộc tính mới này, do đó giá trị tổng thể của r-square tăng lên, điều này có thể dẫn đến mô hình hồi quy kém.

## Tập dữ liệu

### Nguồn dữ liệu

Bài toán áp dụng thuật toán Hgre ồi quy tuyến tính (Linear Re ssion) để dự báo giá trị dựa trên chuỗi dữ liệu đã cho:

* Thermostat Sales: Dự báo doanh số bán máy điều hòa theo tuần
* Air Passengers: Dự báo số lượng hàng khách đi máy bay theo tháng

Được trích xuất trực tiếp từ đề bài với 2 tập dữ liệu là:

* Thermostat Sales: 52 điểm
* Air Passengers: 144 điểm

Được lưu dưới dạng số nguyên, không có giá trị bị thiếu, các giá trị được tiền xử lí để liên tục và không phát sinh lỗi trong quá trình thực hiện

### Mô tả dữ liệu

**Bảng 2.1: Mô tả dữ liệu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên dữ liệu** | **Số mẫu dữ liệu** | **Đơn vị thời gian** | **Kiểu dữ liệu** | **Đặc trưng** | **Mô tả** |
| Thermostat Sales | 52 | Tuần | Số nguyên (int) | -Dữ liệu chuỗi thời gian  -Một chiều  -Khoảng cách thời gian theo tuần  -Không có giá trị phụ thuộc  -Không có giả định  -Có biến động theo thời gian | Số lượng sản phẩm Thermostat bán ra mỗi tuần |
| Air Passengers | 144 | Tháng | Số nguyên (int) | - Dữ liệu chuỗi thời gian  -Một chiều  -Khoảng cách thời gian theo tuần  -Không có giá trị phụ thuộc  -Không có giả định  -Có biến động theo thời gian | Số lượng hàng khách đi máy bay mỗi tháng |

### Tiền xử lý dữ liệu

Dữ liệu được duỗi thẳng thành một hàng dọc để tiện cho quá trình đọc và tiến hành huấn luyện

Có hai giá trị là index được đánh giá trị tự động liên tục từ 0 đến N (điểm giá trị cuối cùng trong tập dữ liệu)

### Trực quan dữ liệu

**Dữ liệu Thermost Sales:**

**Phân bố dữ liệu:**

A graph of blue dots

AI-generated content may be incorrect.

**Hình 2.4: Biểu đồ phân bố cho Thermostat Sales**

Đánh giá:

* Các điểm dữ liệu phân bố rải rác
* Ít các điểm ngoại biên
* Đánh giá dữ liệu tương đối sạch và không cần tiền xử lý nhiều

**Đồ thị đánh giá biểu hiện theo cột:**

A bar chart with blue lines

AI-generated content may be incorrect.

**Hình 2.5: Biều đồ cột cho Thermostat Sales**

Đánh giá:

* Phân bố đồng đều: Biểu đồ cho thấy một sự phân bố khá đồng đều của các giá trị doanh số nhiệt độ trong suốt các tuần, với không có sự biến động quá lớn ở hầu hết các chỉ số. Điều này có thể chỉ ra rằng doanh số bán hàng không có sự dao động lớn theo thời gian.
* Điểm cực trị rõ ràng: Một vài cột có giá trị rất cao so với các cột khác, đặc biệt là ở các chỉ số tuần cao (gần cuối biểu đồ). Điều này có thể chỉ ra các yếu tố ngoại lai hoặc một sự kiện đặc biệt làm tăng doanh số vào các tuần đó.
* Tạo sự dễ dàng phân tích xu hướng: Với đồ thị cột này, người xem có thể dễ dàng nhận diện các tuần có doanh số cao nhất và thấp nhất, giúp dễ dàng đưa ra các phân tích về xu hướng hoặc thói quen tiêu dùng theo từng tuần.

**Biểu đồ hộp:**

A graph with lines and a rectangle

AI-generated content may be incorrect.

**Hình 2.6: Biểu đồ hộp cho Thermostat sales**

Đánh giá:

* Dải giữa (IQR): Phần lớn giá trị doanh số nằm trong khoảng từ khoảng 175 đến 275.
* Giá trị trung bình: Đường giữa hộp thể hiện giá trị trung bình của doanh số là khoảng 225.
* Khoảng giá trị ngoài: Biểu đồ không có dấu hiệu của các giá trị ngoại lai (outliers), vì vậy dữ liệu có vẻ ổn định.

**Dữ liệu Air Passengers:**

**Phân bố dữ liệu:**

A graph of blue dots

AI-generated content may be incorrect.

**Hình 2.7: Phân bố dữ liệu cho Air Passengers**

Đánh giá:

* Các điểm dữ liệu phân bố theo nhóm, gồm nhiều nhóm gần nhau
* Có các điểm ngoại biên
* Tổng quan dữ liệu ổn và có thể sử dụng mà không cần tiền xử lý nhiều

**Theo biểu đồ cột:**

A graph of blue bars

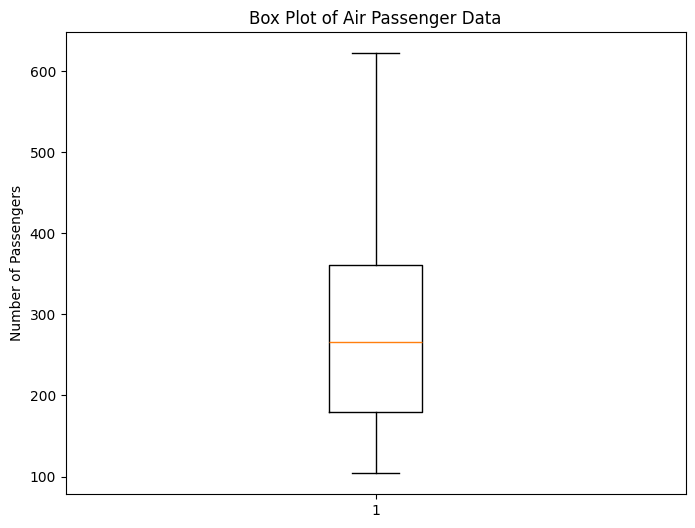
AI-generated content may be incorrect.

**Hình 2.8: Phân bố dạng cột dữ liệu cho Air Passengers**

Đánh giá:

* Biểu đồ cho thấy một xu hướng tăng trong số lượng hành khách hàng không qua các tháng, với một đỉnh rõ ràng vào tháng Mười.
* Dữ liệu được trình bày rõ ràng, với các cột phân bổ hợp lý giúp dễ dàng theo dõi sự tăng trưởng mỗi tháng.
* Trục x đại diện cho các tháng, và trục y đại diện cho số lượng hành khách, giúp so sánh trực quan lượng hành khách trong suốt năm.

**Theo biểu đồ hộp:**



**Hình 2.9: Biểu đồ hộp cho Air Passengers**

Đánh giá:

* Giá trị trung vị (median) của số hành khách rơi vào khoảng 300.
* Phạm vi giữa (IQR) cho thấy phần lớn dữ liệu tập trung từ khoảng 200 đến 400 hành khách.
* Biểu đồ chỉ ra sự hiện diện của các điểm ngoại lai ở cả hai đầu dưới và trên, cho thấy có những giá trị cực đoan trong tập dữ liệu.

# Phương pháp thực hiện

## Quá trình thực hiện

Xây dựng giao diện trực quan dễ nhìn, dễ thao tác và phù hợp mới mọi tệp khách hàng, đem lại cho khách hàng những trải nghiệm tốt nhất

A diagram of a diagram

AI-generated content may be incorrect.

**Hình 3.1: Sơ đồ hoạt động**

Tổng quan:

* Loại thuật toán: Học có giám sát (supervised learning), thuộc nhóm hồi quy (Regression)
* Mục đích: Tìm hàm số tuyến tính gần đúng nhất để dự báo giá trị đầu ra dựa trên đầu vào

## Thực hiện với tập dữ liệu Thermostat Sales

Sau khi tiền xử lí chúng em sẽ bắt đầu huấn luyện mô hình theo 2 cách để có thể đánh giá hiệu quả và tốt nhất, so sánh và đánh giá khả năng của mô hình.

### Thực hiện với hồi quy tuyến tính thủ công

Các tham số được sử dụng là:

* Tham số học(learning rate:lr): 0.0001
* Số lượt huấn luyện mô hình(epoch):100000

Các tham số được chọn là phù hợp với quá trình học và tập dữ liệu

Hàm mất mát trên mô hình này là:

* Mean Squared Error (MSE): 755.106
* Root Mean Squared Error (RMSE): 27.479
* R Square: 0.721

Đánh giá :

* MSE (Mean Squared Error) = 755.106: MSE không quá cao, cho thấy sai số trung bình của mô hình ở mức chấp nhận được.
* RMSE (Root Mean Squared Error) = 27.479: RMSE cho thấy độ lệch chuẩn của sai số xung quanh giá trị dự đoán. Giá trị này cho thấy sai số khá hợp lý, tuy nhiên có thể cải thiện nếu giảm bớt sai số lớn.
* R² = 0.721: Giá trị R² khá cao, cho thấy mô hình giải thích được khoảng 72.1% biến thiên của dữ liệu, chứng tỏ mô hình hoạt động khá tốt trong việc dự đoán.

Hồi quy tuyến tính với tập huấn luyện:

A graph with blue dots and a red line

AI-generated content may be incorrect.

**Hình 3.2: Thermostat Sales thủ công hồi quy tuyến tính trên tập huấn luyện**

Đánh giá:

* Mô hình học khá tốt và gần như hợp vào với dữ liệu
* Mối quan hệ tuyến tính rõ ràng: Dữ liệu có mối quan hệ tuyến tính tăng dần giữa Tuần và Doanh số Thermostat, với đường hồi quy phù hợp với xu hướng dữ liệu.
* Dự đoán chính xác: Đường hồi quy cho thấy khả năng dự đoán tốt với dữ liệu huấn luyện, biểu thị sự tương thích giữa mô hình và dữ liệu.
* Có thể cần thêm dữ liệu: Mặc dù có mối quan hệ tuyến tính, nhưng vẫn có một số điểm dữ liệu ngoài đường hồi quy, chỉ ra rằng mô hình có thể chưa hoàn toàn chính xác đối với tất cả các trường hợp.

Hồi quy tuyến tính với tập huấn luyện:

A graph with green and orange dots

AI-generated content may be incorrect.

**Hình 3.3: Thermostat Sales thủ công hồi quy tuyến tính trên tập kiểm tra**

Đánh giá:

* Mô hình học tốt
* Xu hướng: Có thể quan sát thấy một xu hướng tuyến tính đi lên rõ rệt, chỉ ra mối tương quan tích cực giữa các tuần (Tuần) và doanh số (Doanh số Thermostat).
* Sự phù hợp: Đường hồi quy màu cam dường như bắt được xu hướng chung của các điểm dữ liệu, với một sự ước lượng hơi thấp cho các tuần đầu và khớp tốt hơn ở các tuần sau.
* Phân tán: Các chấm màu xanh lá cây đại diện cho các điểm dữ liệu kiểm tra, có một vài điểm ngoại lai xuất hiện ở phạm vi thấp của các tuần.

### Thực hiện với hồi quy tuyến tính dùng thư viện

Các tham số được sử dụng của mô hình được thư viện chọn nhằm mục đích tối ưu hóa và cho ra kết quả tốt nhất. Nên đây là các tham số tự động và được tự điều chỉnh theo thư viện

Hàm mất mát trên mô hình này là:

* Mean Squared Error (MSE): 783.876
* Root Mean Squared Error (RMSE): 27.998
* R Square: 0.711

Đánh giá :

* MSE (Mean Squared Error) = 783.876: MSE cho thấy sai số trung bình giữa giá trị dự đoán và giá trị thực tế. Giá trị này cho thấy mô hình có một mức sai số khá hợp lý nhưng vẫn có thể cải thiện thêm.
* RMSE (Root Mean Squared Error) = 27.998: RMSE cho thấy độ lệch chuẩn của sai số dự đoán so với thực tế. Giá trị này cho thấy sai số phân bố khá đều và không quá lớn, tuy nhiên có thể tối ưu hóa thêm.
* R² = 0.711: Giá trị R² cho thấy mô hình giải thích được khoảng 71.1% biến thiên của dữ liệu, chứng tỏ mô hình khá tốt trong việc dự đoán nhưng vẫn còn một phần dữ liệu chưa được giải thích.

Hồi quy tuyến tính với tập huấn luyện:

A graph with blue dots and a red line

AI-generated content may be incorrect.

**Hình 3.4: Thermostat Sales thư viện hồi quy tuyến tính trên tập huấn luyện**

Đánh giá:

* Mối quan hệ tuyến tính rõ ràng: Các điểm dữ liệu thể hiện mối quan hệ tuyến tính giữa "Tuần" và "Doanh số Thermostat", với một xu hướng tăng dần.
* Dự đoán của mô hình: Đường hồi quy (đỏ) mô tả tốt xu hướng của dữ liệu huấn luyện, có thể dự đoán chính xác doanh số thermostat theo thời gian.
* Độ phân tán của dữ liệu: Mặc dù mô hình tuyến tính khá phù hợp, vẫn có sự phân tán giữa các điểm dữ liệu, điều này cho thấy có sự ảnh hưởng của yếu tố ngoài mô hình.

Hồi quy tuyến tính với tập kiểm tra:

A graph with a red line

AI-generated content may be incorrect.

**Hình 3.5: Thermostat Sales thư viện hồi quy tuyến tính trên tập kiểm tra**

Đánh giá:

* Mối quan hệ giữa các biến: Đồ thị cho thấy mối quan hệ tuyến tính giữa "Tuần" (x) và "Doanh số Thermostat" (y). Đường hồi quy cho thấy sự tăng trưởng của doanh số theo thời gian.
* Độ phù hợp mô hình: Đường hồi quy tuyến tính khớp khá tốt với các điểm dữ liệu, cho thấy mô hình có thể giải thích một phần sự biến đổi trong dữ liệu.
* Dự báo: Dự báo doanh số theo tuần có thể tin cậy trong khoảng thời gian đã kiểm tra, tuy nhiên cần kiểm tra thêm với các tập dữ liệu ngoài phạm vi để đánh giá tính chính xác dài hạn.

### Đánh giá mô hình với tập dữ liệu trên hai hướng xây dựng mô hình:

**So sánh hai hàm mất mát của hướng:**

* MSE (Mean Squared Error):
  + Manual: 755.1066
  + Library: 783.8765
  + Đánh giá: MSE giữa kết quả thủ công và thư viện có sự chênh lệch nhỏ. MSE thư viện cao hơn một chút, điều này có thể do sự khác biệt trong cách tính toán hoặc làm tròn.
* RMSE (Root Mean Squared Error):
  + Manual: 27.4792
  + Library: 27.9978
  + Đánh giá: Chênh lệch RMSE cũng khá nhỏ, cho thấy kết quả giữa hai phương pháp gần giống nhau. Sự khác biệt này có thể do chi tiết trong quá trình tính toán.
* R² (Coefficient of Determination):
  + Manual: 0.7217
  + Library: 0.7111
  + Đánh giá: R² gần nhau và đều trên 0.7, cho thấy mô hình giải thích tốt dữ liệu. Sự giảm nhẹ trong R² của thư viện không ảnh hưởng đáng kể đến khả năng giải thích mô hình.

Kết luận: Kết quả giữa tính toán thủ công và thư viện rất gần nhau, với các sai số nhỏ chủ yếu do cách tính toán hoặc làm tròn. Mô hình có khả năng giải thích dữ liệu tốt (R² > 0.7) và có thể áp dụng cho dự báo.

So sánh hai phương pháp trên tập huấn luyện:

A graph with blue dots and green lines

AI-generated content may be incorrect.

**Hình 3.6: Thermostat Sales so sánh hồi quy tuyến tính trên tập huấn luyện**

Đánh giá:

* Mối quan hệ giữa dữ liệu và mô hình: Biểu đồ cho thấy sự phù hợp tốt giữa các điểm dữ liệu thực tế (biểu thị bằng các chấm xanh) và đường hồi quy của cả hai mô hình (đường đỏ cho mô hình thủ công và đường xanh lá cho mô hình từ thư viện).
* Hiệu suất của mô hình thủ công và thư viện: Các đường hồi quy của hai mô hình có xu hướng khá giống nhau, với đường của mô hình thư viện có một chút dao động nhỏ so với mô hình thủ công, điều này có thể do các thuật toán tối ưu hóa và quá trình huấn luyện của từng mô hình.
* Tính chính xác**:** Mô hình hồi quy tuyến tính từ thư viện (green) và mô hình thủ công (red) cho thấy kết quả tương đối gần nhau, với độ chính xác khá cao trên tập huấn luyện. Mặc dù vậy, nếu có sai số (như MSE và RMSE), mô hình thư viện có thể có những cải thiện nhờ các tối ưu hóa kỹ thuật trong thư viện.
* Nhận xét chung: Mô hình này có vẻ hợp lý và cho thấy mối quan hệ tuyến tính rõ ràng giữa các yếu tố, tuy nhiên, có thể cải thiện thêm độ chính xác nếu tinh chỉnh thêm các tham số hoặc sử dụng các phương pháp tối ưu hóa phức tạp hơn.
* Tổng thể, mô hình đã đạt được hiệu quả khá tốt, nhưng vẫn có không gian để tối ưu hóa thêm nếu muốn cải thiện độ chính xác.

Đánh giá hai phương pháp với tập kiểm tra:

A graph with a red line and green dots

AI-generated content may be incorrect.

**Hình 3.7: Thermostat Sales so sánh hồi quy tuyến tính trên tập kiểm tra**

Đánh giá:

* Mô hình từ thư viện (Library LR) có sự khớp tốt hơn với dữ liệu thực tế so với mô hình tự xây dựng (Manual LR), thể hiện hiệu quả cao hơn trong việc dự đoán doanh thu.
* Tuy nhiên, vẫn có một vài điểm dữ liệu ngoài dự đoán của cả hai mô hình, cho thấy cần cải thiện thêm trong việc xử lý các ngoại lệ hoặc điểm dữ liệu bất thường.

**Tổng kết:**

* Giữa thủ công và bằng thư viện không có sự chệnh lệnh rõ ràng, sai số rất nhỏ
* Xây dựng mô hình bằng thư viện cho ra độ chính xác tốt hơn, tối ưu hơn cho cả người sử dụng và ứng dụng thực tế, vì người dùng không cần phải tự điều chỉnh tham số để chọn ra tham số tốt nhất cho mô hình
* Xây dựng mô hình thủ công giúp người làm hiểu rõ hơn về quá trình và cách hoạt động của thuật toán, nhằm mục đích phát triển và cải thiện kiến thức
* Mô hình xây dựng bằng thư viện tốt và tối ưu hơn mô hình xây dựng thủ công

## Thực hiện với tập dữ liệu Air Passengers

Sử dụng hai phương pháp huấn luyện là thủ công và dùng thư viện với tập dữ liệu Air Passengers nhằm mục đích đánh giá và so sánh khả năng học của mô hình trên hai tập dữ liệu, từ đó đưa ra đánh giá cho cả dữ liệu và mô hình

### Thực hiện với hồi quy tuyến tính thủ công

Các tham số được sử dụng là:

* Tham số học(learning rate:lr): 0.0001
* Số lượt huấn luyện mô hình(epoch):100000

Các tham số được chọn là phù hợp với quá trình học và tập dữ liệu

Hàm mất mát trên mô hình này là:

* Mean Squared Error (MSE): 12368.7163
* Root Mean Squared Error (RMSE): 111.2147
* R Square: -0.1135

Đánh giá :

* **MSE (Mean Squared Error) = 12368.72**: MSE cao cho thấy sai số giữa giá trị dự đoán và giá trị thực tế là khá lớn, điều này cho thấy mô hình cần được cải thiện để giảm sai số và nâng cao độ chính xác.
* **RMSE (Root Mean Squared Error) = 111.21**: RMSE cao cho thấy độ lệch chuẩn của sai số xung quanh giá trị dự đoán là khá lớn. Điều này cho thấy mô hình có sự dao động lớn trong việc dự đoán, và có thể cần tối ưu hóa thêm để giảm thiểu sai số.
* **R² = -0.1135**: Giá trị R² âm cho thấy mô hình không thể giải thích được sự biến động trong dữ liệu, thậm chí còn tệ hơn việc dự đoán ngẫu nhiên. Điều này cho thấy mô hình cần cải tiến mạnh mẽ và không phù hợp với dữ liệu hiện tại.

Hồi quy tuyến tính với tập dữ liệu huấn luyện:

A graph with blue dots and red line

AI-generated content may be incorrect.

**Hình 3.8: Air Passengers thủ công hồi quy tuyến tính trên tập huấn luyện**

Đánh giá:

* Xu Hướng: Đường hồi quy có độ dốc nhẹ hướng lên, cho thấy có sự tương quan tích cực giữa các biến (Tháng và Số lượng hành khách). Tuy nhiên, các điểm dữ liệu khá phân tán xung quanh đường hồi quy, cho thấy sự biến động lớn giữa các điểm dữ liệu.
* Giải Thích: Mô hình hồi quy thủ công chỉ có thể nắm bắt được xu hướng chung của dữ liệu nhưng không thể phản ánh đầy đủ mối quan hệ giữa các biến. Mặc dù đường hồi quy có sự xu hướng tích cực, nhưng sự phân tán của các điểm dữ liệu cho thấy mô hình chưa đủ chính xác, có thể do việc lựa chọn các tham số chưa tối ưu, hoặc mô hình không thể giải thích hết sự biến động của dữ liệu.

Hồi quy tuyến tính với tập dữ liệu kiểm tra:

A graph with green dots and a line

AI-generated content may be incorrect.

**Hình 3.9: Air Passengers thủ công hồi quy tuyến tính trên tập kiểm tra**

Đánh giá:

* + **Xu Hướng**: Đối với dữ liệu kiểm tra, đường hồi quy cũng có độ dốc hướng lên, nhưng các điểm dữ liệu phân tán rất lớn quanh đường hồi quy. Điều này cho thấy mô hình không thể tổng quát hóa tốt cho dữ liệu chưa thấy.
  + **Giải Thích**: Việc phân tán dữ liệu quá rộng cho thấy mô hình không áp dụng tốt được cho dữ liệu kiểm tra, điều này chỉ ra rằng hồi quy thủ công dễ bị overfitting trên tập huấn luyện. Mặc dù mô hình tìm được một xu hướng chung, nhưng nó không thể dự đoán chính xác được các giá trị của tập kiểm tra. Điều này cho thấy mô hình này có khả năng tổng quát thấp.

### Thực hiện với hồi quy tuyến tính dùng thư viện

Như đã nói ở trên khi sử dụng mô hình từ thư viện thì các tham số của mô hình đã được tối ưu để cho quá trình huấn luyện và kết quả được cho ra là tốt nhất và phù hợp nhất với dữ liệu. Nên ở đây sẽ không có phần các tham số được cài đặt.

Hàm mất mát trên mô hình này là:

* Mean Squared Error (MSE): 12311.8349
* Root Mean Squared Error (RMSE): 110.9587
* R Square: -0.1084

Đánh giá :

* + MSE (Mean Squared Error): MSE của mô hình sử dụng thư viện không quá thấp, cho thấy sai số giữa các giá trị dự đoán và thực tế vẫn còn khá lớn. Tuy nhiên, so với mô hình thủ công, giá trị này có thể được coi là hợp lý, nhưng vẫn cần cải thiện thêm để giảm sai số.
  + RMSE (Root Mean Squared Error): RMSE cho thấy độ lệch chuẩn của sai số xung quanh giá trị dự đoán vẫn còn khá cao. Điều này cho thấy mô hình chưa hoàn toàn chính xác trong việc dự đoán, dù có sự cải thiện nhẹ so với mô hình thủ công.
  + R² (R Square): Giá trị R² âm cho thấy mô hình không thể giải thích được sự biến động trong dữ liệu. Mô hình này cũng tệ hơn việc dự đoán ngẫu nhiên, điều này cho thấy mô hình không phù hợp với dữ liệu và cần được tối ưu hóa thêm.

Hồi quy tuyến tính với tập dữ liệu huấn luyện:

A graph with blue dots and a red line

AI-generated content may be incorrect.

**Hình 3.10: Air Passengers thư viện hồi quy tuyến tính trên tập huấn luyện**

Đánh giá:

* + Xu Hướng: Đối với mô hình sử dụng thư viện, đường hồi quy có xu hướng mượt mà hơn với độ dốc tăng lên, và có sự khớp tốt hơn với dữ liệu kiểm tra so với mô hình thủ công.
  + Giải Thích: Đường hồi quy mượt mà cho thấy mô hình đã được tối ưu hóa tốt hơn so với phương pháp thủ công. Dữ liệu kiểm tra ít phân tán xung quanh đường hồi quy hơn, chứng tỏ mô hình có khả năng tổng quát tốt hơn khi đối mặt với dữ liệu chưa thấy. Điều này có thể do việc sử dụng các thư viện tối ưu giúp mô hình xử lý dữ liệu tốt hơn, và các tham số được tối ưu hóa tự động, giúp giảm thiểu hiện tượng

Hồi quy tuyến tính với tập dữ liệu kiểm tra:

A graph with purple dots and a red line

AI-generated content may be incorrect.

**Hình 3.11: Air Passengers thư viện hồi quy tuyến tính trên tập kiểm tra**

Đánh giá :

* + Xu Hướng: Đối với tập huấn luyện, đường hồi quy cũng có sự khớp tốt, thể hiện rằng mô hình đã học được xu hướng và mối quan hệ trong dữ liệu huấn luyện một cách chính xác hơn.
  + Giải Thích: So với hồi quy thủ công, mô hình sử dụng thư viện có đường hồi quy mượt mà hơn và các điểm dữ liệu ít phân tán xung quanh đường dự đoán. Điều này cho thấy mô hình đã học tốt hơn từ dữ liệu huấn luyện và có khả năng dự đoán chính xác hơn, thể hiện hiệu quả của việc sử dụng thư viện trong việc tối ưu hóa mô hình. Các thư viện này có thể tự động điều chỉnh các tham số và sử dụng các thuật toán học máy mạnh mẽ hơn, giúp mô hình có thể tổng quát tốt hơn.

### Đánh giá mô hình với tập dữ liệu trên hai hướng xây dựng mô hình:

**So sánh hai hàm mất mát của hướng:**

* MSE (Mean Squared Error):
  + Thủ công: 12368.72
  + Sử dụng Thư Viện: 12311.83
  + Đánh giá: MSE của mô hình sử dụng thư viện có giá trị thấp hơn một chút so với mô hình thủ công, cho thấy rằng mô hình sử dụng thư viện có khả năng dự đoán chính xác hơn, mặc dù sự khác biệt này không quá lớn.
* RMSE (Root Mean Squared Error):
  + Thủ công: 111.21
  + Sử dụng Thư Viện: 110.96
  + Đánh giá: RMSE của mô hình sử dụng thư viện thấp hơn một chút so với mô hình thủ công, cho thấy mô hình sử dụng thư viện có độ lệch chuẩn của sai số tốt hơn, dù vẫn còn khá cao. Tuy nhiên, sự cải thiện không đáng kể.
* R² (R-Square):
  + Thủ công: -0.1135
  + Sử dụng Thư Viện: -0.1084
  + Đánh giá: Giá trị R² của mô hình sử dụng thư viện cải thiện một chút so với mô hình thủ công, nhưng vẫn giữ giá trị âm, cho thấy cả hai mô hình đều không thể giải thích được sự biến động trong dữ liệu.

**Kết Luận:**

Mô hình sử dụng thư viện có hiệu suất tốt hơn một chút so với mô hình thủ công, thể hiện qua các chỉ số MSE, RMSE và R² cải thiện nhẹ. Tuy nhiên, cả hai mô hình đều không đạt hiệu suất tốt, với MSE và RMSE vẫn còn khá cao, và giá trị R² âm, cho thấy cả hai mô hình đều chưa phù hợp và cần được cải thiện thêm.

So sánh kết quả của hai phương pháp trên tập huấn luyện:

A graph with blue dots and green lines

AI-generated content may be incorrect.

**Hình 3.12: Air Passengers so sánh hồi quy tuyến tính trên tập huấn luyện**

Đánh giá:

* + Dữ liệu Huấn Luyện (Train data): Các điểm dữ liệu (màu xanh) thể hiện sự phân bố không đồng đều nhưng vẫn có xu hướng tăng theo thời gian (Tuần).
  + Mô hình Hồi Quy Tuyến Tính (Manual LR và Library LR): Cả hai đường hồi quy (đoạn gạch và đường liên tục) đều cho thấy một xu hướng tăng nhẹ theo thời gian, với đường của mô hình thư viện (Library LR) có phần thích ứng tốt hơn so với mô hình thủ công (Manual LR), mặc dù sự khác biệt là nhỏ.
  + Đánh giá: Mô hình hồi quy có thể chưa phản ánh hết sự phức tạp của dữ liệu huấn luyện, vì vẫn còn nhiều điểm dữ liệu phân tán khá xa đường hồi quy.

So sánh kết quả của hai phương pháp trên tập kiểm tra:

A graph with green and red dots

AI-generated content may be incorrect.

**Hình 3.13: Air Passengers so sánh hồi quy tuyến tính trên tập kiểm tra**

Đánh giá:

* + Dữ liệu Kiểm Tra (Test data): Các điểm dữ liệu (màu xanh) phân tán khá rộng, không có xu hướng rõ rệt như trong tập huấn luyện. Dữ liệu kiểm tra ít đồng nhất hơn.
  + Mô hình Hồi Quy Tuyến Tính (Manual LR và Library LR): Đối với mô hình thủ công (Manual LR) và mô hình thư viện (Library LR), cả hai đều tạo ra một đường xu hướng tương tự với dữ liệu, nhưng sự phân tán của dữ liệu kiểm tra gây khó khăn trong việc dự đoán chính xác. Đặc biệt, sự phân tán của dữ liệu kiểm tra có thể khiến mô hình gặp khó khăn trong việc tối ưu.
  + Đánh giá: Mô hình hồi quy hoạt động kém hơn trong tập kiểm tra, cho thấy rằng mô hình chưa đủ mạnh để xử lý dữ liệu kiểm tra với sự phân tán lớn và không có sự tương quan rõ rệt như tập huấn luyện.

Tổng kết:

* + Cả hai phương pháp đều cho ra kết quả tương đối giống nhau và gần như là hòa vào nhau
  + Sử dụng mô hình huấn luyện thủ công mất nhiều thời gian hơn do số epoch được điều chỉnh khá cao
  + Mô hình sử dụng thư viện cho tốc độ nhanh hơn, kết quả cho ra tối ưu hơn
  + Không có quá nhiều sự khác biệt giữa thủ công và thư viện chứng tỏ rằng mô hình hoạt động tốt với bộ tham số tốt hoặc sử dụng thư viện để tránh mất thời gian thử nghiệm và kiểm tra
  + Có thể sử dụng cả hai tùy theo yêu cầu người dùng

## Đánh giá kết quả của mô hình trên hai tập dữ liệu

**Tập Dữ liệu Air Passenger:**

Dữ liệu này có sự phân tán lớn và không có mối quan hệ tuyến tính rõ ràng. Các điểm dữ liệu không theo một xu hướng rõ ràng, khiến mô hình hồi quy tuyến tính không thể bắt được mối quan hệ giữa tuần và số lượng hành khách.

Mô hình không học tốt:

* + Mối quan hệ giữa các biến không tuyến tính, mà thay đổi theo các yếu tố như mùa vụ, sự kiện, khiến mô hình không thể dự đoán chính xác.
  + MSE và RMSE cao cho thấy mô hình có sai số dự đoán lớn, cho thấy khả năng học kém.
  + R² âm chỉ ra rằng mô hình không giải thích được sự biến thiên trong dữ liệu và thậm chí không cải thiện dự đoán so với giá trị trung bình của dữ liệu.

Giới hạn của mô hình: Mô hình hồi quy tuyến tính không thể học tốt khi dữ liệu có mối quan hệ phi tuyến và bị ảnh hưởng bởi nhiều yếu tố không rõ ràng.

**Tập Dữ liệu Thermostat Sales:**

Dữ liệu này có mối quan hệ tuyến tính rõ ràng giữa tuần và doanh số thermostat, các điểm dữ liệu phân bố gần một đường thẳng.

Mô hình học tốt:

* + Mô hình hồi quy tuyến tính có thể bắt được mối quan hệ tuyến tính giữa tuần và doanh số, dẫn đến các dự đoán chính xác.
  + MSE và RMSE thấp cho thấy mô hình dự đoán chính xác, sai số chuẩn không lớn.
  + R² cao (0.72) chứng tỏ mô hình giải thích được sự biến thiên trong dữ liệu, khả năng dự đoán tốt.

**Giới hạn của mô hình**: Mô hình chỉ học tốt khi dữ liệu có mối quan hệ tuyến tính rõ ràng. Nếu có thêm các yếu tố phức tạp, mô hình có thể không hiệu quả.

**Tổng Kết:**

* + **Air Passenger**: Mô hình không học tốt vì dữ liệu không có mối quan hệ tuyến tính và bị ảnh hưởng bởi nhiều yếu tố không xác định.
  + **Thermostat Sales**: Mô hình học tốt vì dữ liệu có mối quan hệ tuyến tính rõ ràng và ít yếu tố nhiễu. Mô hình hồi quy tuyến tính chỉ hiệu quả khi dữ liệu có tính tuyến tính.

# KẾT LUẬN

## Kết quả đạt được

Đánh giá được tổng quan về khả năng học của mô hình bằng hai cách đó là thủ công và sử dụng thư viện để tiến hành huấn luyện và dưa ra kết quả.

Đánh giá được khả năng học của mô hình trên hai tập dữ liệu khác nhau, thấy được ưu và nhược điểm của mô hình đối với các kiểu dữ liệu khác nhau

Tổng kết được kiến thức về mô hình hồi quy tuyến tính và có thể áp dụng nó vào các bài toán khác trong tương lai

## Ứng dụng thực tế

* Dự báo doanh thu bán hàng dựa trên ngân sách quảng cáo và số lương sản phẩm bán ra, xu hướng thị trường
* Ước tính giá bất động sản dựa trên diện tích, vị trí, số phòng, và diện tích xung quang
* Xác định mức lương nhân viên dựa trên kinh nghiệm làm việc, trình độ học vấn và ngành nghề

## Ưu và nhược điểm

### Ưu điểm

Dễ hiểu và dễ áp dụng: Hồi quy tuyến tính là một trong những mô hình đơn giản và dễ nằm bắt. Có tính trực quan tốt, phù hợp cho người mới bắt đầu tiếp cận với trí tuệ nhân tạo

Nhanh chóng và tiết kiệm tài nguyên: Như đã đề cập ở phương pháp thực nghiệm thì việc sử dụng mô hình từ thư viện giúp cho việc mô hình chạy rất nhanh và không cần phải tìm ra bộ tham số tối ưu, đồng thời tốc độ chạy khá là nhanh và không yêu cùa quá nhiều tài nguyên

Dễ triển khai và sử dụng phổ biến: Thuật toán này được sử dụng phổ biến và cho nhiều lĩnh vực khác nhau, không quá phức tạp dễ hiểu giúp người dùng nhanh chóng sử dụng.

### Nhược điểm

Chỉ phù hợp với dữ liệu sạch và đơn giản: Hồi quy tuyến tính hoạt động tốt khi dữ liệu có xu hướng tăng hoặc giảm đều. Nếu dữ liệu có biến động phức tạp mô hình có thể không cho ra kết quả chính xác.

Dễ bị ảnh hưởng bởi các điểm ngoại biên: Những dữ liệu ngoại biên có thể làm việc học của mô hình bị sai lệch

Giới hạn trong việc nắm bắt các mỗi quan hệ: Mô ình không thể phản ảnh các tương tác nâng cao giữa các yếu tố đầu vào nếu không có sự mở rộng

## Hướng mở rộng trong tương lai

Để nâng cao hiệu quả và ứng dụng thực tế của hệ thống, một số hướng mở rộng trong tương lai bao gồm:

* Áp dụng thuật toán dể xây dựng các mô hình trong các lĩnh vực như kinh tế và tài chính
* Trực quan trên web để có thể dễ quan sát và thân thiện với người dùng
* Cải thiện thuật toán hồi quy với các hàm mất mát khác để đánh giá độ chính xác

# Mã nguồn

Giải thuật Hồi quy tuyến tính (Linear regression) được nhóm thực hiện trên google colab

Link mã nguồn:

* + Dành cho Air Passenger: [Air Passengers](https://colab.research.google.com/drive/1rZFqo1nKFhUmFVrK4n_PfyfALMVVrwVX?usp=sharing)
  + Dành cho Thermostat Sales: [Thermostat sales](https://colab.research.google.com/drive/17TYuGkzTGRT7DAXbSJ6tO8jlltGyBBBt?usp=sharing)

Do nhóm làm việc trên hai file colab khác nhau nên việc có nhiều code trùng nhau, nên nhóm đã có chú thích nhưng code dùng chung để tránh làm dài, lang mang và gây khó chịu cho người đọc.

Nhóm có ghi chú và thể hiện để người đọc có thể hiểu và biết được nó được dùng chung.

Hầu hết các hình vẽ đều được dùng ở phần hiện thực kết quả và thực hiện bài toán nên nhóm sẽ không đề cập kết quả ở đây.

Phần mã nguồn nhóm chỉ thực hiện code và giải thích chức năng của code đó.

## Cài đặt các thư viện cần thiết

Một số thư viện được sử dụng là:

* + Thư viện NumPy được sử dụng cho các phép toán ma trận và xử lý dữ liệu số. Đây là thư viện rất phổ biến trong Python khi làm việc với dữ liệu dạng số.
  + Thư viện Matplotlib (với pyplot là con của nó) dùng để tạo các biểu đồ, đồ thị. Đây là thư viện phổ biến để trực quan hóa dữ liệu trong Python.
  + Pandas là thư viện mạnh mẽ để xử lý và phân tích dữ liệu. Nó hỗ trợ các cấu trúc dữ liệu như DataFrame, giúp dễ dàng xử lý dữ liệu từ các nguồn như file CSV.
  + Thư viện Scikit-learn (sklearn) chứa các công cụ học máy. Ở đây, LinearRegression được nhập vào để thực hiện mô hình hồi quy tuyến tính.
  + mean\_squared\_error là hàm tính toán sai số bình phương trung bình (MSE) của mô hình, dùng để đánh giá mức độ chính xác của dự đoán.
  + r2\_score là hàm tính toán R² (hệ số xác định), dùng để đánh giá mức độ mà mô hình giải thích được sự biến thiên trong dữ liệu.

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  import pandas as pd  from sklearn.linear\_model import LinearRegression  from sklearn.model\_selection import train\_test\_split  from sklearn.metrics import mean\_squared\_error  from sklearn.metrics import r2\_score |

## Đọc và tiền xử lý dữ liệu

* Tạo đường dẫn đến dữ liệu
* Đọc dữ liệu với định dạng file csv với thư viện pandas
* Chuẩn hóa dữ liệu
* **Đây là đoạn code dùng chung**
* **Thay đổi tên và đường dẫn**

|  |
| --- |
| # Đường dẫn tới file CSV  air\_passengers='link'  # Đọc dữ liệu  air\_passengers\_data = pd.read\_csv(air\_passengers) #Có thể thay thế nếu dùng tập khác  print(air\_passengers\_data.head())  # Lấy cột giá trị  values = air\_passengers\_data.iloc[:, 0]  y = values.values.reshape(-1, 1)  N = len(y)  print("Data size:", N)  # Tạo chỉ số tuần làm đặc trưng (biến độc lập)  X = np.arange(N).reshape(-1, 1)  # Thêm cột bias (1) vào X  X\_bias = np.hstack([np.ones((N, 1)), X]) |

## Chia dữ liệu thành train và test

1. train\_test\_split(X\_bias, y, test\_size=0.2, random\_state=42):
   * train\_test\_split chia dữ liệu thành 2 phần: tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test).
   * X\_bias là đầu vào (dữ liệu đã thêm hệ số chệch nếu có), còn y là đầu ra (số lượng hành khách).
   * test\_size=0.2: Chia 20% dữ liệu vào tập kiểm tra và 80% vào tập huấn luyện.
   * random\_state=42: Đảm bảo kết quả chia dữ liệu mỗi lần chạy đều giống nhau.
2. print("Train size:", X\_train.shape[0], "| Test size:", X\_test.shape[0]):
   * In ra kích thước của tập huấn luyện và tập kiểm tra (số lượng mẫu).

|  |
| --- |
| from sklearn.model\_selection import train\_test\_split  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_bias, y, test\_size=0.2, random\_state=42)  print("Train size:", X\_train.shape[0], "| Test size:", X\_test.shape[0]) |

## Trực quan dữ liệu

### Cho tập dữ liệu Air Passengers

**Biểu diễn phân bố dữ liệu**

|  |
| --- |
| #Trực quan phân bố dữ liệu  plt.plot( values, 'bo ')  plt.show() |

**Biểu đồ cột cho dữ liệu:**

* + months = range(1, 13):
    - Tạo một danh sách các giá trị từ 1 đến 12, đại diện cho 12 tháng trong năm.
  + plt.figure(figsize=(10, 6)):
    - Tạo một cửa sổ vẽ với kích thước 10x6 inch.
  + plt.bar(months, values[:12]):
    - Vẽ biểu đồ cột với trục x là các tháng từ 1 đến 12 và trục y là 12 giá trị đầu tiên trong values (giả sử dữ liệu về số lượng hành khách trong 12 tháng đầu năm).
  + plt.xlabel("Month") và plt.ylabel("Number of Passengers"):
    - Thêm nhãn cho trục x là "Month" và trục y là "Number of Passengers".
  + plt.title("Air Passengers per Month (First Year)"):
    - Thêm tiêu đề cho biểu đồ.
  + plt.xticks(months):
    - Đặt các nhãn cho trục x là các giá trị tháng từ 1 đến 12.
  + plt.show():
    - Hiển thị biểu đồ cột.

|  |
| --- |
| months = range(1, 13) # Assuming each value corresponds to a month  plt.figure(figsize=(10, 6))  plt.bar(months, values[:12]) # Plotting the first 12 values as an example  plt.xlabel("Month")  plt.ylabel("Number of Passengers")  plt.title("Air Passengers per Month (First Year)")  plt.xticks(months)  plt.show() |

**Biểu đồ hộp:**

* + plt.figure(figsize=(8, 6)):
    - Tạo một cửa sổ vẽ với kích thước 8x6 inch.
  + plt.boxplot(values):
    - Vẽ biểu đồ hộp (box plot) cho dữ liệu values (số lượng hành khách). Biểu đồ hộp sẽ hiển thị các thông tin như giá trị trung vị, các phần tử cực trị, và các phần tử ngoại lai (outliers).
  + plt.ylabel("Number of Passengers"):
    - Thêm nhãn cho trục y là "Number of Passengers" (Số lượng hành khách).
  + plt.title("Box Plot of Air Passenger Data"):
    - Thêm tiêu đề cho biểu đồ là "Box Plot of Air Passenger Data".
  + plt.show():
    - Hiển thị biểu đồ hộp.

|  |
| --- |
| import matplotlib.pyplot as plt  plt.figure(figsize=(8, 6))  plt.boxplot(values)  plt.ylabel("Number of Passengers")  plt.title("Box Plot of Air Passenger Data")  plt.show() |

### Cho dữ liệu Thermostat Sales

Biểu diễn phân bố dữ liệu

|  |
| --- |
| #Trực quan phân bố dữ liệu  plt.plot( values, 'bo ')  plt.show() |

Biểu đồ cột:

categories = thermostat\_sales\_data.index:

* Lấy chỉ mục (index) từ dữ liệu thermostat\_sales\_data để sử dụng làm trục x của biểu đồ (ví dụ: tuần hoặc tháng).

values = thermostat\_sales\_data['Thermostat\_sales']:

* Lấy dữ liệu về doanh số bán thermostat từ cột 'Thermostat\_sales' và gán vào biến values.

plt.figure(figsize=(10, 6)):

* Tạo một cửa sổ vẽ với kích thước 10x6 inch.

plt.bar(categories, values):

* Vẽ biểu đồ cột (bar chart) với trục x là categories (chỉ mục) và trục y là values (doanh số thermostat).

plt.xlabel('Index') và plt.ylabel('Thermostat Sales'):

* Thêm nhãn cho trục x là "Index" và trục y là "Thermostat Sales".

plt.title('Thermostat Sales - Bar Chart'):

* Thêm tiêu đề cho biểu đồ.

plt.tight\_layout():

* Điều chỉnh bố cục để tránh các nhãn bị chồng lên nhau.

plt.show():

* Hiển thị biểu đồ.

|  |
| --- |
| categories = thermostat\_sales\_data.index  values = thermostat\_sales\_data['Thermostat\_sales/ AirPassengers ']  # Create a bar chart  plt.figure(figsize=(10, 6))  plt.bar(categories, values)  # Add labels and title  plt.xlabel('Index')  plt.ylabel('Thermostat Sales ')  plt.title('Thermostat Sales - Bar Chart')  # Show the plot  plt.tight\_layout() # Adjust layout to prevent labels overlapping  plt.show() |

**Biểu đồ hộp:**

plt.figure(figsize=(8, 6)):

* Tạo một cửa sổ vẽ với kích thước 8x6 inch.

plt.boxplot(thermostat\_sales\_data['Thermostat\_sales']):

* Vẽ biểu đồ hộp (box plot) cho dữ liệu trong cột 'Thermostat\_sales' của thermostat\_sales\_data. Biểu đồ hộp này sẽ hiển thị thông tin về phân bố dữ liệu như giá trị trung vị, các phần tử cực trị và các phần tử ngoại lai (outliers).

plt.ylabel('Thermostat Sales'):

* Thêm nhãn cho trục y là "Thermostat Sales" (Doanh số thermostat).

plt.title('Thermostat Sales - Box Plot'):

* Thêm tiêu đề cho biểu đồ là "Thermostat Sales - Box Plot".

plt.grid(True):

* Bật lưới (grid) cho biểu đồ, giúp dễ dàng theo dõi các giá trị trên trục y.

plt.show():

* Hiển thị biểu đồ hộp.

|  |
| --- |
| plt.figure(figsize=(8, 6))  plt.boxplot(thermostat\_sales\_data['Thermostat\_sales])  plt.ylabel('Thermostat Sales ')  plt.title('Thermostat Sales - Box Plot')  plt.grid(True)  plt.show() |

## Xây dựng mô hình hồi quy thủ công

### Xây dựng mô hình

* Hàm predict(x, theta):
* Chức năng: Tính toán dự đoán của mô hình dựa trên đầu vào x và tham số theta.
* Hàm compute\_gradient(y\_hat, y\_true, x):
* Chức năng: Tính toán đạo hàm (gradient) của hàm mất mát để cập nhật tham số theta.
* Hàm update\_theta(theta, gradient, lr):
* Chức năng: Cập nhật tham số theta dựa trên gradient và tỷ lệ học (learning rate).
* Hàm compute\_loss(y\_hat, y\_true):
* Chức năng: Tính toán lỗi của mô hình bằng cách sử dụng hàm mất mát Mean Squared Error (MSE).
* **Đây là code dùng chung**

|  |
| --- |
| # 1. Hàm dự đoán  def predict(x, theta):      return x @ theta  # 2. Hàm tính đạo hàm (gradient)  def compute\_gradient(y\_hat, y\_true, x):      return 2 \* x.T @ (y\_hat - y\_true) / x.shape[0]  # 3. Cập nhật tham số  def update\_theta(theta, gradient, lr):      return theta - lr \* gradient  # 4. Hàm tính lỗi MSE  def compute\_loss(y\_hat, y\_true):      return np.mean((y\_hat - y\_true) \*\* 2) |

### Huấn luyện

* Khởi tạo tham số theta và các giá trị liên quan (tỷ lệ học, số vòng lặp tối đa).
* Trong mỗi vòng lặp:
* Tính dự đoán (y\_hat), lỗi (MSE), và gradient.
* Cập nhật tham số theta dựa trên gradient và tỷ lệ học.
* Lưu lỗi vào danh sách và in kết quả mỗi 10.000 vòng lặp.
* **Đây là code dùng chung**

|  |
| --- |
| # Huấn luyện bằng Gradient Descent trên tập train  epoch\_max = 100000  lr = 0.0001  theta = np.zeros((X\_train.shape[1], 1))  # [w0, w1]  losses = []  for epoch in range(epoch\_max):      y\_hat = predict(X\_train, theta)      loss = compute\_loss(y\_hat, y\_train)      gradient = compute\_gradient(y\_hat, y\_train, X\_train)      theta = update\_theta(theta, gradient, lr)      losses.append(loss)      if (epoch + 1) % 10000 == 0:          print(f"Epoch {epoch+1}: Loss = {loss:.4f}, theta = {theta.ravel()}") |

## Kết quả

### Hiển thị các giá trị mất mát:

Dự đoán giá trị trên tập kiểm tra (X\_test).

Tính các chỉ số đánh giá:

* MSE (Mean Squared Error).
* RMSE (Root Mean Squared Error).
* R² (Hệ số xác định).

In kết quả các độ đo trên.

Tính và hiển thị phương trình hồi quy.

**Đây là code dùng chung**

|  |
| --- |
| # Dự đoán trên tập test  pred1 = predict(X\_test, theta)  # Các độ đo  MSE\_1 = mean\_squared\_error(y\_test, pred1)  RMSE\_1 = np.sqrt(MSE\_1)  R2\_1 = r2\_score(y\_test, pred1)  print("Mean Squared Error (MSE) of the 1st model:\n", MSE\_1)  print("Root Mean Squared Error (RMSE) of the 1st model:\n", RMSE\_1)  print("R Square of the 1st model:\n", R2\_1)  #In kết quả  print(f"Phương trình hồi quy: y = {theta[0][0]:.2f} + {theta[1][0]:.2f} \* x") |

### Trực quan đường hồi quy trên tập huấn luyện

Đoạn code này vẽ đường hồi quy tuyến tính cho cả hai tập dữ liệu (Thermostat Sales và Air Passengers) trên tập huấn luyện (X\_train, y\_train):

1. Dự đoán trên tập huấn luyện (y\_train\_pred = predict(X\_train, theta)):
   * Dự đoán giá trị đầu ra cho tập huấn luyện sử dụng mô hình với tham số theta.
2. Vẽ đồ thị:
   * plt.scatter: Vẽ các điểm dữ liệu thực tế từ tập huấn luyện (X\_train[:, 1] là giá trị của biến độc lập, y\_train là giá trị thực tế).
   * plt.plot: Vẽ đường hồi quy (dự đoán) trên tập huấn luyện.
   * Các tham số khác: Thiết lập nhãn, tiêu đề, legend và lưới cho đồ thị.
3. Chức năng:

* Vẽ biểu đồ đường hồi quy tuyến tính cho tập huấn luyện trên hai tập dữ liệu: Thermostat Sales và Air Passengers.

**Đây là code dùng chung**

|  |
| --- |
| # Dự đoán trên tập train để vẽ đường hồi quy cho tập train  y\_train\_pred = predict(X\_train, theta)  plt.figure(figsize=(8, 5))  plt.scatter(X\_train[:, 1], y\_train, color='blue', label='Train data')  plt.plot(X\_train[:, 1], y\_train\_pred, color='red', label='Linear Regression (Train)')  plt.xlabel('Tùy theo dữ liệu(Tuần/Tháng)')  plt.ylabel('Tùy theo dữ liệu(Doanh số Thermostat/ Số lượng khách hàng (Passenger)  ')  plt.title('Linear Regression - Tập Train')  plt.legend()  plt.grid(True)  plt.tight\_layout()  plt.show() |

### Trực quan đường hồi quy trên tập kiểm tra

Hàm plot\_regression\_test:

* Hàm này nhận các tham số:
  + X\_test: Dữ liệu đầu vào của tập kiểm tra.
  + y\_test: Giá trị thực tế của tập kiểm tra.
  + y\_test\_pred: Dự đoán của mô hình trên tập kiểm tra.
  + x\_label: Nhãn cho trục x (thường là Tuần hoặc Tháng).
  + y\_label: Nhãn cho trục y (thường là Doanh số Thermostat hoặc Số lượng khách hàng).
  + title: Tiêu đề của biểu đồ (tùy vào dữ liệu là Thermostat Sales hay Air Passengers).

Sử dụng hàm plot\_regression\_test:

* Bạn chỉ cần gọi hàm này với các tham số tương ứng cho mỗi loại dữ liệu:
  + Một lần với các tham số cho Thermostat Sales.
  + Một lần với các tham số cho Air Passengers.
  + **Đây là code dùng chung**

|  |
| --- |
| # Dự đoán trên tập test để vẽ đường hồi quy trên test  y\_test\_pred = predict(X\_test, theta)  plt.figure(figsize=(8, 5))  plt.scatter(X\_test[:, 1], y\_test, color='green', label='Test data')  plt.plot(X\_test[:, 1], y\_test\_pred, color='orange', label='Linear Regression (Test)')  plt.xlabel('Tùy theo dữ liệu(Tuần/Tháng)')  plt.ylabel('Tùy theo dữ liệu(Doanh số Thermostat/ Số lượng khách hàng (Passenger)  )  plt.title('Linear Regression - Tập Test')  plt.legend()  plt.grid(True)  plt.tight\_layout()  plt.show() |

## Xây dựng mô hình hồi quy với thư viện

### Khởi tạo và huấn luyện mô hình:

* model2 = LinearRegression(): Tạo một đối tượng mô hình hồi quy tuyến tính từ thư viện sklearn.
* model2.fit(X\_train[:, 1].reshape(-1, 1), y\_train): Huấn luyện mô hình trên dữ liệu tập huấn luyện. Dữ liệu X\_train[:, 1] được chuyển thành mảng 2 chiều (vì mô hình LinearRegression yêu cầu đầu vào là 2D).
* **Đây là code dùng chung**

|  |
| --- |
| model2 = LinearRegression()  model2.fit(X\_train[:, 1].reshape(-1, 1), y\_train) |

### Tính toán hàm mất mát và hiển thị

* MSE: Tính Mean Squared Error giữa giá trị thực tế và dự đoán.
* RMSE: Tính Root Mean Squared Error (căn bậc hai của MSE).
* R²: Tính hệ số xác định (R²) cho mô hình.
* Tính phương trình hồi quy
* **Đây là code dùng chung**

|  |
| --- |
| # Tính toán các độ đo cho mô hình từ thư viện  MSE\_2 = mean\_squared\_error(y\_test, y\_pred\_lib)  RMSE\_2 = np.sqrt(MSE\_2)  R2\_2 = r2\_score(y\_test, y\_pred\_lib)  print("\nMean Squared Error (MSE) of the 2nd model (Library):", MSE\_2)  print("Root Mean Squared Error (RMSE) of the 2nd model (Library):", RMSE\_2)  print("R Square of the 2nd model (Library):", R2\_2)  # In các hệ số của mô hình từ thư viện  print(f"Phương trình hồi quy (Library): y = {model2.intercept\_[0]:.2f} + {model2.coef\_[0][0]:.2f} \* x") |

### Trực quan đường hồi quy

* Hàm plot\_regression\_result có thể nhận các tham số khác nhau cho tập Train và Test (ví dụ: X\_train, y\_train, X\_test, y\_test).
* Đối với tập Test, bạn sẽ truyền vào X\_test, y\_test và dự đoán từ mô hình.
* Tương tự với tập Train, bạn truyền vào X\_train, y\_train và dự đoán từ mô hình.
* Chỉ cần thay đổi nhãn và tiêu đề phù hợp với dữ liệu.
* **Đây là code dùng chung**

|  |
| --- |
| # Biểu đồ kết quả của mô hình từ thư viện trên tập test  plt.figure(figsize=(8, 5))  plt.scatter(X\_test[:, 1], y\_test, color='purple', label='Test data')  plt.plot(X\_test[:, 1], y\_pred\_lib, color='brown', label='Linear Regression (Library)')  plt.xlabel('”Tuần/Tháng”')  plt.ylabel('”Doanh số Thermostat/Air Passengers’')  plt.title('Linear Regression (Library) - Tập Test')  plt.legend()  plt.grid(True)  plt.tight\_layout()  plt.show()  import matplotlib.pyplot as plt  # Visualize the training result using the library model  plt.figure(figsize=(10, 6))  plt.scatter(X\_train[:, 1], y\_train, color='blue', label='Training Data')  plt.plot(X\_train[:, 1], model2.predict(X\_train[:, 1].reshape(-1, 1)), color='red', label='Library LR Prediction (Train)')  plt.xlabel(' Tuần/Tháng ')  plt.ylabel(Doanh số Thermostat/Air Passengers)  plt.title('Linear Regression (Library) - Tập Train')  plt.legend()  plt.grid(True)  plt.tight\_layout()  plt.show() |

## So sánh kết quả từ mô hình thủ công và mô hình từ thư viện

### So sánh trên biểu đồ trực quan của đường hồi quy

Dự đoán trên tập train và test:

* Đối với tập train, em đã thực hiện dự đoán cả bằng mô hình tự xây dựng (y\_train\_pred\_manual) và mô hình từ thư viện (y\_train\_pred\_lib).
* Đối với tập test, em cũng có dự đoán cho cả mô hình tự xây dựng (y\_test\_pred) và mô hình từ thư viện (y\_pred\_lib).

Biểu đồ so sánh:

* Tập train: Bạn sẽ vẽ biểu đồ so sánh giữa dữ liệu thực tế (y\_train) và các dự đoán từ hai mô hình.
* Tập test: Tương tự, em vẽ biểu đồ so sánh giữa dữ liệu thực tế (y\_test) và các dự đoán từ hai mô hình.

**Đây là code dùng chung**

|  |
| --- |
| # Dự đoán trên tập train bằng mô hình tự xây dựng  y\_train\_pred\_manual = predict(X\_train, theta)  # Dự đoán trên tập train bằng mô hình từ thư viện  y\_train\_pred\_lib = model2.predict(X\_train[:, 1].reshape(-1, 1))  plt.figure(figsize=(10, 6))  plt.scatter(X\_train[:, 1], y\_train, color='blue', label='Train data')  plt.plot(X\_train[:, 1], y\_train\_pred\_manual, color='red', linestyle='--', label='Manual LR (Train)')  plt.plot(X\_train[:, 1], y\_train\_pred\_lib, color='green', linestyle='-', label='Library LR (Train)')  plt.xlabel('Tuần')  plt.ylabel('Doanh số Thermostat')  plt.title('So sánh Linear Regression trên Tập Train (Tự xây dựng vs Thư viện)')  plt.legend()  plt.grid(True)  plt.tight\_layout()  plt.show()  # So sánh trực quan kết quả trên tập test  plt.figure(figsize=(10, 6))  plt.scatter(X\_test[:, 1], y\_test, color='green', label='Test data')  plt.plot(X\_test[:, 1], y\_test\_pred, color='red', linestyle='--', label='Manual LR (Test)')  plt.plot(X\_test[:, 1], y\_pred\_lib, color='brown', linestyle='-', label='Library LR (Test)')  plt.xlabel('Tuần')  plt.ylabel('Doanh số Thermostat')  plt.title('So sánh Linear Regression trên Tập Test (Tự xây dựng vs Thư viện)')  plt.legend()  plt.grid(True)  plt.tight\_layout()  plt.show() |

### So sánh kết quả dựa trên hàm mất mát

In ra so sánh MSE:

* print(f"MSE (Manual): {MSE\_1:.4f} | MSE (Library): {MSE\_2:.4f}"): So sánh Mean Squared Error (MSE) giữa mô hình tự xây dựng và mô hình từ thư viện, với độ chính xác đến 4 chữ số thập phân.

In ra so sánh RMSE:

* print(f"RMSE (Manual): {RMSE\_1:.4f} | RMSE (Library): {RMSE\_2:.4f}"): So sánh Root Mean Squared Error (RMSE) giữa hai mô hình.

In ra so sánh R²:

* print(f"R2 (Manual): {R2\_1:.4f} | R2 (Library): {R2\_2:.4f}"): So sánh R² giữa hai mô hình.

**Đây là code dùng chung**

|  |
| --- |
| # So sánh kết quả với mô hình tự xây dựng  print("\nComparison:")  print(f"MSE (Manual): {MSE\_1:.4f} | MSE (Library): {MSE\_2:.4f}")  print(f"RMSE (Manual): {RMSE\_1:.4f} | RMSE (Library): {RMSE\_2:.4f}")  print(f"R2 (Manual): {R2\_1:.4f} | R2 (Library): {R2\_2:.4f}") |

# Tài liệu tham khảo

Vũ Hữu Tiệp, 2016 <https://machinelearningcoban.com/2016/12/28/linearregression/>

2025, <https://www.geeksforgeeks.org/machine-learning/ml-linear-regression/>

Scikit-learn: [LinearRegression — scikit-learn 1.7.1 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html)