

ĐẠI HỌC HUẾ

**KHOA K**Ỹ **THU**Ậ**T VÀ CÔNG NGH**Ệ

🙠🙟🕮🙝🙣

**BÁO CÁO ĐỒ ÁN**

**Học kỳ I, năm học 2024 – 2025**

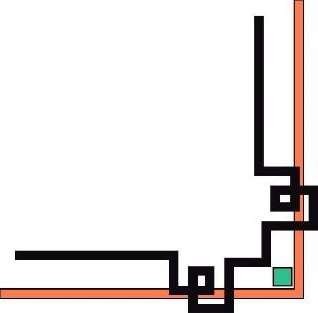
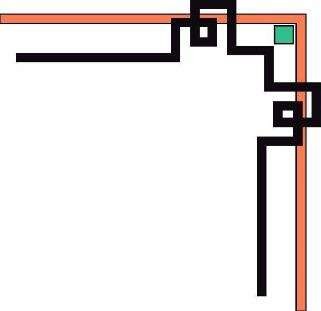
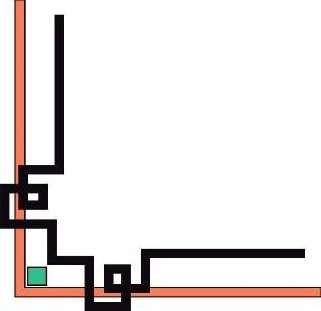
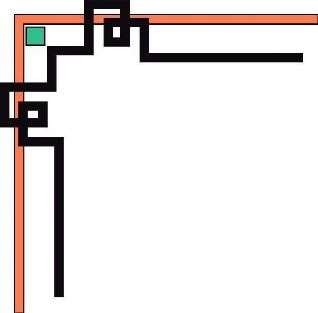
# Học phần: Học máy 1

# Tìm hiểu phương pháp ứng dụng mô hình hồi quy tuyến tính dự đoán tiền lương dựa trên số năm kinh nghiệm làm việc.

**Số phách**

*(Do hội đồng chấm thi ghi)*

**Thừa Thiên Huế, tháng năm 2025**



ĐẠI HỌC HUẾ

**KHOA K**Ỹ **THU**Ậ**T VÀ CÔNG NGH**Ệ

🙠🙟🕮🙝🙣

#### 

**BÁO CÁO ĐỒ ÁN**

**Học kỳ I, năm học 2024 – 2025**

**Học phần: Học máy 1**

#### **Giảng viên hướng dẫn: Nguyễn Đăng Trị**

#### **Sinh viên thực hiện: Hoàng Thị Hương Giang**

**Lớp: KHDL&TTNT K4**

**Số phách**

*(Do hội đồng chấm thi ghi)*

#### **Thừa Thiên Huế, tháng năm 2025**

# *LỜI CẢM ƠN*

Quả thực, thành công của mỗi người đều là kết quả của sự chung tay góp sức, của những mối quan hệ tốt đẹp xung quanh. Trong suốt thời gian từ khi bắt đầuhọc tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của Thầy Cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi lời cảm ơn chân thành đến quý Thầy Cô của Khoa Kỹ thuật và Công nghệ - Trường Đại học Huế. Nhờ những kiến thức chuyên môn sâu rộng và kinh nghiệm giảng dạy phong phú, thầy cô đã trang bị cho em hành trang kiến thức vững chắc để sản sàng hoàn thành đồ án một cách thuận lợi nhất.

Đặc biệt, trong quá trình thực hiện đồ án, em đã gặp không ít khó khăn khiến em cảm thấy bỡ ngỡ và lo lắng. Tuy nhiên, dưới sự hướng dẫn tận tình của thầy Nguyễn Đăng Trị, em đã dần tự tin hơn và vượt qua được những trở ngại đó. Thầy không chỉ truyền đạt cho em những kiến thức chuyên môn mà còn giúp sinh viên rèn luyện kỹ năng tư duy logic, khả năng phân tích và tổng hợp thông tin.

Em hiểu rằng, trong quá trình thực hiện đồ án, chắc chắn còn nhiều thiếu sót. Em rất mong nhận được sự góp ý của thầy cô để bản thân có thể hoàn thiện hơn trong các đồ án tiếp theo.

Một lần nữa, em xin chân thành cảm ơn!

# *Mục lục*

*Bìa* *1*

*Phụ bìa 2*

[*LỜI CẢM ƠN*](#_3znysh7) *3*

[*Mục lục*](#_z1ifcctlivto) *4*

[*Phụ lục hình ảnh*](#_j6qmb7kzcs9u) *6*

[*CHƯƠNG I GIỚI THIỆU SƠ LƯỢC VỀ ĐỒ ÁN*](#_tmew5f98tfjo) *8*

[*1.1 LÍ DO CHỌN ĐỀ TÀI*](#_pfiq8zdrcffs) *8*

[*1.2 MỤC TIÊU*](#_nwyndkvfa80) *8*

[*1.3 VỀ DỮ LIỆU*](#_4tanavhfn7q2) *9*

[*CHƯƠNG II ỨNG DỤNG HỒI QUY TUYẾN TÍNH TRONG PHÂN TÍCH DỰ ĐOÁN MỨC LƯƠNG*](#_wcqdytmb8e5v) *10*

[*2.1 SƠ LƯỢC VỀ CÁC BƯỚC THỰC HIỆN*](#_ocskd5j57nmx) *10*

[*2.2 CHI TIẾT CẤU TRÚC, ĐẶC TRƯNG CỦA DATASET*](#_hclvs6gxxu9h) *14*

[*2.3 XỬ LÝ DỮ LIỆU*](#_hi4z0kcdplru) *18*

[*2.4 XÂY DỰNG VÀ HUẤN LUYỆN MÔ HÌNH HỌC MÁY*](#_x6rzryj7xa1y) *31*

[*CHƯƠNG III KẾT LUẬN*](#_s922i2ycatko) *43*

[*Nguồn tham khảo:*](#_qiz58bn7hyji) *44*

[*Link code colab :*](#_fbi0k3jd9br) *44*

[*PHIẾU ĐÁNH GIÁ ĐỒ ÁN/TIỂU LUẬN/BÀI TẬP LỚN*](#_1yyy98l) *45*

# 

# 

***Phụ lục hình ảnh***

##### *Figure 0. Quy trình tạo mô hình học máy*

##### *Figure 1. Các thư viện cần thiết* *Figure 2. Cấu trúc dữ liệu*

##### *Figure 3. Info của dataset*

##### *Figure 4. Biến X và Y*

##### *Figure 6. Mảng X và Y*

##### *Figure 7.* Biểu đồ scatter trực quan hóa X và Y

##### *Figure 8.* Check dữ liệu trùng lặp

##### *Figure 9. Check dữ liệu thiếu*

##### *Figure 10.* Xử lý dữ liệu thiếu

##### *Figure 11.* Dữ liệu sau khi được xử lý

##### *Figure 12. Dữ liệu sau khi được xử lý*

##### *Figure 13. Chuẩn hóa dữ liệu*

##### *Figure 14.* Tiền xử lý dữ liệu

##### *Figure 15. Tập dữ liệu sau khi bước tiền xử lý*

*Figure 16. Huấn luyện và dự đoán*

##### *Figure 17. Trực quan hóa dự đoán*

##### *Figure 18. Kết quả trực quan dữ liệu*

##### *Figure 19. Đánh giá*

##### *Figure 20. Công thức MSE*

##### *Figure 21. Công thức RMSE*

##### *Figure 22. Công thức MAE*

##### *Figure 23. Công thức R - squared*

##### *Figure 24. Kết quả đánh giá*

##### *Figure 25. Vẽ biểu đồ*

##### *Figure 26. Biểu đồ*

# 

# CHƯƠNG I GIỚI THIỆU SƠ LƯỢC VỀ ĐỒ ÁN

## **1.1 LÍ DO CHỌN ĐỀ TÀI**

Em lựa chọn đề tài "**Ứng dụng mô hình hồi quy tuyến tính để dự đoán tiền lương dựa trên số năm kinh nghiệm làm việc**" vì nhiều lý do. Thứ nhất, hồi quy tuyến tính là một kỹ thuật thống kê được sử dụng để tìm mối quan hệ tuyến tính (dạng đường thẳng) giữa hai hoặc nhiều biến số. Nói cách khác, nó giúp dự đoán giá trị của một biến dựa trên giá trị của các biến khác, đánh giá mức độ ảnh hưởng của các biến độc lập đến biến phụ thuộc, xây dựng các mô hình để mô tả các quy luật trong dữ liệu. Trong trường hợp này, em muốn khám phá mối liên hệ giữa số năm kinh nghiệm và mức lương, một vấn đề có ý nghĩa thực tiễn lớn trong lĩnh vực nhân sự và kinh tế. Thứ hai, đề tài này cho phép em ứng dụng kiến thức đã học về thống kê và học máy vào một bài toán cụ thể, qua đó rèn luyện kỹ năng xây dựng mô hình, đánh giá mô hình và đưa ra dự đoán. Cuối cùng, kết quả của nghiên cứu này có thể cung cấp thông tin hữu ích cho các nhà tuyển dụng, người lao động và các nhà hoạch định chính sách trong việc đưa ra quyết định liên quan đến lương thưởng.

## **1.2 MỤC TIÊU**

Mục tiêu chính nhằm xây dựng mô hình dự đoán tiền lương, phát triển một mô hình hồi quy tuyến tính có khả năng dự đoán mức lương dựa trên số năm kinh nghiệm làm việc của cá nhân. Phân tích tìm hiểu mối quan hệ giữa số năm kinh nghiệm làm việc và mức lương, xác định mức độ ảnh hưởng của kinh nghiệm đến lương. Đánh giá độ chính xác và hiệu suất của mô hình hồi quy tuyến tính thông qua các chỉ số như R², MSE, hoặc MAE, để đảm bảo mô hình phù hợp với dữ liệu thực tế. Ứng dụng thực tiễn bằng việc cung cấp một công cụ hỗ trợ dự đoán mức lương trong các tổ chức nhân sự hoặc cá nhân, giúp đưa ra quyết định liên quan đến định giá công việc, tăng lương, hoặc lập kế hoạch nghề nghiệp. Đặc biết là sẽ đưa ra các khuyến nghị dựa trên kết quả phân tích, cung cấp các khuyến nghị cho các cá nhân hoặc tổ chức về việc sử dụng kinh nghiệm làm yếu tố quan trọng trong việc định giá lương. Ngoài ra còn có thể mở rộng nghiên cứu bằng cách đề xuất các yếu tố khác có thể kết hợp với số năm kinh nghiệm (ví dụ: trình độ học vấn, kỹ năng chuyên môn, ngành nghề) để nâng cao độ chính xác của mô hình trong tương lai.

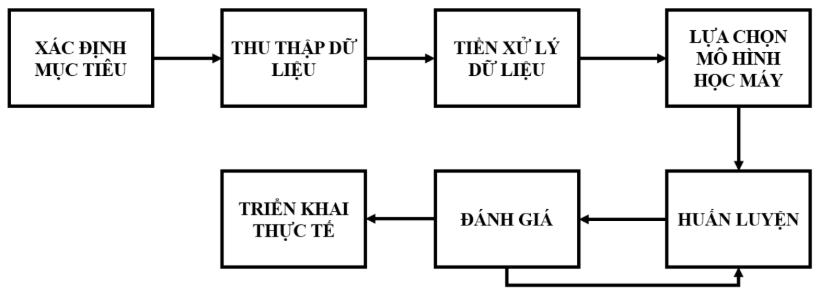
## **1.3 VỀ DỮ LIỆU**

Dữ liệu được sử dụng trong dự án "Ứng dụng mô hình hồi quy tuyến tính để mong đợi tiền lương dựa trên số năm kinh nghiệm làm việc" được thu thập từ nguồn uy tín trên trang web Kaggle : https://www.kaggle.com/datasets.

# CHƯƠNG II ỨNG DỤNG HỒI QUY TUYẾN TÍNH TRONG PHÂN TÍCH DỰ ĐOÁN MỨC LƯƠNG

## **2.1 SƠ LƯỢC VỀ CÁC BƯỚC THỰC HIỆN**

Theo thông thường thì quy trình tạo mô hình học máy hiệu quả nhất sẽ phải trải qua 7 bước như sau :

****

##### *Figure 0. Quy trình tạo mô hình học máy*

Xác định mục tiêu: Đây là bước đầu tiên và quan trọng nhất. Ở đây, ta cần xác định rõ ràng vấn đề mà bạn muốn mô hình học máy giải quyết. Nhằm xác định rõ ràng vấn đề để có thể lựa chọn thuật toán và dữ liệu phù hợp. Mục đích cụ thể trong dự án này là dự đoán tiền lương dựa trên số năm kinh nghiệm làm việc.

Thu thập dữ liệu: Sau khi xác định mục tiêu, bạn cần thu thập dữ liệu liên quan đến vấn đề đó. Nhằm cung cấp đủ dữ liệu chất lượng cho mô hình học tập. Dữ liệu có thể bao gồm thông tin từ các nền tảng trực tuyến, các bộ dữ liệu công khai, dữ liệu do người dùng cung cấp, hoặc được thu thập từ các cảm biến và thiết bị IoT. Lưu ý cần chú ý đến chất lượng của nguồn dữ liệu để không làm ảnh hưởng đến quá trình huấn luyện mô hình học máy. Nguồn dữ liệu ở dự án này được khai thác từ các bộ dữ liệu công khai cụ thể là từ website Kaggle : https://www.kaggle.com/datasets.

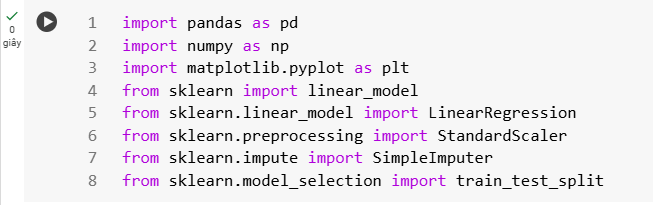
Tiền xử lý dữ liệu: Là dữ liệu thu thập thường chưa sạch và cần được xử lý trước khi đưa vào mô hình. Nhằm làm sạch dữ liệu, loại bỏ nhiễu, chuẩn hóa dữ liệu để đảm bảo chất lượng. Với các kỹ thuật phổ biến như Scaling: Min-max scaling, standardization. Encoding: One-hot encoding, label encoding hay Feature selection: Filter methods, wrapper methods, embedded methods. Mà cụ thể ở dự án này ta sử dụng phương pháp chuẩn hóa Min-max scaling, xử lý các dữ liệu thiếu và chia dữ liệu thành Train data và Test data.

Lựa chọn mô hình học máy: Là dựa vào bản chất của vấn đề và dữ liệu, ta sẽ chọn một mô hình học máy phù hợp. Nhằm chọn mô hình có khả năng giải quyết vấn đề tốt nhất. Các mô hình phổ biến như Hồi quy: Linear regression, polynomial regression, decision tree regression. Phân loại: Logistic regression, support vector machine, random forest, neural network hay clustering: K-means, hierarchical clustering. Vậy mô hình học máy được sử dụng ở đồ án này không gì khác chính là linear regression.

Huấn luyện mô hình: Là dùng dữ liệu đã xử lý để huấn luyện mô hình, giúp mô hình học được các quy luật ẩn trong dữ liệu. Nhằm giúp mô hình học được cách dự đoán chính xác.

Đánh giá mô hình: Là sau khi huấn luyện, ta cần đánh giá xem mô hình có hoạt động tốt hay không. Nhằm đánh giá độ chính xác của mô hình và tìm ra các điểm cần cải thiện. Ta có Các kỹ thuật đánh giá như cross-validation: Chia dữ liệu thành nhiều phần, huấn luyện và đánh giá mô hình trên từng phần. Hay holdout: Chia dữ liệu thành tập huấn luyện và tập kiểm thử.

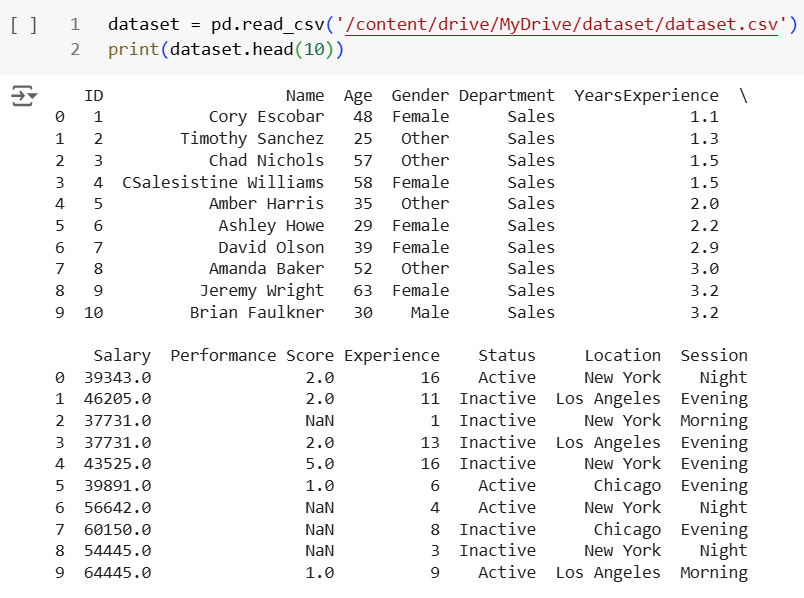
Cuối cùng là triển khai thực tế: Sau khi đã có một mô hình tốt ta có thể triển khai nó vào ứng dụng thực tế. Nhằm sử dụng mô hình để đưa ra dự đoán hoặc quyết định. Các công cụ triển khai như Web application: Flask, Django. API: REST API, GraphQL hay cloud platform: AWS, GCP, Azure. Các vấn đề cần lưu ý đó là tính hiệu quả, khả năng mở rộng và bảo mật.



##### *Figure 1. Các thư viện cần thiết*

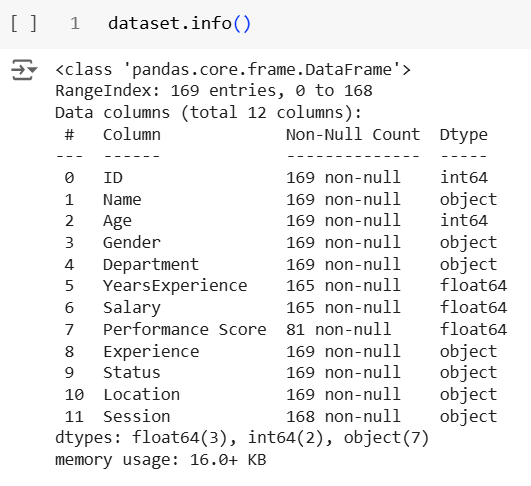
Và trước khi đi vào xây dựng và huấn luyện mô hình học máy ta cần nhập các thư viện cần thiết cho một dự án học máy. Đọc và xử lý dữ liệu (pandas, numpy), trực quan hóa dữ liệu (matplotlib), xây dựng mô hình hồi quy (sklearn.linear\_model), tiền xử lý dữ liệu (sklearn.preprocessing, sklearn.impute), chia dữ liệu thành tập huấn luyện và tập kiểm thử (sklearn.model\_selection).

## **2.2 CHI TIẾT CẤU TRÚC, ĐẶC TRƯNG CỦA DATASET**

****

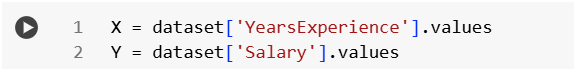
##### *Figure 2. Cấu trúc dữ liệu*

Trong tập dữ liệu dataset này ta có tổng là 12 cột dữ liệu về ‘Name’ thể hiện tên, ‘Age’ thể hiện tuổi trong khoảng độ tuổi từ 18 - 65, ‘Gender’ thể hiện giới tính Female - Male - Other, ‘Department’ thể hiện vị trí việc làm đa số là Sales, ‘YearsExperience’ thể hiện số năm kinh nghiệm làm việc, ‘Salary’ thể hiện giá trị về tiền lương tính theo đơn vị Đô La, ‘Performance Score’, ‘Experience’ thể hiện hiệu suất làm việc, ‘Status’ biểu thị trạng thái gồm Active - Inactive , ‘Location’ thể hiện vị trí làm việc cụ thể là ba thành phố lớn ở Hoa Kỳ gồm New York - Chicago - Los Angeles, và ‘Session’ cho ta biết về phiên làm việc Moring - Evening - Night.



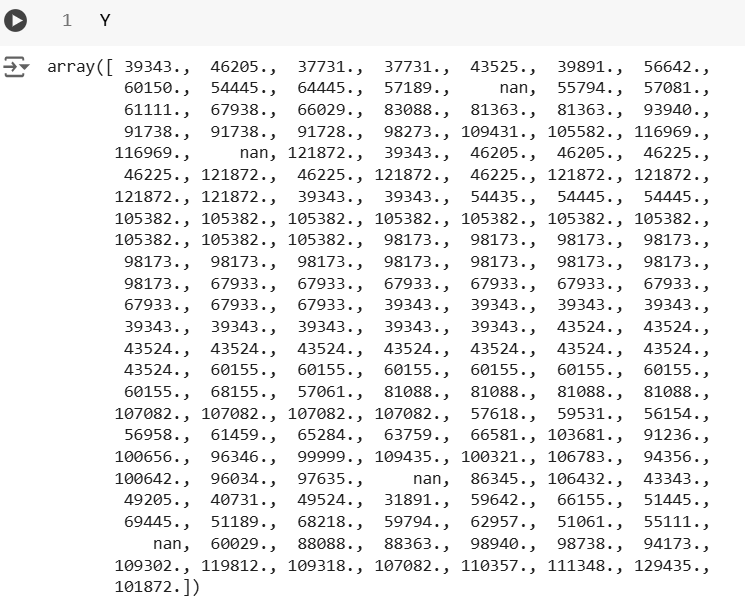
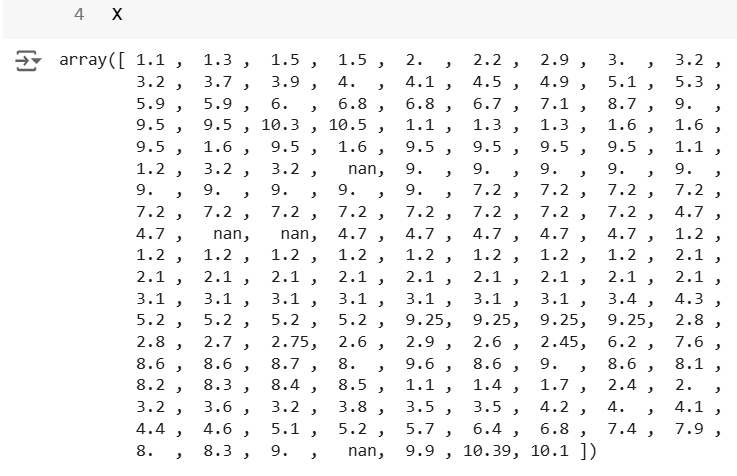
##### *Figure 3. Info của dataset*

Tiếp theo ta có thể dùng dòng lệnh *Dataset.info()* để nắm tổng quan về các thông tin cụ thể của tập dữ liệu. Giá trị trả về cho thấy dữ liệu có 169 đối tượng ( hàng ), mỗi đối tượng được mô tả bởi 12 thuộc tính ( cột ), có 3 kiểu dữ liệu chính int, float và object. Ngoài ra cũng hiển thị số lượng giá trị không null (không rỗng) trong mỗi cột. Ví dụ, cột "Name" có 169 giá trị không null, nghĩa là tất cả các hàng đều có giá trị cho cột này.



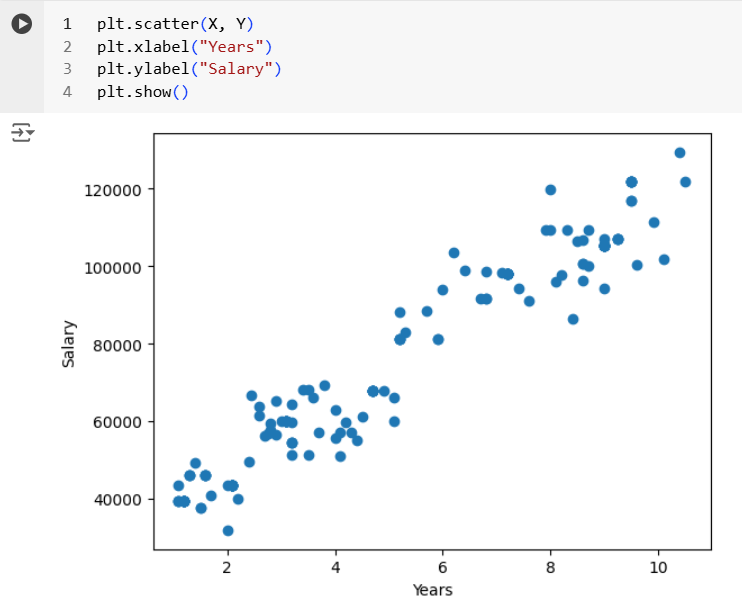
##### *Figure 4. Biến X và Y*

Dù vậy trong dự án " Ứng dụng mô hình hồi quy tuyến tính để dự đoán tiền lương dựa trên số năm kinh nghiệm làm việc " ta chỉ xử dụng hai thuộc tính chính đó là ‘YearsExperience’ và ‘Salary’ nên trước tiền ta thực hiện việc trích xuất hai cột dữ liệu cụ thể từ một DataFrame (bảng dữ liệu) trong Pandas và gán chúng vào hai biến X và Y.



##### *Figure 6. Mảng X và Y*

kết quả ta có hai mảng như trên.



##### *Figure 7. Biểu đồ scatter trực quan hóa X và Y*

Tiếp theo để trực quan hóa dữ liệu ta sử dụng các dòng lệnh như sau :

*plt.scatter(X, Y)*: Lệnh này tạo ra một biểu đồ scatter, trong đó mỗi điểm có tọa độ (x, y) tương ứng với một cặp dữ liệu (số năm kinh nghiệm, mức lương).

*plt.xlabel("Years"):* Đặt nhãn cho trục hoành là "Years" (số năm).

*plt.ylabel("Salary"):* Đặt nhãn cho trục tung là "Salary" (mức lương).

*plt.show():* Hiển thị biểu đồ trên màn hình.

Trực quan hóa dữ liệu: Biểu đồ scatter giúp chúng ta dễ dàng nhìn thấy xu hướng chung giữa hai biến số: số năm kinh nghiệm và mức lương.

Hiển thị mối quan hệ giữa số năm kinh nghiệm và mức lương: Biểu đồ giúp trực quan hóa xem mức lương có xu hướng tăng theo số năm kinh nghiệm hay không, và nếu có thì mối quan hệ đó mạnh mẽ đến mức nào. Nhìn chung, các điểm dữ liệu có xu hướng tăng dần từ trái sang phải, cho thấy mối quan hệ dương giữa số năm làm việc (Years) và mức lương (Salary). Điều này hoàn toàn hợp lý, vì thông thường, khi số năm kinh nghiệm tăng lên, mức lương cũng có xu hướng tăng theo. Đồng thời không có dấu hiệu của các giá trị ngoại lệ chênh lệch lớn xuất hiện cho thấy trong dữ liệu.

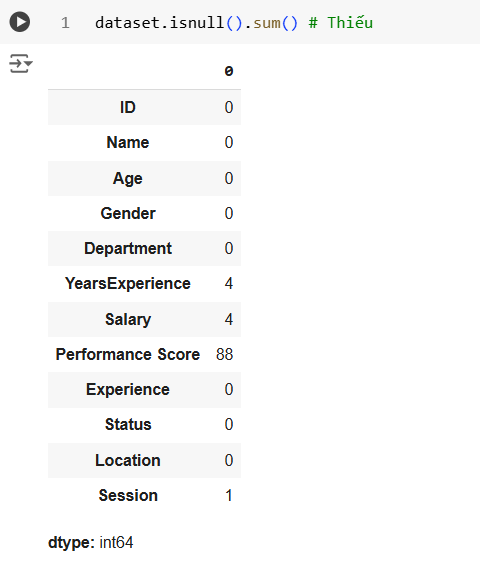
## **2.3 XỬ LÝ DỮ LIỆU**

****

##### *Figure 8. Check dữ liệu trùng lặp*

## Giải thích dòng lệnh *dataset.duplicated().sum()* Dòng lệnh này được sử dụng với chức năng đếm số lượng hàng bị trùng lặp trong bảng dữ liệu. Nhằm kiểm tra chất lượng dữ liệu, đánh giá chất lượng của dữ liệu. Cụ thể trong phần này sẽ kiểm tra từng hàng trong DataFrame, so sánh với các hàng trước đó. Nếu một hàng giống hệt một hàng đã xuất hiện trước đó, nó sẽ trả về giá trị True, ngược lại là False. Sau khi có một Series chứa các giá trị True và False, hàm *.sum()* sẽ tính tổng số giá trị True. Mỗi giá trị True đại diện cho một hàng bị trùng lặp.

## Khi kết quả trả về bằng 0 tức tổng số giá trị True bằng 0, đồng nghĩa với việc không có hàng nào bị đánh dấu là trùng lặp, điều này chứng tỏ dữ liệu không có bản ghi nào bị lặp lại, đây là một dữ liệu chất lượng.



##### *Figure 9. Check dữ liệu thiếu*

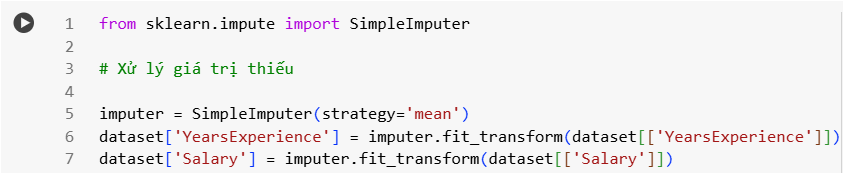
Tương tự hàm *isnull()* này được sử dụng để kiểm tra xem mỗi giá trị trong DataFrame có phải là null (không có giá trị) hay không. Nó sẽ trả về một DataFrame mới có cùng kích thước với DataFrame ban đầu, nhưng các giá trị sẽ là True (nếu là null) hoặc False (nếu không phải null). Sau đó hàm *sum()* sẽ đếm số lượng giá trị True (null) trên mỗi cột. và nhìn vào bảng kết quả trả về cho thấy:

Các cột thuộc tính ‘ID’, ‘Name’, ‘Age’, ‘Gender’, ‘Department’, ‘Experience’, ‘Status’ và ‘Location’ không có giá trị null nào.

Còn một số cột thuộc tính đang gặp vấn đề về dữ liệu thiếu (null) như "YearsExperience", "Salary", và "Session". Cụ thể "YearsExperience" xuất hiện 4 giá trị null , "Salary" xuất hiện 4 giá trị null, và "Session" xuất hiện duy nhất 1 giá trị null

### Vậy tại sao lại có giá trị null? Có nhiều lý do dẫn đến việc xuất hiện giá trị null trong dữ liệu, chẳng hạn như: Dữ liệu chưa được thu thập đầy đủ do lỗi trong quá trình thu thập dữ liệu, người dùng không nhập thông tin, hoặc các thiết bị đo đạc bị hỏng. Dữ liệu bị xóa hoặc mất do lỗi trong quá trình xử lý dữ liệu hoặc do lỗi hệ thống. Dữ liệu không áp dụng, giá trị null có thể được sử dụng để biểu thị rằng một giá trị không áp dụng cho một quan sát cụ thể.

Việc xuất hiện các giá trị null trong dữ liệu sẽ ít nhiều ảnh hưởng đến kết quả phân tích chẳng hạn như kết quả không chính xá, nhiều thuật toán và mô hình học máy không thể xử lý các giá trị null. Việc bỏ qua chúng hoặc xử lý chúng một cách không phù hợp có thể dẫn đến các kết quả phân tích không chính xác và gây hiểu lầm. Giảm độ tin cậy của mô hình, các mô hình được xây dựng trên dữ liệu chứa nhiều giá trị null thường có độ tin cậy thấp hơn và khả năng tổng quát kém hơn. Gây ra lỗi trong quá trình tính toán, lỗi khi thực hiện phép tính: Nhiều phép tính toán học và thống kê không thể thực hiện được với các giá trị null. Làm giảm chất lượng dữ liệu, thiếu thông tin, giá trị null làm giảm lượng thông tin có sẵn trong dữ liệu, điều này có thể ảnh hưởng đến khả năng khám phá các mối quan hệ và xu hướng trong dữ liệu. làm cho dữ liệu không đồng nhất: Sự tồn tại của giá trị null làm cho dữ liệu trở nên không đồng nhất và khó xử lý. Ảnh hưởng đến quá trình ra quyết định, quyết định sai lầm: Nếu dữ liệu được sử dụng để đưa ra quyết định kinh doanh quan trọng mà không được xử lý đúng cách, các quyết định đó có thể dựa trên thông tin không đầy đủ và dẫn đến những hậu quả tiêu cực.



##### *Figure 10. Xử lý dữ liệu thiếu*

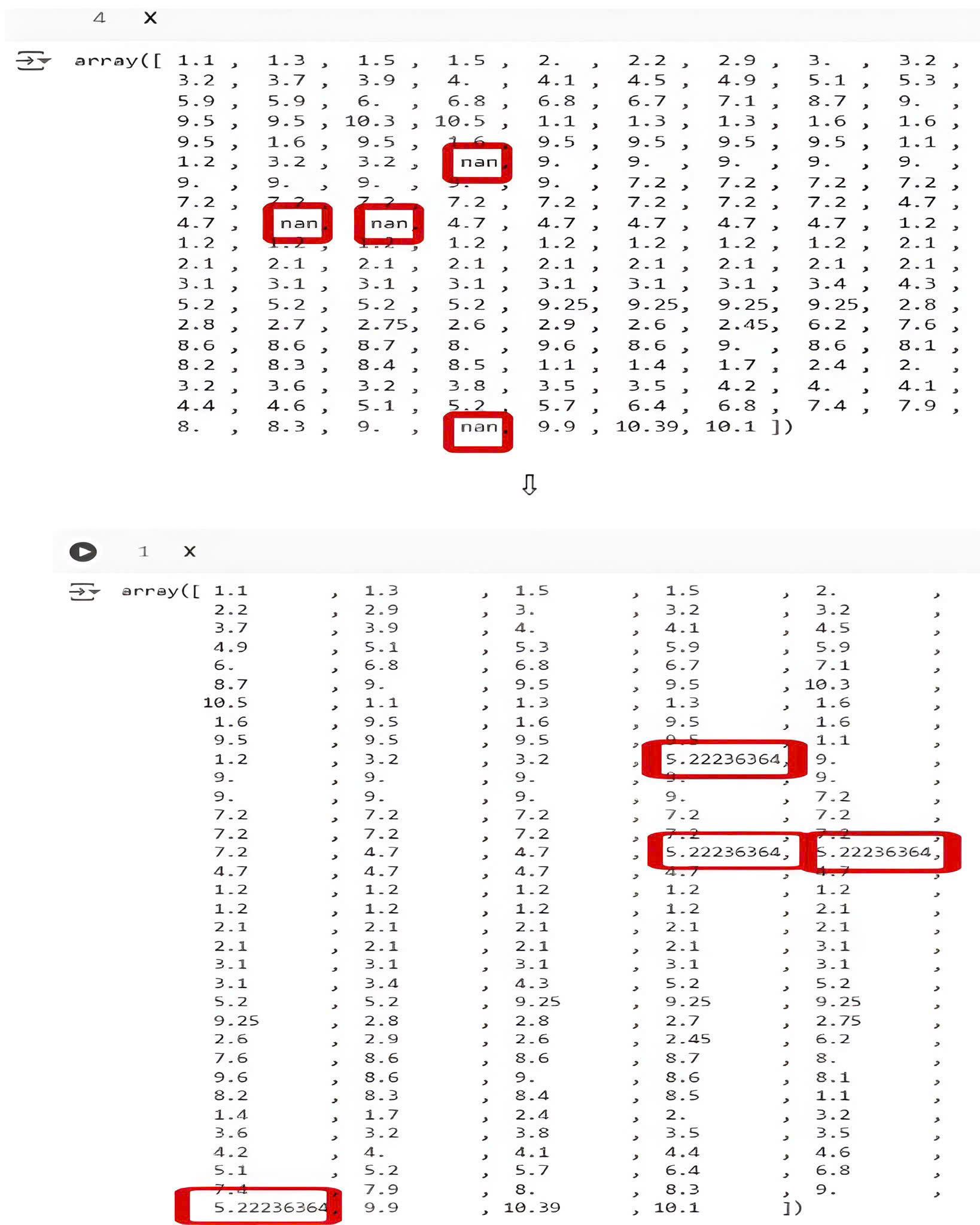
## Các cách xử lý giá trị null phổ biến đó là xóa bỏ các dòng hoặc cột chứa giá trị null và điền vào các giá trị null bằng các giá trị mean, giá trị trung vị, …. Tuy nhiên, phương pháp xóa bỏ các dòng hoặc cột chứa giá trị null chỉ nên sử dụng khi số lượng giá trị null không quá lớn và không ảnh hưởng đáng kể đến kích thước của dữ liệu nên phương pháp này không phổ biến và ít khi được sư dụng. Vậy nên ta sẽ sử dụng phương pháp điền vào các giá trị null bằng các giá trị thay thế cụ thể ở đây ta sẽ thay thế bằng các giá trị mean. Vì trong mô hình huấn luyện chỉ cần đến hai thuộc tính là "YearsExperience" và "Salary" nên ta chỉ tiến hành thay thế trên hai thuộc tính ấy.

Đoạn mã *imputer = SimpleImputer(strategy='mean')* khởi tạo một đối tượng *SimpleImputer* từ thư viện scikit-learn, với tham số *strategy='mean'* chỉ định phương pháp điền giá trị thiếu bằng giá trị trung bình.

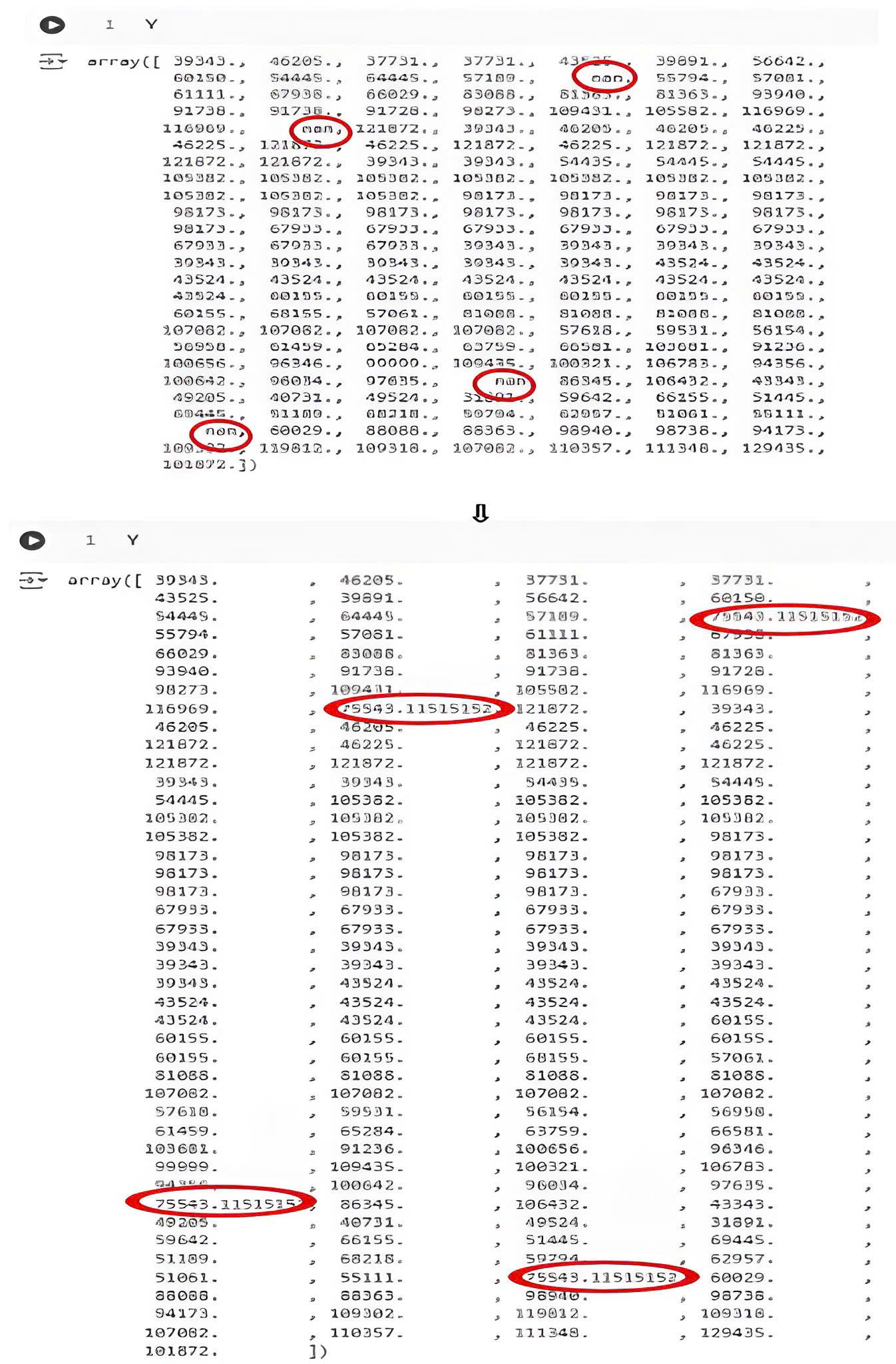
Trong quá trình tiền xử lý dữ liệu, việc xử lý các giá trị thiếu là một bước quan trọng để đảm bảo chất lượng mô hình. Đối tượng *SimpleImputer* được áp dụng lên các cột số liệu liên tục như *YearsExperience* và *Salary* bằng hàm *fit\_transform*.

Hàm *fit\_transform* thực hiện hai tác vụ: Tính toán thống kê: Đối với mỗi cột, hàm này tính toán giá trị trung bình (mean) của các quan sát có giá trị hợp lệ .Điền giá trị thiếu: Sau khi tính toán được giá trị trung bình, hàm này sẽ điền vào tất cả các ô dữ liệu bị thiếu trong cột tương ứng bằng chính giá trị trung bình đó.

Quá trình này đảm bảo rằng các cột số liệu không còn chứa giá trị thiếu, giúp cho các thuật toán học máy hoạt động hiệu quả hơn. Việc sử dụng giá trị trung bình để điền vào giá trị thiếu là một phương pháp đơn giản nhưng hiệu quả trong nhiều trường hợp, đặc biệt khi dữ liệu phân bố tương đối đồng đều.

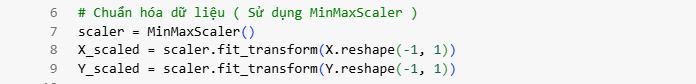


##### *Figure 11. Dữ liệu sau khi được xử lý*



##### *Figure 12. Dữ liệu sau khi được xử lý*

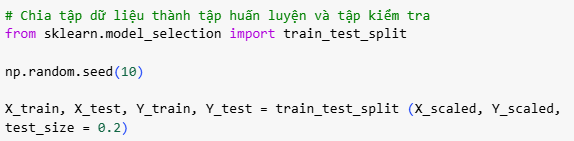
Như vậy ta đã tiến hành thay thế các giá trị null bởi các giá trị trung bình, cụ thể ở mảng X biểu thị các giá trị về *số năm kinh nghiệm* thì các giá trị null đã được thay thế bằng giá trị mean là *5.22236364*, còn ở mảng Y biểu thị các giá trị về *số tiền lương* thì các giá trị null đã được thay thế bằng giá trị mean là *75543*.*11515152.* Như vậy các giá trị null đã được xử lí xong.

**

##### *Figure 13. Chuẩn hóa dữ liệu*

Min-Max Scaling là một phương pháp chuẩn hóa dữ liệu phổ biến và hiệu quả, đặc biệt phù hợp trong các trường hợp dữ liệu có khoảng giá trị khác nhau, khi các đặc trưng trong dữ liệu có khoảng giá trị khác nhau rất lớn, các thuật toán học máy có thể bị ảnh hưởng bởi các đặc trưng có giá trị lớn hơn, dẫn đến kết quả không chính xác. Min-Max Scaling giúp đưa tất cả các đặc trưng về cùng một khoảng giá trị, đảm bảo rằng chúng có tầm quan trọng tương đương trong quá trình học. Cụ thể ở đây ta đang xây dựng một mô hình dự đoán tiền lương dựa trên đặc trưng là số năm kinh nghiệm làm việc. Tiền lương có thể dao động từ 50 đến 120000 Đô la, trong khi số số năm kinh nghiệm chỉ từ 1 đến 10. Trong trường hợp chỉ có một đặc trưng, việc sử dụng Min-Max Scaling có thể mang lại lợi ích, nhưng không quá cần thiết như khi có nhiều đặc trưng. Nó giúp đơn giản hóa tính toán, việc chuẩn hóa một đặc trưng duy nhất thường không làm phức tạp thêm quá trình tính toán. tương thích với các thuật toán, một số thuật toán học máy vẫn yêu cầu dữ liệu đầu vào được chuẩn hóa, ngay cả khi chỉ có một đặc trưng. Tóm lại, Min-Max Scaling là một công cụ hữu ích để chuẩn hóa dữ liệu và cải thiện hiệu suất của các mô hình học máy. Tuy nhiên, nó không phải là phương pháp duy nhất. Việc lựa chọn phương pháp chuẩn hóa phù hợp phụ thuộc vào đặc điểm của dữ liệu và yêu cầu của bài toán.

Để có thể chuẩn hóa dữ liệu ta tiến hành Tạo một đối tượng *scaler*: Dòng này tạo ra một đối tượng scaler thuộc lớp MinMaxScaler. Đối tượng này sẽ được sử dụng để thực hiện quá trình chuẩn hóa dữ liệu. *MinMaxScaler* là một phương thức trong thư viện scikit-learn, được sử dụng để chuẩn hóa dữ liệu vào khoảng từ 0 đến 1. Nó sẽ tìm giá trị lớn nhất và nhỏ nhất trong mỗi feature (tính năng) và sau đó chia tỷ lệ các giá trị khác sao cho chúng nằm trong khoảng từ 0 đến 1. Sau đó tính toán giá trị lớn nhất và nhỏ nhất của mỗi feature trong tập dữ liệu X và áp dụng công thức chuẩn hóa để chuyển đổi các giá trị trong X thành các giá trị mới nằm trong khoảng từ 0 đến 1, tiếp theo ta cần *reshape X* thành một mảng 2 chiều với một cột. Điều này là cần thiết để *MinMaxScaler* có thể xử lý dữ liệu. Cuối cùng là gán kết quả của quá trình chuẩn hóa được vào biến *X\_scaled* bằng dòng lệnh *X\_scaled = scaler.fit\_transform(X.reshape(-1, 1)) làm tương tự với Y.*



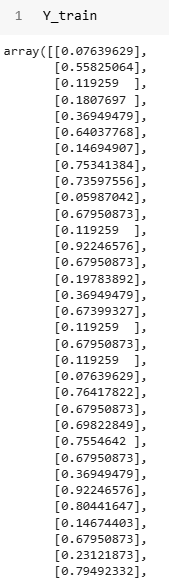
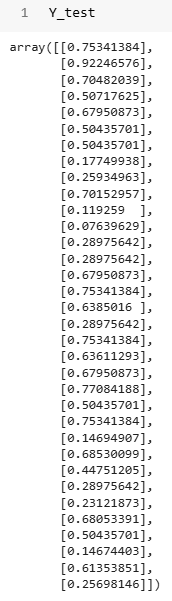
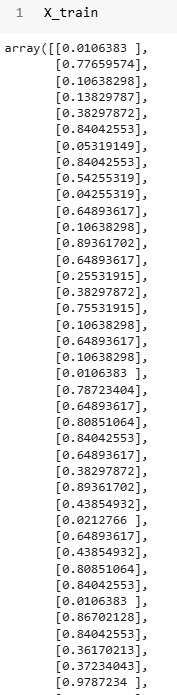
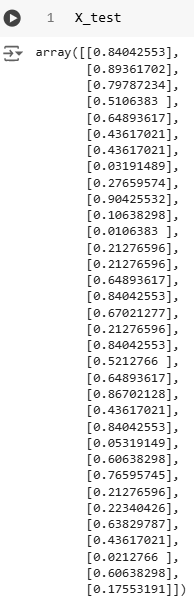
##### *Figure 14. Tiền xử lý dữ liệu*

Bước cuối cùng của tiền xử lý dữ liệu đó là chia tập dữ liệu thành hai phần: một phần để huấn luyện mô hình (train data) và một phần để kiểm tra hiệu quả của mô hình (test data). Việc chia tập dữ liệu thành tập huấn luyện và tập kiểm tra là một bước cực kỳ quan trọng trong quá trình xây dựng và đánh giá mô hình máy học. Việc này cần thiết để đánh giá độ chính xác của mô hình, tránh overfitting, khi một mô hình học quá kỹ các dữ liệu huấn luyện, nó có thể trở nên quá phức tạp và không thể tổng quát hóa tốt cho dữ liệu mới. Việc sử dụng tập kiểm tra giúp ta đánh giá xem mô hình có khả năng dự đoán chính xác trên dữ liệu chưa từng thấy hay không, từ đó phát hiện ra tình trạng overfitting. Đánh giá độ tin cậy, tập kiểm tra cho phép ta đánh giá độ tin cậy của mô hình trên dữ liệu thực tế. Một mô hình có độ chính xác cao trên tập huấn luyện nhưng lại có độ chính xác thấp trên tập kiểm tra thì không phải là một mô hình tốt. Điều chỉnh siêu tham số, tìm kiếm siêu tham số tối ưu, siêu tham số là những tham số của thuật toán học máy mà chúng ta phải thiết lập trước khi huấn luyện. Việc chia tập dữ liệu giúp ta có thể thực hiện quá trình điều chỉnh siêu tham số để tìm ra bộ siêu tham số tốt nhất cho mô hình. Khi ta có nhiều mô hình khác nhau, việc sử dụng tập kiểm tra giúp ta so sánh hiệu suất của các mô hình và chọn ra mô hình tốt nhất.

Đầu tiên ta cần đặt seed cho hàm sinh số ngẫu nhiên của NumPy bằng lệnh np.random.seed(10). Việc đặt seed giúp cho quá trình chia tập dữ liệu trở nên tái lập được. Nghĩa là, nếu bạn chạy lại đoạn code này nhiều lần với cùng một seed, bạn sẽ luôn nhận được kết quả chia giống nhau. Điều này rất hữu ích khi bạn muốn so sánh kết quả của các mô hình khác nhau hoặc muốn tái tạo lại một thí nghiệm. Hạt giống chỉ là một giá trị khởi tạo cho thuật toán sinh số ngẫu nhiên. Dù ta chọn số nào, thuật toán sẽ sử dụng giá trị đó để tạo ra một dãy số ngẫu nhiên có vẻ ngẫu nhiên. Không có số nào đặc biệt hơn số khác khi làm hạt giống. ta có thể chọn bất kỳ số nguyên nào bạn muốn. Dù ta chọn số nào, nếu ta sử dụng cùng một hạt giống cho nhiều lần chạy chương trình, ta sẽ luôn nhận được cùng một dãy số ngẫu nhiên. ở đây ta chọn ngẫu nhiên là số 10.

Cuối cùng cần chia tập dữ liệu bằng dòng lệnh *X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X\_scaled, Y\_scaled, test\_size=0.2).* Trong đó *X\_scaled* và *Y\_scaled* là các biến chứa dữ liệu đầu vào (features) và dữ liệu đầu ra (labels) của ta. test\_size=0.2 nghĩa là 20% dữ liệu sẽ được dùng làm tập kiểm tra (test set), còn lại 80% sẽ là tập huấn luyện (train set). Kết quả của việc chia sẽ được gán vào các biến:

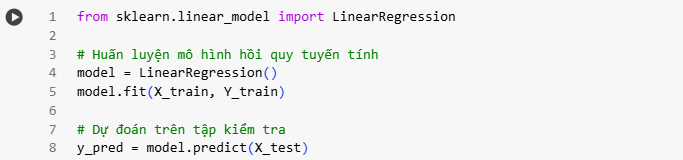
* X\_train: Dữ liệu đầu vào dùng để huấn luyện.
* X\_test: Dữ liệu đầu vào dùng để kiểm tra.
* Y\_train: Dữ liệu đầu ra tương ứng với X\_train.
* Y\_test: Dữ liệu đầu ra tương ứng với X\_test.



##### *Figure 15.* Tập dữ liệu sau khi bước tiền xử lý

**Tóm lại** ta đã thực hiện việc chia ngẫu nhiên một tập dữ liệu thành hai phần để phục vụ cho việc huấn luyện và đánh giá mô hình Machine Learning. Việc chia tập dữ liệu như vậy giúp đảm bảo rằng mô hình được huấn luyện trên một phần dữ liệu và được đánh giá trên một phần dữ liệu khác mà mô hình chưa từng thấy, từ đó giúp đánh giá chính xác hơn khả năng tổng quát của mô hình.

## **2.4 XÂY DỰNG VÀ HUẤN LUYỆN MÔ HÌNH HỌC MÁY**

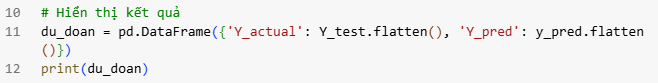
****

##### *Figure 16. Huấn luyện và dự đoán*

Đầu tiên ta cần tạo một đối tượng của lớp *LinearRegression* và gán nó cho biến *model*. Điều này khởi tạo một mô hình hồi quy tuyến tính mới, sẵn sàng để được huấn luyện. Việc tạo đối tượng *model* giống như việc tạo một bản thiết kế cho một ngôi nhà. Bản thiết kế này đã xác định rõ cấu trúc của ngôi nhà (mô hình hồi quy tuyến tính), nhưng các thông số cụ thể (như vị trí các phòng, kích thước...) vẫn chưa được xác định. Đối tượng *model* này giờ đã sẵn sàng để được huấn luyện trên dữ liệu của bạn. Trong quá trình huấn luyện, mô hình sẽ học cách tìm ra mối quan hệ tuyến tính giữa các biến độc lập và biến phụ thuộc, từ đó đưa ra các dự đoán. Tóm lại đây là bước đầu tiên và quan trọng nhất trong quá trình xây dựng một mô hình hồi quy tuyến tính bằng scikit-learn. Nó tạo ra một khung sườn cho mô hình của bạn, sẵn sàng để bạn điền vào các thông tin cụ thể và tiến hành huấn luyện.

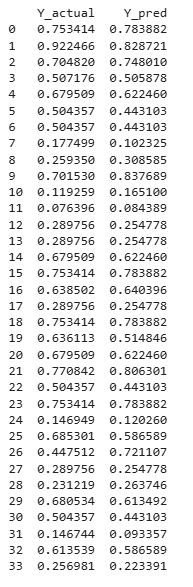
Bước tiếp ta đi vào huấn luyện mô hình bằng dòng lệnh *model.fit(X\_train*, *Y\_train)* trong đó *X\_train* là ma trận chứa các dữ liệu huấn luyện cho các biến độc lập. *Y\_train* là vector chứa các giá trị thực tế tương ứng với các biến độc lập trong *X\_train*. Hàm *model.fit()* sẽ sử dụng thuật toán hồi quy tuyến tính để tìm ra các hệ số (weights) của mô hình, sao cho mô hình có thể dự đoán giá trị của biến phụ thuộc dựa trên các biến độc lập một cách chính xác nhất trên tập dữ liệu huấn luyện.

Tiếp theo ta đến với việc dự đoán, so sánh các dự đoán của mô hình với các giá trị thực tế để đánh giá độ chính xác của mô hình bằng dòng lệnh *y\_pred = model.predict(X\_test)* trong đó *y\_pred*: Đây là một biến mà chúng ta tạo ra để lưu trữ kết quả dự đoán của mô hình. "y" thường được sử dụng để biểu diễn biến phụ thuộc (biến đầu ra) mà chúng ta muốn dự đoán. "pred" là viết tắt của "prediction" (dự đoán). *model* là đối tượng đại diện cho mô hình hồi quy tuyến tính mà bạn đã huấn luyện trước đó. Nó chứa tất cả các thông tin cần thiết để thực hiện dự đoán. *Predict* là một phương thức (method) của đối tượng model. Khi bạn gọi phương thức này, bạn đang yêu cầu mô hình thực hiện việc dự đoán giá trị của biến phụ thuộc dựa trên các giá trị đầu vào mới. *X\_test* là tập dữ liệu mà bạn muốn sử dụng để kiểm tra độ chính xác của mô hình. Nó chứa các giá trị của các biến độc lập (biến đầu vào) mà bạn muốn dự đoán. Thực hiện việc đưa tập dữ liệu kiểm tra *X\_test* vào mô hình đã được huấn luyện *model*. Mô hình sẽ sử dụng các thông số đã học được trong quá trình huấn luyện để tính toán và đưa ra dự đoán cho từng mẫu dữ liệu trong *X\_test*. Các kết quả dự đoán này sẽ được lưu trữ vào biến *y\_pred*.

****

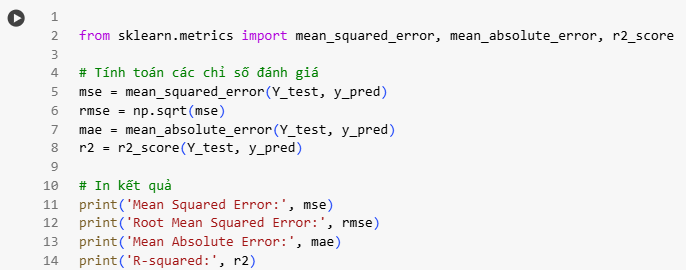
##### *Figure 17. Trực quan hóa dự đoán*

Để trực quan thì ta sẽ tiến hành tạo một *DataFrame* (một bảng dữ liệu) để so sánh trực quan giữa giá trị thực tế (Y\_test) và giá trị dự đoán (y\_pred) của mô hình. biến *du\_doan* sẽ chứa một *DataFrame* có hai cột: *Y\_actual* là giá trị thực tế của biến mục tiêu chứa giá trị của *Y\_test* (là một ma trận hoặc một DataFrame) và "làm phẳng" thành một vector một chiều. Điều này giúp dễ dàng đưa vào DataFrame, và tương tự *Y\_pred* là giá trị dự đoán của mô hình chứa giá trị của *y\_pred* được "làm phẳng" thành một vector một chiều.



##### *Figure 18. Kết quả trực quan dữ liệu*

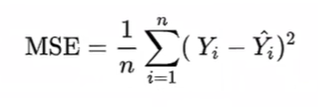
Trên là bảng so sánh giữa giá trị thực tế và giá trị dự đoán của mô hình, sau đó in bảng này ra màn hình để chúng ta dễ dàng quan sát và đánh giá hiệu suất của mô hình. Nhìn vào bảng dự đoán trên ta cung đã có thể đánh giá độ chính xác của mô hình. Tuy nhiên, để đánh giá chi tiết hơn, chúng ta cần sử dụng các chỉ số đánh giá như MSE, RMSE, MAE, R-squared.

****

##### *Figure 19. Đánh giá*

Giá trị *Mean Squared Error (MSE)* để đo lường độ lệch bình phương trung bình giữa giá trị dự đoán (*y\_pred*) và giá trị thực tế (*Y\_test*). Giá trị MSE càng nhỏ, mô hình càng phù hợp với dữ liệu.

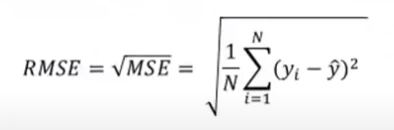
*Công thức:*

****

##### *Figure 20. Công thức MSE*

Giá trị *Root Mean Squared Error (RMSE)* là căn bậc hai của MSE. RMSE có cùng đơn vị với biến mục tiêu, giúp dễ dàng so sánh với dữ liệu gốc.

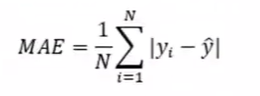
*Công thức:*

****

##### *Figure 21. Công thức RMSE*

Mean Absolute Error (MAE): Đo lường độ lệch tuyệt đối trung bình giữa giá trị dự đoán và giá trị thực tế. MAE ít nhạy cảm hơn MSE với các giá trị ngoại lệ.

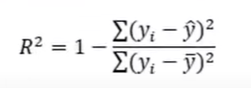
*Công thức:*

****

##### *Figure 22. Công thức MAE*

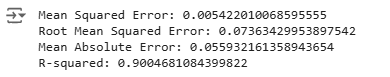
R-squared (R²): Đo lường phần phương sai của biến phụ thuộc (Y\_test) được giải thích bởi biến độc lập (y\_pred). Giá trị R² càng gần 1, mô hình càng giải thích tốt dữ liệu.

*Công thức:*

****

##### *Figure 23. Công thức R - squared*

Vậy các chỉ số trên đã cung cấp những thông tin quan trọng để đánh giá hiệu suất của một mô hình dự đoán.

****

##### *Figure 24. Kết quả đánh giá*

Kết quả trong hình thể hiện các chỉ số đánh giá mô hình dự đoán, bao gồm:

* Mean Squared Error (MSE): 0.00542

MSE đo lường độ lớn trung bình của bình phương sai số giữa giá trị dự đoán và giá trị thực tế. Giá trị này càng nhỏ thì mô hình càng tốt. Với giá trị kết quả trả về là 0.00542 chứng tỏ mô hình khá tốt.

* Root Mean Squared Error (RMSE): 0.07363

RMSE là căn bậc hai của MSE, giữ nguyên đơn vị của dữ liệu ban đầu. Nó cung cấp một chỉ số dễ hiểu hơn về sai số trung bình của mô hình. Với giá trị kết quả trả về là 0.07363 cho thấy sai số trung bình của mô hình ở mức cho phép.

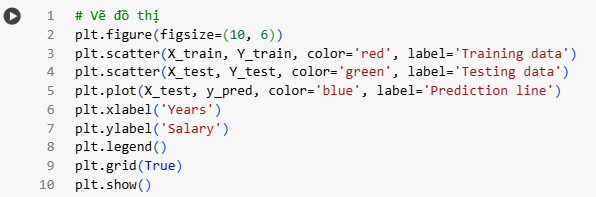
* Mean Absolute Error (MAE): 0.05593

MAE đo lường sai số tuyệt đối trung bình giữa dự đoán và giá trị thực tế. Giá trị nhỏ cũng thể hiện mô hình tốt. Với giá trị kết quả trả về là 0.05593 thể hiện ở mức chấp nhận được.

* R-squared (R²): 0.90047

R² thể hiện mức độ giải thích của mô hình đối với phương sai của dữ liệu thực tế. Giá trị càng gần 1, mô hình càng tốt trong việc dự đoán. Với giá trị kết quả trả về là 0.90047 khá gần với 1 cho thấy độ phù hợp của mô hình với dữ liệu ở mức khá tốt.

### ***Kết luận:*** Với R² = 0.9, mô hình có khả năng giải thích 90% phương sai của dữ liệu, cho thấy mô hình khá tốt. Các giá trị lỗi (MSE, RMSE, MAE) đều nhỏ, chứng tỏ sai số dự đoán không lớn.

****

##### *Figure 25. Vẽ biểu đồ*

##### 

Cuối cùng để hiểu rõ dữ liệu thì đồ thị là công cụ giúp chúng ta có một cái nhìn trực quan về mối quan hệ giữa các biến số, từ đó dễ dàng phát hiện các xu hướng, ngoại lệ hoặc các mẫu quan trọng trong dữ liệu. Giúp ta dễ dàng đánh giá mô hình bằng cách so sánh đường dự đoán với dữ liệu thực tế, chúng ta có thể đánh giá xem mô hình đã được huấn luyện tốt đến mức nào. Ngoài ra để truyền đạt thông tin thì đồ thị là một cách hiệu quả để truyền đạt thông tin đến người khác, đặc biệt là những người không am hiểu kỹ thuật. Vậy nên ta sẽ tiến hành vẽ một biểu đồ *scatter* để trực quan hóa dữ liệu.

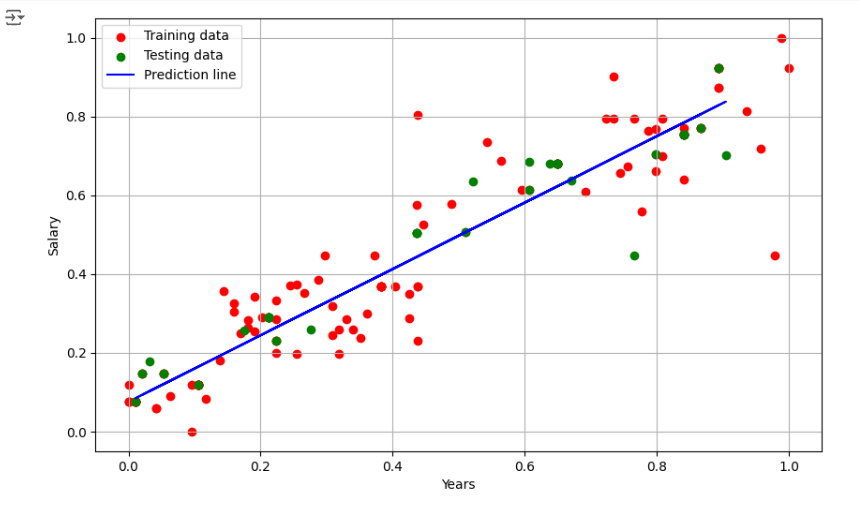
Tiến hành tạo một figure mới bằng lệnh *plt.figure(figsize=(16,6))*, tức là một vùng chứa cho đồ thị, thiết lập kích thước của figure, với chiều rộng là 16 đơn vị và chiều cao là 6 đơn vị. Điều này giúp điều chỉnh kích thước đồ thị cho phù hợp với màn hình hiển thị.

Tiếp theo sử dụng hàm *plt.scatter()* dùng để vẽ biểu đồ phân tán (scatter plot) trong thư viện *Matplotlib*. Biểu đồ phân tán giúp chúng ta trực quan hóa mối quan hệ giữa hai biến số. Trong đó không thể thiếu hai mảng chứa dữ liệu huấn luyện của mô hình là *X\_train* và *Y\_train* . Và để tăng tính truyền đạt và sự rõ ràng ta thêm màu cho các điểm dữ liệu bằng lệnh *color='red'* (màu đỏ). Và không thể quên gán nhãn cho tập dữ liệu này là *"Training data"* để dễ phân biệt khi có nhiều tập dữ liệu khác nhau. Thực hiện tương tự với tập dữ liệu *“Testing data” n*hưng ta sẽ dùng *color='green'* (màu xanh), ta sử dụng hai màu khác nhau để biểu diễn hai tập dữ liệu *"Training data"* và “*Testing data”* nhằm dễ dàng quan sát và so sánh để đi đến các đánh giá khách quan công bằng nhất.

Một thành phần vô cùng quan trọng đó là đường thẳng trên đồ thị, để vẽ được đường thẳng này ta dùng *plt.plot(X\_test, y\_pred, color='blue', label='Prediction line')* với *X\_test* là một mảng chứa các giá trị của biến độc lập (số năm kinh nghiệm) trong tập dữ liệu kiểm tra. Các giá trị này sẽ được biểu diễn trên trục hoành (trục x) của đồ thị. Và *y\_pred* là một mảng chứa các giá trị dự đoán của biến phụ thuộc (mức lương) tương ứng với các giá trị trong *X\_test*. Các giá trị này sẽ được biểu diễn trên trục tung (trục y) của đồ thị. Vì ta đã sử dụng màu đỏ và xanh lá cho các điểm dữ liệu nên ta sẽ sử dụng màu xanh da trời cho được thẳng tuyến tính qua việc sử dụng *color='blue',* cuối cùng là gán nhãn cho đường thẳng là "*Prediction* *line*" (đường dự đoán). Nhãn này sẽ được hiển thị trong legend (hộp chú giải) của đồ thị.

Sau khi đặt tên nhãn "*Years*" vào vị trí trục hoành của đồ thị và "*Salary*" vào vị trí trục tung của đồ thị sẽ giúp người xem hiểu rõ hơn về dữ liệu được biểu diễn trên trục đó. Và để rõ ràng dễ hiểu hơn nữa ta tiến hành cho hiển thị một legend (hộp chú giải) bằng dòng lệnh *plt.legend()*  để giải thích ý nghĩa của các màu và nhãn trên đồ thị và hiển thị luôn cả lưới trên đồ bằng lệnh *plt.grid(True)* thị để dễ dàng đọc giá trị. Cuối cùng cho hiển thị đồ thị trên màn hình bằng dòng lệnh *plt.show().*

.

******

##### *Figure 26. Biểu đồ*

Biểu đồ minh họa mối quan hệ giữa số năm kinh nghiệm (Years) trên trục hoành và mức lương (Salary) trên trục tung, sử dụng mô hình hồi quy tuyến tính đơn giản để phân tích dữ liệu. Dữ liệu huấn luyện (các điểm màu đỏ) được sử dụng để xây dựng và huấn luyện mô hình, trong khi dữ liệu kiểm tra (các điểm màu xanh lá) được sử dụng để đánh giá độ chính xác và khả năng dự đoán của mô hình trên các dữ liệu chưa từng thấy trước đó. Đường màu xanh dương, đại diện cho phương trình hồi quy tuyến tính mà mô hình đã học, phản ánh xu hướng tổng thể của dữ liệu. Quan sát biểu đồ cho thấy mối quan hệ tương đối tuyến tính giữa số năm kinh nghiệm và mức lương: khi số năm kinh nghiệm tăng, mức lương cũng có xu hướng tăng. Đường dự đoán phản ánh xu hướng tổng thể của dữ liệu và phù hợp với các điểm quan sát, cho thấy rằng mô hình hoạt động khá hiệu quả. Hầu hết các điểm kiểm tra nằm gần đường dự đoán, minh chứng cho khả năng dự đoán tốt của mô hình.

# CHƯƠNG III KẾT LUẬN

Phương pháp hồi quy có lẽ là một trong những phương pháp phân tích số liệu thống kê được ứng dụng rộng rãi nhất, đặc biệt là phân tích dự đoán. Đã có nhiều nghiên cứu phân tích thống kê ứng dụng hồi quy như dự đoán giá bất động sản, dự đoán chứng khoán, v.v. Ở mỗi dữ liệu phân tích đều có những đặc trưng khác nhau, tùy vào dữ liệu phân tích mà có những mô hình dự đoán riêng. Và trong khuôn khổ đồ án này cũng không ngoại lệ.

Trong phạm vi đồ án này, đã thực hiện xây dựng mô hình dự đoán trên dữ liệu mức lương ở Hoa Kỳ được cung cấp bởi Kaggle. Trên cơ sở dữ liệu đã cho, xây dựng mô hình dự đoán mức lương dựa trên các đặc trưng số năm kinh nghiệm làm việc mà dữ liệu cung cấp. Sau đó phân tích mức độ ảnh hưởng của đặc trưng lên mức lương để đưa ra một mô hình dự đoán tối ưu nhất .

Có thể còn hạn hẹp nhưng dự án này cũng đã thực hiện theo mô hình khai thác dữ liệu , thực hiện các bước từ tìm hiểu nghiệp vụ, tìm hiểu dữ liệu, chuẩn bị dữ liệu cho đến mô hình hóa và đánh giá mô hình dựa trên các kiến thức về thống kê học mà mà em nghiên cứu được.

# *Nguồn tham khảo:*

<https://colab.research.google.com/drive/1R7KtE5zcqY98iJ-qlvYLyC5f1CH501XC?usp=sharing>

<https://drive.google.com/drive/folders/1Wx16ZTMs9uP3yF1TLNxupptOFusErGf-?usp=sharing>

<https://machinelearningcoban.com/2016/12/26/introduce/>

<https://csr.greenwich.edu.vn/ml-3-hoi-quy-tuyen-tinh/>

<https://www.youtube.com/watch?v=fm8nn374caw>

<https://www.youtube.com/watch?v=iVhzbMjAmwM>

# *Link code colab :*

https://colab.research.google.com/drive/1PPfj4J0LoQYFMxtZAyDTvuOfCLpI9FG4?usp=sharing

***-- HẾT --***

ĐẠI HỌC HUẾ

**KHOA K**Ỹ **THU**Ậ**T VÀ CÔNG NGHỆ**

🙠🙟🕮🙝🙣

# PHIẾU ĐÁNH GIÁ ĐỒ ÁN/TIỂU LUẬN/BÀI TẬP LỚN

**Học kỳ I, năm học 2024 – 2025**

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1** | **Cán bộ chấm thi 2** |
| **Nhận xét:**  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  **Điểm đánh giá của CBCT1:**  Bằng số: ......................................... Bằng chữ: ......................................... | **Nhận xét:**  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  ............................................................  **Điểm đánh giá của CBCT2:**  Bằng số: ......................................... Bằng chữ: ......................................... |

**Điểm kết luận**:.........................................................................................................

**Bằng số**: ..................................................................................................................

**Bằng chữ**: *...............................................................................................................*

*Thừa Thiên Huế, ngày tháng năm 2025*

### **Cán bộ chấm thi 1** **Cán bộ chấm thi 2**

### *(Ký và ghi rõ họ và tên) (Ký và ghi rõ họ và tên)*