

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA TOÁN - ỨNG DỤNG



**BÀI TIỂU LUẬN**

**Breast Cancer Prediction using Machine Learning**

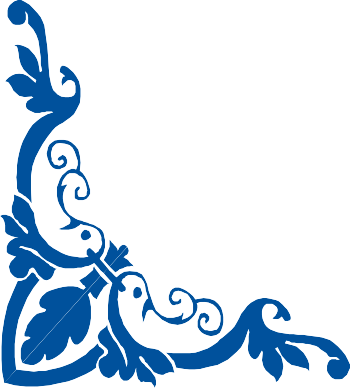
HỌ VÀ TÊN : NGUYỄN QUỐC THUẬN

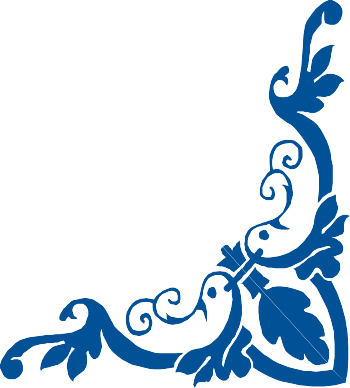
MSSV: 3123580048

HỌC PHẦN : MÁY HỌC

**GIẢNG VIÊN:**

**TS. VŨ NGỌC THANH SANG**



Thành phố Hồ Chí Minh - 2025

**Breast Cancer Prediction using Machine Learning**

**Mục lục**

[**1. Lý do chọn đề tài 1**](#_Toc199445384)

[**2. Mục đích nghiên cứu 1**](#_Toc199445385)

[**3. Đối tượng nghiên cứu 1**](#_Toc199445386)

[**4. Phạm vi nghiên cứu 1**](#_Toc199445387)

[**5. Nhiệm vụ nghiên cứu 2**](#_Toc199445388)

[Chương II. Thông tin và xử lý dữ liệu 2](#_Toc199445389)

[**1.Tập dữ liệu 2**](#_Toc199445390)

[**2. Tiền xử lý dữ liệu 3**](#_Toc199445391)

[**3. Xem trước dữ liệu 5**](#_Toc199445392)

[**3.1 Cấu trúc bằng dữ liệu ( Data Structure ) 6**](#_Toc199445393)

[**3.2 Tên các cột và mô tả sơ bộ (Column Names and Their Meaning) 6**](#_Toc199445394)

[**3.3 Kiểu dữ liệu của các cột (Data Types) 7**](#_Toc199445395)

[**3.4 Giá trị thiếu (Missing Values) 7**](#_Toc199445396)

[**3.5 Một vài dòng đầu tiên (First Rows - Sample Preview) 7**](#_Toc199445397)

[**3.6 Phân bố dữ liệu mục tiêu (Target Balance) 7**](#_Toc199445398)

[**4. Kiểm tra dữ liệu đặc trưng đầu vào 8**](#_Toc199445399)

[**4.1 Phân loại kiểu dữ liệu thông dụng trong dữ liệu đầu vào 9**](#_Toc199445400)

[**4.2 Xác định cột nào là đặc trưng đầu vào, cột nào là nhãn mục tiêu 10**](#_Toc199445401)

[**4.3 Phân tích từng loại kiểu dữ liệu xuất hiện trong đặc trưng đầu vào 10**](#_Toc199445402)

[**4.3.1 Dữ liệu số thức ( float ) – Đặc trưng quan trọng nhất 10**](#_Toc199445403)

[**4.3.2 Dữ liệu phân loại ( categorical ) / chuỗi ( string ) 10**](#_Toc199445404)

[**4.3.2 Cột không phù hợp để đưa vào mô hình 11**](#_Toc199445405)

[**4.3.3 Các vấn đề phổ biến khi kiểm tra dữ liệu 11**](#_Toc199445406)

[**4.3.4 Kết luận về kiểm tra dữ liệu 11**](#_Toc199445407)

[**5. Kiểm tra giá trị bị thiếu 11**](#_Toc199445408)

[**6. Kiểm tra hiển thị và phát hiện các hàng trùng lặp trong dataset 15**](#_Toc199445409)

[Chương III. Phân tách và lựa chọn đặc trưng của dữ liệu 17](#_Toc199445410)

[**1. Phân tách dữ liệu thành dữ liệu số và dữ liệu phân loại 17**](#_Toc199445411)

[**1.1. Mục đích của bước phân tách dữ liệu 17**](#_Toc199445412)

[**1.2. Định nghĩa hai loại dữ liệu 18**](#_Toc199445413)

[**1.3. Ý nghĩa và lợi ích của bước phân tách 18**](#_Toc199445414)

[**1.4. Ví dụ trong bài toán “ Breast Cancer “ 19**](#_Toc199445415)

[**1.5. Tổng kết 20**](#_Toc199445416)

[**2. Trực quan hóa từng cột dữ liệu số trong bộ dữ liệu Breast Cancer 20**](#_Toc199445417)

[**3. Vẽ Biểu Đồ Thanh Ngang cho Dữ Liệu Phân Loại 52**](#_Toc199445418)

[**3.1 Mục tiêu 52**](#_Toc199445419)

[**3.2 Ý nghĩa biểu đồ 52**](#_Toc199445420)

[**3.3 Quy trình thực hiện 52**](#_Toc199445421)

[**3.4. Phân tích dữ liệu phân loại – Cột diagnosis 53**](#_Toc199445422)

[**4. Phân tích mối tương quan giữa các đặc trưng dạng số bằng bản đồ nhiệt (heatmap) 54**](#_Toc199445423)

[**4.1.Phân tích mối tương quan giữa các đặc trưng dạng số bằng biểu đồ nhiệt 56**](#_Toc199445424)

[**4.1.1. Mục đích 56**](#_Toc199445425)

[**4.1.2 Biểu đồ nhiệt 56**](#_Toc199445426)

[**4.1.3 Nhận xét từ biểu đồ 56**](#_Toc199445427)

[**4.1.4 Ý nghĩa đối với mô hình hóa 56**](#_Toc199445428)

[**4.1.5 Kết luận 57**](#_Toc199445429)

[**5. Chuẩn hóa dữ liệu và chia tập huấn luyện – kiểm tra 57**](#_Toc199445430)

[**5.1 Chia tập dữ liệu (Train-Test Split) 57**](#_Toc199445431)

[**5.2 Chuẩn hóa dữ liệu (Feature Scaling) 58**](#_Toc199445432)

[**5.3 Ý nghĩa 58**](#_Toc199445433)

[**6. Loại bỏ các đặc trưng có tương quan cao ( redundant featues ) 59**](#_Toc199445434)

[**6.1 Mục đích 59**](#_Toc199445435)

[**6.2 Phân tích tương quan giữa các đặc trưng 59**](#_Toc199445436)

[**6.3 Cách thực hiện loại bỏ 59**](#_Toc199445437)

[**6.4 Ví dụ từ dữ liệu 60**](#_Toc199445438)

[**6.5 Tác động tích cực sau khi loại bỏ 61**](#_Toc199445439)

[**6.6 Kết quả loại bỏ các đặc trưng có tương quan cao 61**](#_Toc199445440)

[**6.7 Tác động tổng thể đến mô hình 62**](#_Toc199445441)

[**7. Quá trình loại bỏ đặc trưng đệ quy (Recursive Feature Elimination - RFE) 63**](#_Toc199445442)

[**7.1 Tổng quan về RFE 63**](#_Toc199445443)

[**7.2 Quy trình thực hiện RFE 63**](#_Toc199445444)

[**7.3 Ưu điểm của RFE 63**](#_Toc199445445)

[**7.4 Cài đặt trong python 63**](#_Toc199445446)

[**7.5 Kết quả thu được 67**](#_Toc199445447)

[**7.6 Đánh giá hiệu quả 67**](#_Toc199445448)

[**7.7 Kết luận 68**](#_Toc199445449)

[Chương IV: Huấn luyện và đánh giá mô hình dự đoán ung thư vú 68](#_Toc199445450)

[**1. Huấn luyện, đánh giá và trực quan hóa kết quả của mô hình Logistic Regression 68**](#_Toc199445451)

[**1.1 Giới thiệu mô hình Logistic Regression 68**](#_Toc199445452)

[**1.2 Huấn luyện mô hình 68**](#_Toc199445453)

[**1.3 Đánh giá hiệu suất mô hình 70**](#_Toc199445454)

[**1.3.1 Độ chính xác tổng thể (Accuracy) 70**](#_Toc199445455)

[**1.3.3 Ma trận nhầm lẫn (Confusion Matrix) 70**](#_Toc199445456)

[**1.3.4 Trực quan hóa kết quả dữ đoán 71**](#_Toc199445457)

[**1.3.5 Kết luận 71**](#_Toc199445458)

[**2. Huấn luyện mô hình Random Forest 72**](#_Toc199445459)

[**2.1 Giới thiệu 72**](#_Toc199445460)

[**2.2 Tập dữ liệu 72**](#_Toc199445461)

[**2.2.1 Nguồn gốc dữ liệu 72**](#_Toc199445462)

[**2.2.2 Mô tả dữ liệu 72**](#_Toc199445463)

[**2.2.3 Tiền xử lý 73**](#_Toc199445464)

[**2.3 Giới thiệu thuật toán Random Forest 73**](#_Toc199445465)

[**2.4 Huấn luyện mô hình Random Forest 73**](#_Toc199445466)

[**2.5 Kết quả đánh giá mô hình 75**](#_Toc199445467)

[**2.5.1 Classification Report 75**](#_Toc199445468)

[**2.5.2 Ma trận nhầm lẫn (Confusion Matrix) 76**](#_Toc199445469)

[**2.6 Kết luận 76**](#_Toc199445470)

[**3. Huấn luyện Mô hình K-Nearest Neighbors (KNN) 77**](#_Toc199445471)

[**3.1 Giới thiệu 77**](#_Toc199445472)

[**3.2 Tập dữ liệu 77**](#_Toc199445473)

[**3.3 Tiền xử lý dữ liệu 77**](#_Toc199445474)

[**3.3.1 Làm sạch và mã hóa 77**](#_Toc199445475)

[**3.3.2 Tách tập huấn luyện và kiểm tra 77**](#_Toc199445476)

[**3.3.3 Chuẩn hóa dữ liệu 77**](#_Toc199445477)

[**3.4 Xây dựng và huấn luyện Mô hình KNN 77**](#_Toc199445478)

[**3.4.1 Khỏi tạo mô hình 77**](#_Toc199445479)

[**3.4.2 Huấn luyện mô hình 77**](#_Toc199445480)

[**3.5 Đánh giá hiệu năng mô hình 78**](#_Toc199445481)

[**3.5.1 Kết quả dự đoán 78**](#_Toc199445482)

[**3.5.2 Ma trận nhầm lẫn 79**](#_Toc199445483)

[**3.6 Kết luận 80**](#_Toc199445484)

[**4. Huấn Luyện Mô Hình Cây Quyết Định (Decision Tree) 80**](#_Toc199445485)

[**4.1 Giới thiệu 80**](#_Toc199445486)

[**4.2 Tập dữ liệu 80**](#_Toc199445487)

[**4.3 Tiền xử lý dữ liệu 81**](#_Toc199445488)

[**4.3.2 Chuẩn hóa và mã háo 81**](#_Toc199445489)

[**4.3.3 Chia tập dữ liệu 81**](#_Toc199445490)

[**4.4 Huấn Luyện Mô Hình Decision Tree 81**](#_Toc199445491)

[**4.4.1 Khởi tạo mô hình 81**](#_Toc199445492)

[**4.4.2 Huấn luyện mô hình 81**](#_Toc199445493)

[**4.5 Đánh giá hiệu năng mô hình 82**](#_Toc199445494)

[**4.5.1 Kết quả tổng thể 82**](#_Toc199445495)

[**4.5.2 Ma trận nhầm lẫn 82**](#_Toc199445496)

[**4.6 Phân tích và nhận xét 83**](#_Toc199445497)

[**4.7 Kết luận 84**](#_Toc199445498)

[**5. Mô Hình Gaussian Naive Bayes 84**](#_Toc199445499)

[**5.1 Giới thiệu 84**](#_Toc199445500)

[**5.2 Tập dữ liệu 84**](#_Toc199445501)

[**5.3 Tiền xử lý dữ liệu 85**](#_Toc199445502)

[**5.4 Mô Hình Gaussian Naive Bayes 85**](#_Toc199445503)

[**5.4.1 Khởi tạo mô hình 85**](#_Toc199445504)

[**5.4.2 Huấn luyện mô hình 85**](#_Toc199445505)

[**5.5 Đánh giá mô hình 86**](#_Toc199445506)

[**5.5.1 Kết quả chung 86**](#_Toc199445507)

[**5.5.2 Báo cáo phân loại chi tiết 86**](#_Toc199445508)

[**5.5.3 Ma trận nhầm lẫn 86**](#_Toc199445509)

[**5.6 Phân tích nhận xét 87**](#_Toc199445510)

[**5.7 Kết luận 88**](#_Toc199445511)

[Chương V. So sánh và đánh giá các mô hình 88](#_Toc199445512)

[**1. So sánh các Accuracy giữa các mô hình 88**](#_Toc199445513)

[**1.1 Logistic Regression (màu xanh dương nhạt) 89**](#_Toc199445514)

[**1.2 Random Forest (màu xanh lá cây) 90**](#_Toc199445515)

[**1.3 Gaussian Naive Bayes (màu đỏ) 90**](#_Toc199445516)

[**1.4 K-Nearest Neighbors (màu cam) 90**](#_Toc199445517)

[**1.5 Decision Tree (màu tím) 90**](#_Toc199445518)

[**2. Lựa chọn đặc trưng (Feature Selection) 91**](#_Toc199445519)

[**2.1 Mục tiêu của lựa chọn đặc trưng 92**](#_Toc199445520)

[**2.2 Phương pháp L1 Regularization ( Lasso ) 92**](#_Toc199445521)

[**2.3 Phương pháp L2 Regularization ( Ridge ) 92**](#_Toc199445522)

[**2.4 So sánh hiệu quả hai phương pháp 93**](#_Toc199445523)

[**2.5 Cài đặt thực nghiệm (Implementation Details) 93**](#_Toc199445524)

[**2.6 Kết luận 96**](#_Toc199445525)

[Chương VI. Tổng kết 97](#_Toc199445526)

[Lời cảm ơn 98](#_Toc199445527)

[Tài liệu tham khảo 99](#_Toc199445528)

**Chương I . Tổng quan đề tài**

## **1. Lý do chọn đề tài**

Ung thư vú ( Breast Cancer ) là một trong những loại ung thư phổ biến nhất ở phụ nữ trên toàn thế giới, với tỷ lệ mắc ngày càng gia tăng. Việc phát hiện sớm ung thư vú đóng vai trò quan trọng trong việc cải thiện tỷ lệ sống sót và giảm thiểu nguy cơ biến chứng. Tuy nhiên, chẩn đoán ung thư vú truyền thống thường dựa vào xét nghiệm mô học, hình ảnh y khoa và kinh nghiệm của bác sĩ, điều này có thể tốn kém thời gian và tiềm ẩn sai sót.

Với sự phát triển của trí tuệ nhân tạo ( AI ) và máy học ( Machine Learning ), các mô hình dự đoán có thể hỗ trợ bác sĩ trong việc chẩn đoán ung thư vú chính xác hơn bằng cách phân tích dữ liệu y khoa một cách nhanh chóng và hiệu quả. Do đó, nhóm chúng tôi quyết định lựa chọn đề tài “ Dự đoán ung thư vú bằng phương pháo máy học ( Breast Cancer Prediction using Machine Learning )” nhằm nghiên cứu cách áp dụng các thuật toán máy học để hỗ trợ phát hiện ung thư vú sớm hơn và chính xác hơn. Bằng việc nghiên cứu và ứng dụng máy học vào lĩnh vực y tế, chúng tôi hy vọng có thể góp phần cung cấp một phương pháp hỗ trợ hiệu quả trong việc chuẩn đoán ugn thư vú, từ đó giúp giảm thiểu rủi ro và nâng cao chất lượng chăm sóc sức khỏe.

1. **Mục đích nghiên cứu**

Mục đích của nghiên cứu này là tìm hiểu và ứng dụng các thuật tuoans máy học để dự đoán ung thư vú từ đó hỗ trợ các bác sĩ và chuyên gia y tế trong việc chuẩn đoán bệnh một cách nhanh chóng và chính xác hơn. Tìm hiểu lý thuyết về máy học và các thuật toán phổ biến trong dự đoán bệnh, đặc biệt là các phương pháp được áp dụng trong chẩn đoán ung thư vú. Xây dựng mô hình dự đoán ung thư vú dựa trên bộ dữ liệu y tế thực tế, sử dụng các thuật toán máy học như Hồi quy Logistic,…Đánh giá hiệu suất mô hình thông qua các chỉ số như độ chính xác (Accuracy), độ nhạy (Sensitivity), độ đặc hiệu(Specificity) và F1-score. So sánh các mô hình khác nhau để tìm ra thuật toán tối ưu giúp cải thiện độ chính xác của dự đoán. Đề xuất giải pháp cải thiện mô hình, hướng đến ứng dụng thực tiễn trong hỗ trợ chẩn đoán y khoa

## **Đối tượng nghiên cứu**

Đề tài tập trung nghiên cứu các thuật toán máy học trong dự đoán ung thư vú, bao gồm Hồi quy Logistic, Cây quyết định, Rừng ngẫu nhiên, SVM và mạng nơ-ron nhận tạo, nhằm tìm ra mô hình có độ chính xác cao nhất. Dữ liệu sử dụng là bộ dữ liệu từ Kangle, với các đặc trưng quan trọng như kích thước khối u, hình dạng và mật độ tế bào. Nghiên cứu cũng đá gias hiệu suất mô hình dựa trên các tiêu chí như Accuracy, Precision, Recall và F1-score, hướng đến việc tối ưu hóa kĩ năng chẩn đoán ung thư vú bằng công nghệ máy học.

1. **Phạm vi nghiên cứu**

Nghiên cứu này tập trung vào việc ứng dụng các thuật toán máy học trong dự đoán ung thư vú, giới hạn trong phạm vi học có giám sát với các mô hình như Hồi quy Logistics, Cây quyết định, Rừng ngẫu nhiên, SVM và Mạng nơ-ron nhân tạo. Nghiên cứu chỉ tập trung vào quá trình tiền xử lý dữ liệu, xây dụng mô hình, đánh giá hiệu suất dự đoán và so sánh các thuật toán, không đi sâu vào các yếu tố y học chuyên sâu hay thử nghiệm lâm sang

1. **Nhiệm vụ nghiên cứu**

Nghiên cứu này nhằm thực hiện các nhiệm vụ sau: Tìm hiểu lý thuyết về máy học và các thuật toán được sử dụng trong dự đoán ung thư vú. Thu thập và tiền xử lý dữ liệu, sử dụng bộ dữ liệu Breast Cancer để chuẩn hóa, loại bỏ nhiễu và chọn đặc trưng quan trọng. Xây dựng mô hình dự đoán, áp dụng các thuật toán như Hồi quy Logistics, Cây quyết định, Rừng ngẫu nhiên, SVM và Mạng nơ-ron nhân tạo. Đánh giá và so sánh mô hình, sử dụng các chỉ số như Accuracy, Precision, Recall, F1-score để các định thuật toán hiệu quả nhất. Đề xuất giải pháp ciair thiện, tối ưu hóa mô hình nhằm nâng cao độ chính xác và khẳ năng ứng dụng trong thực tế.

# Chương II. Thông tin và xử lý dữ liệu

## **1.Tập dữ liệu**

Trong nghiên cứu này, chúng tôi sử dụng bộ dữ liệu "Breast Cancer Wisconsin (Diagnostic)" – một trong những bộ dữ liệu tiêu chuẩn và nổi bật trong lĩnh vực y sinh học và học máy. Tập dữ liệu này do Tiến sĩ William H. Wolberg thuộc Đại học Wisconsin, Hoa Kỳ cung cấp, được trích xuất từ các hình ảnh sinh thiết của khối u vú và được đăng tải công khai thông qua thư viện UCI Machine Learning Repository. Mục tiêu cốt lõi của tập dữ liệu là hỗ trợ các thuật toán học máy trong việc học cách phân biệt giữa hai loại khối u: **lành tính (Benign)** và **ác tính (Malignant)**, từ đó phục vụ cho chẩn đoán sớm ung thư vú – một trong những căn bệnh gây tử vong hàng đầu ở phụ nữ.

Cụ thể, bộ dữ liệu bao gồm **569 mẫu quan sát**, trong đó mỗi mẫu đại diện cho một bệnh nhân và được mô tả thông qua **32 thuộc tính**. Các thuộc tính này bao gồm một mã định danh (id), một nhãn phân loại (diagnosis), và 30 đặc trưng định lượng được trích xuất tự động từ ảnh chụp kỹ thuật số của mô tế bào ung thư. Mỗi hình ảnh đã được xử lý để xác định nhân tế bào và trích xuất một loạt các chỉ số mô học quan trọng phản ánh hình thái và cấu trúc của khối u.

Cột id chỉ đóng vai trò là định danh kỹ thuật và không chứa thông tin có giá trị cho quá trình học máy nên sẽ bị loại bỏ trong giai đoạn tiền xử lý. Cột diagnosis là nhãn mục tiêu, chứa giá trị chữ cái M (tương ứng với malignant – khối u ác tính) và B (benign – khối u lành tính). Vì hầu hết các thuật toán học máy yêu cầu đầu vào là giá trị số, nên nhãn này sẽ được ánh xạ thành dạng nhị phân: M được gán giá trị 1 và B được gán giá trị 0.

Điểm nổi bật của tập dữ liệu này là sự phong phú và cụ thể của các đặc trưng đầu vào. Ba mươi đặc trưng số thực được tính toán dựa trên ba nhóm đặc tính hình thái học: **giá trị trung bình (mean)**, **sai số chuẩn (standard error)**, và **giá trị lớn nhất (worst)**. Mỗi nhóm này bao gồm cùng một bộ đặc trưng cơ bản, bao gồm: bán kính (radius), độ nhẵn (smoothness), chu vi (perimeter), diện tích (area), độ lõm (concavity), độ đối xứng (symmetry), độ đặc (compactness), và chiều kích phân dạng (fractal dimension). Điều này dẫn đến việc mỗi đặc trưng được thể hiện qua ba góc nhìn khác nhau, ví dụ như radius\_mean, radius\_se, và radius\_worst.

Ý nghĩa sinh học của các đặc trưng này là rất rõ ràng: bán kính và diện tích phản ánh kích thước khối u; độ lõm và độ đặc phản ánh mức độ phức tạp và bất thường trong cấu trúc tế bào; trong khi độ đối xứng hay độ nhẵn lại liên quan đến tính chất hình học của biên tế bào. Các nghiên cứu trước đây đã chỉ ra rằng các đặc trưng như area\_worst, concavity\_mean, và radius\_worst có mức độ tương quan cao với việc phân biệt giữa u lành và u ác.

Một yếu tố quan trọng trong tiền xử lý dữ liệu là việc xác định tính đầy đủ của dữ liệu. Qua quá trình kiểm tra sơ bộ, tập dữ liệu không chứa bất kỳ giá trị thiếu nào. Tất cả các thuộc tính số đều ở định dạng float64, cho phép mô hình học máy tiếp nhận và xử lý trực tiếp mà không cần thao tác làm sạch dữ liệu quá phức tạp.

Về mặt phân bố nhãn, tập dữ liệu không hoàn toàn cân bằng, nhưng cũng không quá lệch. Cụ thể, có khoảng 357 mẫu được chẩn đoán là lành tính (chiếm gần 63%), trong khi số mẫu ác tính là khoảng 212 (chiếm 37%). Điều này có thể gây ra ảnh hưởng nhất định đến mô hình nếu chỉ đánh giá dựa trên độ chính xác (accuracy). Do đó, trong các bước đánh giá mô hình, chúng tôi sẽ sử dụng thêm các chỉ số như precision, recall và F1-score để có cái nhìn toàn diện và chính xác hơn.

Tổng kết lại, tập dữ liệu "Breast Cancer Wisconsin (Diagnostic)" là một bộ dữ liệu chất lượng cao, không chứa dữ liệu thiếu, được xây dựng bài bản và rõ ràng về mặt y học. Với 30 đặc trưng số học mang ý nghĩa sinh học sâu sắc, tập dữ liệu này là lựa chọn lý tưởng để áp dụng và đánh giá hiệu quả của các thuật toán học máy trong chẩn đoán sớm ung thư vú.

## **2. Tiền xử lý dữ liệu**

Trước khi đưa tập dữ liệu vào các mô hình học máy, bước tiền xử lý dữ liệu đóng vai trò vô cùng quan trọng trong việc đảm bảo độ chính xác, tính hiệu quả và độ ổn định của mô hình. Việc tiền xử lý không chỉ giúp dữ liệu trở nên phù hợp với các yêu cầu kỹ thuật của thuật toán, mà còn góp phần cải thiện khả năng học của mô hình, tránh overfitting hoặc bị ảnh hưởng bởi nhiễu và dữ liệu sai lệch. Trong nghiên cứu này, quá trình tiền xử lý bao gồm nhiều bước: loại bỏ cột không cần thiết, mã hóa nhãn phân loại, kiểm tra giá trị thiếu, chuẩn hóa dữ liệu và phân chia tập huấn luyện – kiểm tra.

Bước đầu tiên trong quy trình tiền xử lý là **loại bỏ cột không mang giá trị huấn luyện**. Trong tập dữ liệu ban đầu, cột id là một mã định danh duy nhất cho từng bệnh nhân. Tuy nhiên, đây là một thông tin thuần kỹ thuật, không có ý nghĩa thống kê hay giá trị dự đoán, do đó cần được loại bỏ khỏi dữ liệu đầu vào để tránh ảnh hưởng không mong muốn đến mô hình học.

Tiếp theo, chúng tôi tiến hành **chuyển đổi biến nhãn diagnosis từ dạng ký tự sang dạng số học**, nhằm đáp ứng yêu cầu đầu vào của hầu hết các thuật toán học máy hiện nay. Cụ thể, chúng tôi sử dụng phương pháp mã hóa nhị phân: các giá trị 'M' (Malignant – u ác tính) được ánh xạ thành 1, và 'B' (Benign – u lành tính) được ánh xạ thành 0. Việc chuẩn hóa nhãn theo dạng nhị phân này giúp mô hình phân biệt rõ ràng giữa hai lớp và đảm bảo tính nhất quán trong toàn bộ quá trình huấn luyện và đánh giá.

Một trong những bước quan trọng tiếp theo là **kiểm tra dữ liệu thiếu (missing values)**. Trong tập dữ liệu này, quá trình kiểm tra bằng hàm .isnull().sum() cho thấy không có bất kỳ giá trị thiếu nào trong toàn bộ các cột thuộc tính. Điều này là một ưu điểm đáng kể của tập dữ liệu, vì chúng tôi không cần thực hiện các thao tác xử lý phức tạp như loại bỏ mẫu, nội suy dữ liệu, hoặc thay thế bằng giá trị trung bình. Dữ liệu sạch sẽ góp phần đáng kể vào độ chính xác và độ ổn định của mô hình học máy.

Tuy nhiên, một vấn đề quan trọng khác là **khác biệt về thang đo giữa các đặc trưng**. Trong tập dữ liệu này, các đặc trưng như area\_mean, radius\_mean có giá trị lớn hơn nhiều lần so với các đặc trưng như fractal\_dimension\_se, smoothness\_se, gây ra sự mất cân bằng về tỷ lệ trong khi huấn luyện. Nếu không xử lý, các mô hình dựa trên khoảng cách như K-Nearest Neighbors, SVM hay Logistic Regression có thể bị lệch trọng số, ưu tiên các đặc trưng có giá trị lớn hơn. Để khắc phục, chúng tôi sử dụng **chuẩn hóa Min-Max (MinMaxScaler)**, một phương pháp đưa toàn bộ giá trị của từng đặc trưng về cùng một khoảng giá trị từ 0 đến 1. Phương pháp này không làm mất đi mối quan hệ tương đối giữa các điểm dữ liệu, đồng thời giúp thuật toán học nhanh hơn và chính xác hơn.

Sau khi dữ liệu đã được mã hóa và chuẩn hóa, bước cuối cùng là **phân chia dữ liệu thành tập huấn luyện và tập kiểm tra**. Điều này giúp đánh giá khả năng khái quát hóa (generalization) của mô hình trên những dữ liệu chưa từng thấy. Chúng tôi chia tập dữ liệu theo tỷ lệ 80:20, nghĩa là 80% dữ liệu sẽ được sử dụng để huấn luyện mô hình, trong khi 20% còn lại được giữ lại để đánh giá hiệu suất của mô hình trên dữ liệu mới. Việc chia ngẫu nhiên này được thực hiện bằng hàm train\_test\_split trong thư viện sklearn, có kèm tham số random\_state=42 để đảm bảo tính tái lập kết quả.

Tổng kết lại, toàn bộ quá trình tiền xử lý dữ liệu bao gồm các bước: loại bỏ thuộc tính không liên quan, mã hóa nhãn mục tiêu, xác nhận tính đầy đủ của dữ liệu, chuẩn hóa thang đo thuộc tính, và phân chia dữ liệu hợp lý. Những bước này tuy đơn giản nhưng mang tính quyết định đến hiệu quả của mô hình sau cùng. Nhờ dữ liệu được chuẩn bị cẩn thận và đồng nhất, các thuật toán học máy sau đó có thể tập trung vào việc học ra được quy luật ẩn trong dữ liệu một cách tối ưu nhất.

data.drop(columns=["id"], inplace=True)

data["diagnosis"] = data["diagnosis"].map({"M": 1, "B": 0})

# M: 1 (Malignant), B: 0 (Benign)

X = data.drop(columns=["diagnosis"])

y = data["diagnosis"]

Sau khi nạp dữ liệu, ta cần thực hiện một số bước xử lý để chuẩn bị dữ liệu cho quá trình huấn luyện mô hình học máy.

data.drop(columns=["id"], inplace=True)

Loại bỏ cột không cần thiết:

Cột id chỉ là mã định danh của mỗi bệnh nhân, không mang thông tin hữu ích cho việc phân loại.

Do đó, ta xóa cột này để tránh gây nhiễu cho mô hình.

 Mã hóa nhãn mục tiêu:

Cột diagnosis ban đầu chứa các giá trị dạng chuỗi: "M" (Malignant: Ác tính) và "B" (Benign  Lành tính).

Mô hình học máy yêu cầu đầu vào là dữ liệu số → ta mã hóa:

"M" → 1: biểu thị u ác tính

"B" → 0: biểu thị u lành tínhĐiều này giúp mô hình hiểu và xử lý được nhãn đầu ra.

X: là bảng dữ liệu chứa các đặc trưng đầu vào (features), dùng để huấn luyện mô hình. Ta loại bỏ cột diagnosis vì đây là nhãn.

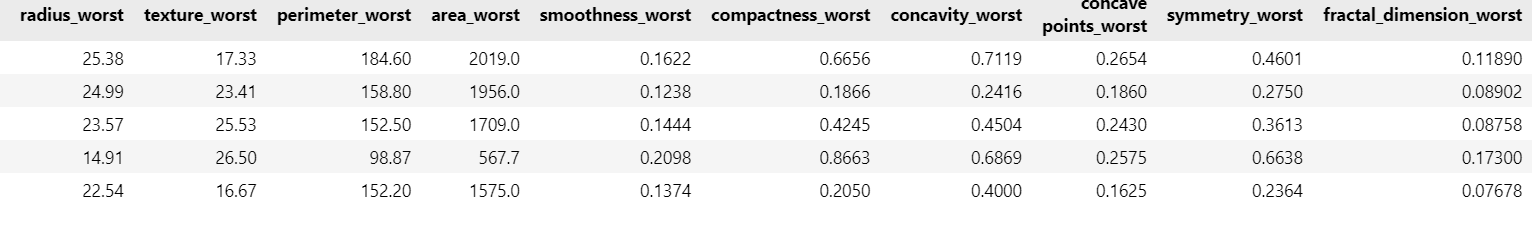
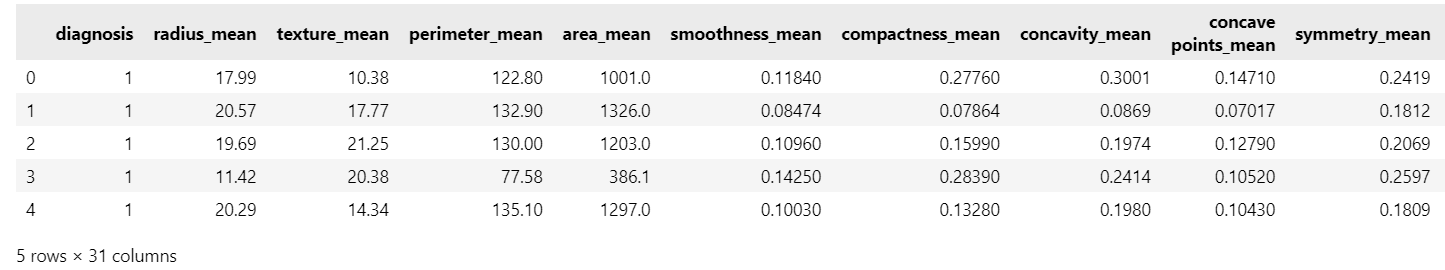
y: là biến mục tiêu (target) chứa nhãn phân loại  chính là cột diagnosis đã được mã hóa ở bước trước.

Việc tách X và y là cần thiết để sử dụng các hàm như train\_test\_split() của scikit-learn trong các bước tiếp theo.

## **3. Xem trước dữ liệu**

Sau khi nạp và xử lý dữ liệu ban đầu, ta sử dụng phương thức .head() của pandas để xem **5 dòng đầu tiên** trong bảng dữ liệu:

data.head()



### **3.1 Cấu trúc bằng dữ liệu ( Data Structure )**

Khi mở tập tin Excel “Breast Cancer.xlsx”, ta sẽ thấy một bảng dữ liệu có dạng hàng – cột. Mỗi **hàng (row)** đại diện cho một bệnh nhân hoặc một mẫu quan sát, trong khi mỗi **cột (column)** biểu diễn một đặc trưng (feature) liên quan đến thông tin y khoa hay chẩn đoán bệnh.

* **Số lượng dòng (samples):** Cho biết có bao nhiêu quan sát được thu thập. Đây là kích thước dữ liệu, giúp đánh giá độ lớn của tập dữ liệu. Nếu số mẫu quá ít, mô hình có thể học chưa tốt.
* **Số lượng cột (features):** Biểu thị số đặc trưng được dùng để mô tả mỗi mẫu. Mỗi cột có thể là một yếu tố lâm sàng như kích thước khối u, mật độ tế bào, hay kết quả xét nghiệm.

### **3.2** **Tên các cột và mô tả sơ bộ (Column Names and Their Meaning)**

Tên các cột thường thể hiện nội dung của từng thuộc tính. Ví dụ, trong một tập dữ liệu ung thư vú điển hình, có thể xuất hiện các cột như:

* id: Mã định danh của bệnh nhân/mẫu.
* diagnosis: Nhãn mục tiêu – cho biết bệnh nhân bị ung thư lành tính (B – Benign) hay ác tính (M – Malignant).
* Các cột còn lại như: radius\_mean, texture\_mean, perimeter\_mean, area\_mean, smoothness\_mean,... là các chỉ số đặc trưng được trích xuất từ ảnh quét tế bào.

Việc quan sát tên cột giúp ta phác họa sơ lược về bản chất dữ liệu – là định lượng hay định tính, và liên quan tới chẩn đoán lâm sàng ra sao.

### **3.3** **Kiểu dữ liệu của các cột (Data Types)**

Dữ liệu trong mỗi cột có thể thuộc nhiều kiểu khác nhau:

* **Số liên tục (float)**: Như kích thước khối u, mật độ tế bào.
* **Số nguyên (integer)**: Như mã ID.
* **Chuỗi văn bản (string/char)**: Ví dụ, nhãn chẩn đoán như “B” hay “M”.

Xác định được kiểu dữ liệu sẽ rất hữu ích để biết cột nào cần chuẩn hóa, cột nào cần mã hóa (encode), hay cột nào sẽ là mục tiêu huấn luyện mô hình (target variable).

### **3.4 Giá trị thiếu (Missing Values)**

Một yếu tố quan trọng khác cần kiểm tra là xem liệu có cột nào có giá trị bị thiếu (null, NaN) hay không. Ví dụ:

* Nếu một vài cột có nhiều dữ liệu bị thiếu, có thể cần loại bỏ hoặc điền giá trị thay thế (impute).
* Nếu một mẫu bị thiếu toàn bộ giá trị, có thể bỏ luôn dòng đó.

Dữ liệu thiếu có thể làm sai lệch kết quả mô hình hoặc gây lỗi trong các bước xử lý.

### **3.5 Một vài dòng đầu tiên (First Rows - Sample Preview)**

Một cách phổ biến để xem nhanh dữ liệu là hiển thị 5–10 dòng đầu tiên. Điều này giúp:

* Đánh giá cách dữ liệu được nhập: có hàng tiêu đề đúng không? Có dữ liệu lạ, giá trị bất thường không?
* Kiểm tra xem có dòng nào chứa metadata hoặc tiêu đề phụ không – có thể cần loại bỏ.
* Nhìn nhanh nhãn diagnosis để xác định bài toán là phân loại 2 lớp (binary classification).

### **3.6 Phân bố dữ liệu mục tiêu (Target Balance)**

Nếu cột diagnosis là nhãn mục tiêu, ta nên xem nhanh tỷ lệ giữa hai giá trị:

* Ví dụ: có bao nhiêu mẫu là M (malignant) và bao nhiêu là B (benign)?
* Nếu dữ liệu bị lệch quá nhiều (ví dụ 90% là B, 10% là M), cần xem xét kỹ lưỡng để xử lý mất cân bằng (imbalance).

**Kết luận về bước xem trước dữ liệu**

Bước xem trước dữ liệu giống như việc **lướt qua “bản đồ”** trước khi bắt đầu hành trình xây dựng mô hình. Nó giúp chúng ta:

* Xác định rõ cấu trúc và đặc trưng của tập dữ liệu.
* Phát hiện ra các vấn đề tiềm ẩn như giá trị thiếu, kiểu dữ liệu sai hoặc lệch phân bố.
* Hiểu rõ hơn về nội dung thông tin trong dữ liệu, từ đó định hướng các bước xử lý và phân tích tiếp theo.

## **4. Kiểm tra dữ liệu đặc trưng đầu vào**

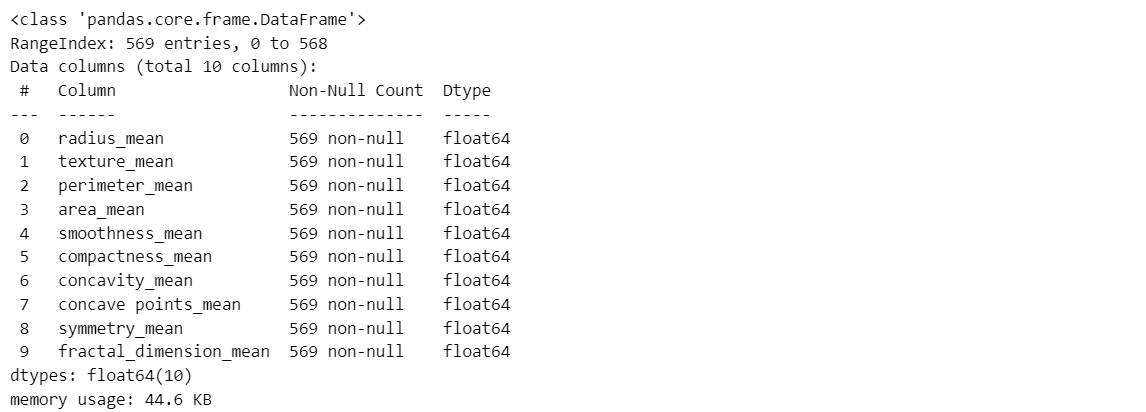
Khi làm việc với dữ liệu, đặc biệt là dữ liệu từ tệp Excel, một bước thiết yếu là **kiểm tra loại dữ liệu của từng cột (hoặc đặc trưng)** để:

* Biết được liệu mô hình có thể hiểu và xử lý đúng các đặc trưng không.
* Phân loại xem dữ liệu là **số hay phân loại**, **rời rạc hay liên tục**, **chuỗi hay boolean**, v.v.
* Phát hiện các vấn đề ẩn như: giá trị bị ghi sai kiểu, dữ liệu số lại được hiểu là chuỗi (do dấu phẩy, ký hiệu, khoảng trắng, v.v.).

Kiểm tra thông tin đặc trưng của 10 dữ liệu đầu

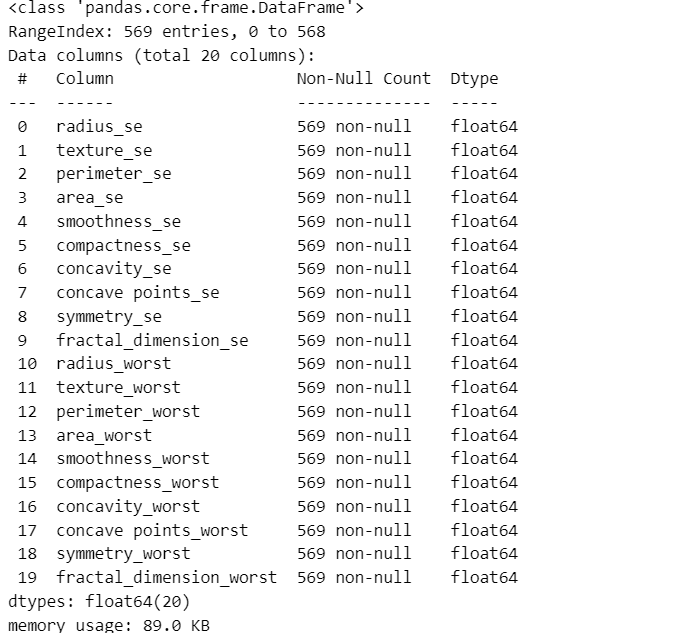
data = data.drop(columns=['diagnosis'])

data[data.columns[:10]].info()



Kiểm tra thông tin đặc trưng của các dữ liệu còn lại

data[data.columns[10:]].info()



### **4.1 Phân loại kiểu dữ liệu thông dụng trong dữ liệu đầu vào**

Trong một bảng dữ liệu, đặc biệt là dữ liệu y học như về bệnh ung thư vú, thường xuất hiện các loại dữ liệu sau:

|  |  |  |
| --- | --- | --- |
| Loại dữ liệu | Mô tả | Ví dụ |
| Số thực (float) | Dữ liệu liên tục, có phần thập phân. | radius\_mean = 14.2 |
| Số nguyên (int) | Dữ liệu rời rạc, không có phần thập phân. | id = 842302 |
| Chuỗi (string / object) | Dữ liệu dạng văn bản, ký hiệu hoặc phân loại. | diagnosis = "M" |
| Boolean (True/False) | Giá trị đúng/sai, có/không. Ít gặp trong dữ liệu này. | has\_symptom = True |
| Ngày tháng (datetime) | Dữ liệu dạng thời gian. Hiếm khi có trong tập này. | exam\_date = 2021-05-13 |

### **4.2 Xác định cột nào là đặc trưng đầu vào, cột nào là nhãn mục tiêu**

Trước khi đi vào chi tiết từng cột, cần phân biệt:

* **Cột mục tiêu (target):** Đây là cột cho biết kết quả cần dự đoán – trong tập này có thể là cột diagnosis.
* **Đặc trưng đầu vào (input features):** Là những cột còn lại dùng để huấn luyện mô hình dự đoán. Đây mới là phần cần kiểm tra kiểu dữ liệu.

Thông thường, trong tập dữ liệu “Breast Cancer” của UCI hoặc Kaggle, các đặc trưng đầu vào sẽ là những chỉ số được rút trích từ ảnh quét tế bào – ví dụ: radius\_mean, texture\_mean, smoothness\_se, v.v.

### **4.3 Phân tích từng loại kiểu dữ liệu xuất hiện trong đặc trưng đầu vào**

### **4.3.1 Dữ liệu số thức ( float )** **– Đặc trưng quan trọng nhất**

Hầu hết các cột trong tập “Breast Cancer” là **chỉ số dạng số thực** vì chúng phản ánh đặc điểm hình học, vật lý và sinh học của khối u:

|  |  |  |
| --- | --- | --- |
| Cột | Ý nghĩa | Kiểu dữ liệu |
| radius\_mean | Bán kính trung bình của nhân tế bào | Float |
| texture\_mean | Độ gồ ghề trung bình | Float |
| area\_mean | Diện tích trung bình | Float |
| compactness\_se | Độ đặc khối – sai số chuẩn | Float |

### **4.3.2 Dữ liệu phân loại ( categorical ) / chuỗi ( string )**

|  |  |  |
| --- | --- | --- |
| **Cột** | **Ý nghĩa** | **Kiểu dữ liệu** |
| **diagnosis** | Chẩn đoán khối u là Lành (B) hoặc Ác (M) | String (Categorical) |
| **ID** | Mã định danh – thường không dùng để huấn luyện | String hoặc Integer |

Lưu ý:

* Cột diagnosis tuy là string (M, B) nhưng sẽ cần được **chuyển sang số** để mô hình hiểu (ví dụ: M = 1, B = 0).
* Nếu Excel lưu diagnosis là “object” hoặc “string”, bạn cần mã hóa nó bằng Label Encoding hoặc One-hot Encoding.

### **4.3.2 Cột không phù hợp để đưa vào mô hình**

Một số cột như id dù có thể ở dạng số nguyên, nhưng **không có giá trị dự đoán**. Mã ID không liên quan đến đặc điểm y học, nên thường sẽ bị loại bỏ

### **4.3.3 Các vấn đề phổ biến khi kiểm tra dữ liệu**

- Kiểu dữ liệu không đúng

* Ví dụ: một cột radius\_mean chứa toàn số, nhưng chỉ vì có một ô bị lỗi đánh máy như “14,2” (dấu phẩy), cả cột có thể bị đọc là “chuỗi”.
* Dữ liệu có thể chứa khoảng trắng, ký hiệu %, đơn vị tính,... khiến nó bị nhận diện sai.
* Giá trị rỗng hoặc bất thường
* Một số ô có thể trống hoặc ghi “n/a”, “?”, hoặc NULL – gây lỗi kiểu dữ liệu nếu không xử lý.

**4.3.4 Kết luận về kiểm tra dữ liệu**

* Việc xác định đúng **kiểu dữ liệu** của từng cột đầu vào là nền tảng để chọn **phương pháp xử lý phù hợp**:
  + Chuẩn hóa số thực.
  + Mã hóa biến phân loại.
  + Loại bỏ cột không cần thiết.
* Nó giúp tránh lỗi khi đưa vào mô hình và tăng hiệu quả dự đoán.

## **5. Kiểm tra giá trị bị thiếu**

data.isnull() trả về một DataFrame có giá trị True hoặc False, trong đó mỗi giá trị True chỉ ra rằng phần tử trong DataFrame đó là giá trị bị thiếu (NaN).

sum() được sử dụng để tính tổng số lượng giá trị thiếu (NaN) trong mỗi cột của DataFrame. Kết quả là một đối tượng Pandas Series, trong đó mỗi phần tử biểu thị số lượng giá trị thiếu trong một cột.

plt.figure(figsize=(6, 4)): Đặt kích thước của hình vẽ (6x4 inch).

sns.heatmap(data.isnull(), cmap='viridis', cbar=False): Sử dụng seaborn.heatmap để vẽ biểu đồ nhiệt (heatmap) thể hiện dữ liệu thiếu.

data.isnull() trả về DataFrame boolean với True là giá trị thiếu.

cmap='viridis' chỉ định sử dụng màu sắc theo bảng màu "viridis" để hiển thị giá trị (giá trị True hoặc False).

cbar=False loại bỏ thanh màu sắc (colorbar).

plt.title('Missing Data Visualization'): Thêm tiêu đề cho biểu đồ.

plt.show(): Hiển thị biểu đồ.

In số lượng dữ liệu bị thiếu:

Print("Missing Data Counts:\n", missing\_data)

In ra số lượng giá trị thiếu trong mỗi cột của DataFrame. Thông tin này giúp người dùng hiểu rõ cột nào có nhiều dữ liệu bị thiếu và cần được xử lý.

Giải thích:

Mục đích: Đoạn mã này giúp người dùng xác định và trực quan hóa các giá trị thiếu trong dữ liệu, từ đó có thể quyết định các phương pháp xử lý, như thay thế giá trị thiếu bằng giá trị khác, hoặc loại bỏ các dòng/cột có quá nhiều giá trị thiếu.

Kết quả mong đợi: Biểu đồ nhiệt giúp nhận diện dễ dàng các vùng có dữ liệu thiếu, và phần in ra số lượng giá trị thiếu cho từng cột giúp người dùng có cái nhìn rõ hơn về mức độ ảnh hưởng của các giá trị thiếu trong bộ dữ liệu.

# Xác định dữ liệu bị thiếu

missing\_data = data.isnull().sum()

# Trực quan hóa dữ liệu bị thiếu

plt.figure(figsize=(6, 4))

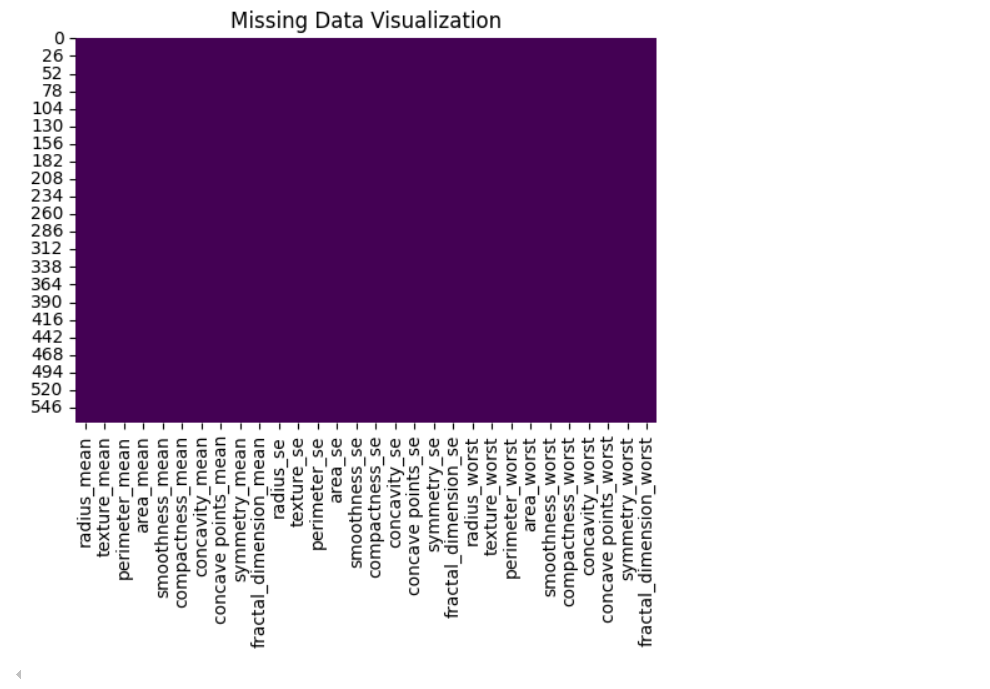
sns.heatmap(data.isnull(), cmap='viridis', cbar=False)

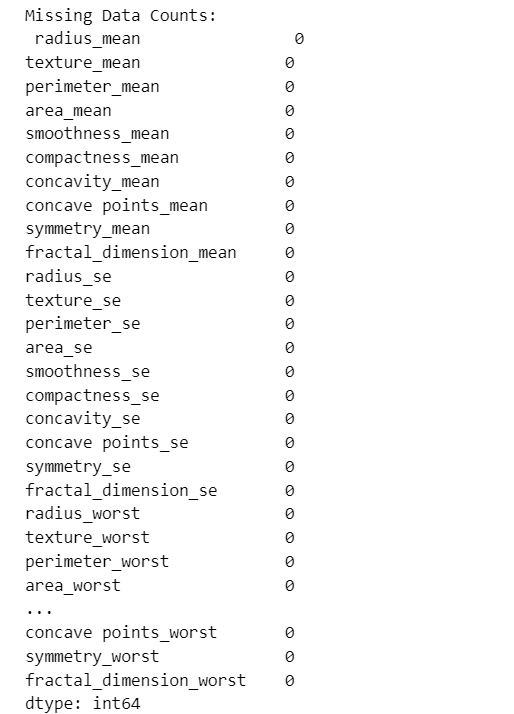
plt.title('Missing Data Visualization')

plt.show()

# In số lượng dữ liệu bị thiếu

print("Missing Data Counts:\n", missing\_data)

****



Giải thích output:

Biểu đồ (phần trên):

Trục hoành (x-axis): là tên các cột (feature) trong dataset như radius\_mean, texture\_mean, perimeter\_mean, v.v.

Trục tung (y-axis): là chỉ số của các dòng (index) trong dataset.

Màu sắc: Màu tím đậm thể hiện không có giá trị bị thiếu trong các cột đó. Nếu có các ô màu trắng hoặc sáng hơn thì đó là nơi thiếu dữ liệu.

Vì toàn bộ biểu đồ có màu tím đồng nhất → dataset không có giá trị nào bị thiếu.

Bảng liệt kê phía dưới (Missing Data Counts):

Đây là bảng thống kê số lượng giá trị bị thiếu cho từng cột.

Tất cả các giá trị đều bằng 0 → xác nhận rằng không có dữ liệu bị thiếu trong bất kỳ cột nào.

Kết luận:

=> Dataset hoàn chỉnh, không có missing value, nên không cần xử lý giá trị bị thiếu trước khi xây dựng mô hình.

## **6. Kiểm tra hiển thị và phát hiện các hàng trùng lặp trong dataset**

Các hàng trùng lặp xảy ra khi toàn bộ giá trị trong một hàng giống hệt một hàng khác. Việc giữ lại các bản ghi trùng lặp như vậy sẽ gây ra sai lệch trong việc huấn luyện mô hình, đặc biệt là khi mô hình học máy đánh giá mức độ xuất hiện của từng quan sát.

**Hậu quả của các hàng trùng lặp có thể bao gồm:**

* Làm lệch phân phối thống kê của dữ liệu.
* Khiến mô hình bị "thiên lệch" học quá mức vào các quan sát lặp lại.
* Gây lãng phí tài nguyên tính toán không cần thiết.

Do đó, việc **phát hiện và loại bỏ các hàng trùng lặp** là bước cần thiết để đảm bảo độ sạch và độ chính xác của dữ liệu đầu vào.

Việc phát hiện các hàng trùng lặp thường bao gồm:

* Kiểm tra xem có bao nhiêu hàng trùng lặp hoàn toàn.
* Xem xét cụ thể các hàng đó để đảm bảo chúng không có sự khác biệt tiềm ẩn do lỗi nhập liệu (ví dụ: khác biệt ký tự trắng, định dạng khác nhau).
* Xác định xem có cần giữ lại bất kỳ bản ghi trùng lặp nào không (trong một số trường hợp hiếm hoi, việc trùng lặp có thể là có chủ đích).
* Nếu không cần thiết, tiến hành loại bỏ hoàn toàn các hàng trùng để giữ cho dữ liệu sạch sẽ và gọn gàng hơn.

Việc làm sạch dữ liệu, cụ thể là loại bỏ các hàng trùng lặp, đóng vai trò quan trọng trong việc cải thiện chất lượng mô hình. Một tập dữ liệu được xử lý kỹ lưỡng sẽ giúp mô hình:

* Dễ dàng học được các đặc trưng thực sự có ý nghĩa trong dữ liệu.
* Giảm rủi ro overfitting do học từ các mẫu lặp.
* Tăng độ chính xác và khả năng tổng quát hóa khi dự đoán dữ liệu mới.

data.duplicated() tạo ra một Series Boolean: True với mỗi dòng trùng lặp (tức là đã xuất hiện trước đó), và False với dòng đầu tiên xuất hiện.

data[data.duplicated()] trả về toàn bộ các dòng trùng lặp (ngoại trừ dòng đầu tiên của mỗi bản sao).

Kiểm tra xem duplicate\_rows có rỗng không:

Nếu có dữ liệu trùng: in ra các hàng trùng.

Nếu không có: in ra thông báo "No duplicated rows found.".

Mục đích đoạn mã: kiểm tra và báo cáo số lượng (và nội dung) các dòng bị trùng lặp trong DataFrame.

Đây là một bước quan trọng trong quá trình tiền xử lý dữ liệu, giúp đảm bảo dữ liệu đầu vào sạch và không gây thiên lệch cho mô hình.

# Xác định dữ liệu trùng lặp

duplicate\_rows = data[data.duplicated()]

# Hiển thị các hàng trùng lặp nếu chúng tồn tại

if not duplicate\_rows.empty:

    print("\nDuplicated Rows:")

    print(duplicate\_rows)

else:

    print("\nNo duplicated rows found.")

# In số hàng trùng lặp

print(f"\nNumber of Duplicated Rows: {len(duplicate\_rows)}")



 Giải thích chi tiết:

data.drop\_duplicates(): Xóa tất cả các dòng trùng lặp trong DataFrame data.

Mặc định chỉ giữ lại bản đầu tiên của mỗi dòng trùng (nếu có).

Nếu trước đó dữ liệu không có bản sao nào (như bạn đã kiểm tra bằng data.duplicated()), thì thao tác này không thay đổi gì.

.reset\_index(drop=True): Sau khi xóa dòng trùng, chỉ số (index) của các dòng trong DataFrame có thể không còn liên tiếp (ví dụ: 0, 1, 3, 4,...).

Lệnh này đặt lại chỉ mục về dãy số liên tục từ 0, và drop=True nghĩa là không giữ lại chỉ mục cũ làm cột mới.

print(data.shape): In ra kích thước của DataFrame sau khi xử lý.

Trả về một tuple: (số dòng, số cột).

# Loại bỏ các bản sao và đặt lại chỉ mục

data = data.drop\_duplicates().reset\_index(drop=True)

print(data.shape)

ouput:



# Chương III. Phân tách và lựa chọn đặc trưng của dữ liệu

## **1. Phân tách dữ liệu thành dữ liệu số và dữ liệu phân loại**

### **1.1. Mục đích của bước phân tách dữ liệu**

Trong bất kỳ bài toán phân tích dữ liệu hay xây dựng mô hình học máy nào, việc hiểu rõ bản chất và kiểu dữ liệu là bước khởi đầu thiết yếu. Dữ liệu trong một tập dữ liệu có thể bao gồm nhiều loại: từ các con số định lượng (ví dụ: tuổi, chiều cao, cân nặng, nồng độ hormone,...) cho đến các giá trị định tính hoặc phân loại (ví dụ: giới tính, chẩn đoán bệnh lành tính hay ác tính, tên loại mô, trạng thái,...).

Việc phân biệt rõ giữa **dữ liệu số (numerical data)** và **dữ liệu phân loại (categorical data)** là cực kỳ quan trọng vì:

* **Các kỹ thuật tiền xử lý khác nhau** được áp dụng cho từng loại dữ liệu. Ví dụ, dữ liệu số thường cần được chuẩn hóa (standardization) hoặc chuẩn tỷ lệ (normalization), trong khi dữ liệu phân loại cần được mã hóa (encoding) thành dạng số.
* **Một số thuật toán máy học chỉ chấp nhận dữ liệu dạng số**, do đó, ta cần xác định dữ liệu nào cần chuyển đổi trước khi huấn luyện mô hình.
* Giúp nhà phân tích hiểu được bản chất của từng đặc trưng (feature) để đưa ra quyết định xử lý phù hợp, tránh rủi ro gây sai lệch kết quả mô hình.

### **1.2. Định nghĩa hai loại dữ liệu**

Dữ liệu số (Numerical Data)

Là những dữ liệu có thể biểu diễn bằng các con số và mang ý nghĩa định lượng. Dữ liệu số chia làm hai loại chính:

* **Dữ liệu rời rạc (discrete data)**: như số lần xuất hiện, số lượng khối u, số lượng tế bào, v.v.
* **Dữ liệu liên tục (continuous data)**: như kích thước, tỷ lệ, chỉ số đo lường, nồng độ, v.v.

Những đặc trưng dạng số thường được dùng trực tiếp trong các mô hình học máy như hồi quy tuyến tính, cây quyết định, mạng nơ-ron,…

Dữ liệu phân loại (Categorical Data)

Là những dữ liệu biểu diễn các thuộc tính hoặc nhãn định tính, không mang ý nghĩa toán học trực tiếp. Ví dụ: “Giới tính” có thể gồm “Nam” và “Nữ”; “Chẩn đoán” có thể là “Benign” (lành tính) hoặc “Malignant” (ác tính).

Dữ liệu phân loại có thể chia thành:

* **Nominal (danh nghĩa)**: không có thứ tự rõ ràng, ví dụ: màu sắc, quốc tịch.
* **Ordinal (có thứ tự)**: có sự sắp xếp theo mức độ, ví dụ: mức độ nguy cơ “thấp”, “vừa”, “cao”.

### **1.3. Ý nghĩa và lợi ích của bước phân tách**

Việc phân tách dữ liệu thành hai nhóm rõ ràng này mang lại nhiều lợi ích:

* Hỗ trợ tiền xử lý hiệu quả

Khi đã biết được cột nào là số, cột nào là phân loại, ta có thể:

* Chuẩn hóa dữ liệu số (ví dụ: chuyển về khoảng [0,1] hoặc chuẩn hóa theo z-score).
* Mã hóa dữ liệu phân loại (ví dụ: dùng One-hot encoding hoặc Label encoding).
* Tránh lỗi trong quá trình huấn luyện mô hình

Nếu mô hình nhận dữ liệu phân loại ở dạng văn bản (chữ) thay vì số, nó có thể báo lỗi hoặc xử lý sai. Việc tách biệt giúp đảm bảo rằng mỗi loại dữ liệu được xử lý đúng cách trước khi đưa vào mô hình.

* Phân tích thống kê chính xác hơn

Khi làm thống kê mô tả, ta thường tính trung bình, phương sai, v.v. trên dữ liệu số, trong khi dữ liệu phân loại lại dùng đếm tần suất hoặc vẽ biểu đồ phân phối. Việc phân nhóm giúp chọn đúng phương pháp thống kê phù hợp cho từng loại dữ liệu.

### **1.4. Ví dụ trong bài toán “ Breast Cancer “**

# Chọn dữ liệu số

numeric\_data = data.select\_dtypes(include='number').columns

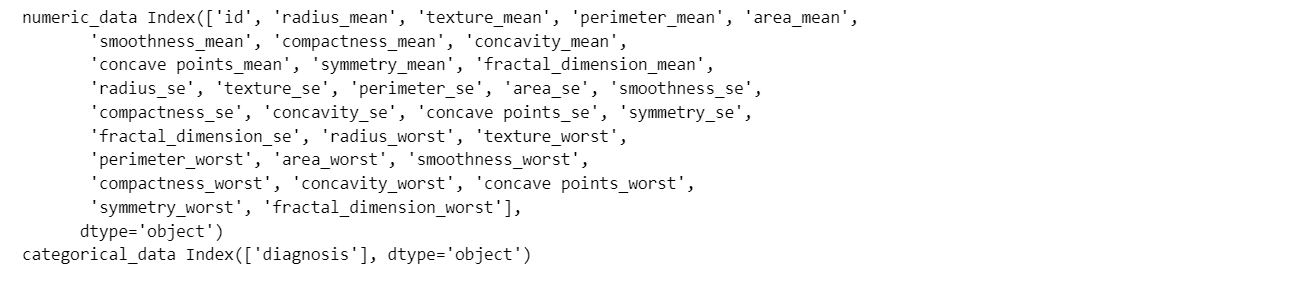
print("numeric\_data", numeric\_data)

# Chọn dữ liệu phân loại

categorical\_data = data.select\_dtypes(exclude='number').columns

print("categorical\_data", categorical\_data)

**output:**

****

for col in categorical\_data:

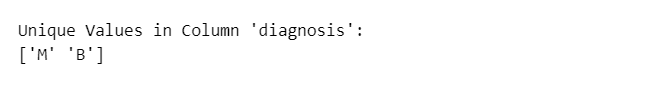
    unique\_values = data[col].unique()

    # In các giá trị duy nhất cho cột hiện tại

    print(f"\nUnique Values in Column '{col}':")

    print(unique\_values)

    # Tùy chọn, in số lượng giá trị nếu cần



Trong bộ dữ liệu "Breast Cancer" (Ung thư vú), các đặc trưng có thể chia thành:

* **Dữ liệu số**: như "radius\_mean" (bán kính trung bình), "texture\_mean", "area\_se", "compactness\_worst", v.v. Đây là các chỉ số đo lường đặc trưng về khối u.
* **Dữ liệu phân loại**: như "diagnosis" (kết quả chẩn đoán: B = lành tính, M = ác tính), "id" (mã bệnh nhân), v.v.

Phân tách rõ ràng sẽ giúp bạn:

* Chuẩn hóa các đặc trưng số để mô hình không bị ảnh hưởng bởi sự khác biệt đơn vị.
* Mã hóa nhãn "diagnosis" thành dạng số (ví dụ: B = 0, M = 1) để phục vụ cho mô hình phân loại nhị phân.

### **1.5. Tổng kết**

Bước phân loại dữ liệu thành số và phân loại không chỉ là một thao tác kỹ thuật mà còn thể hiện sự hiểu biết sâu sắc về cấu trúc của dữ liệu. Nó là nền tảng để thực hiện các bước tiền xử lý một cách chính xác, đảm bảo cho mô hình học máy hoạt động hiệu quả, tránh lỗi sai và đạt được kết quả dự đoán tốt hơn.

Nếu bạn đang xây dựng một pipeline cho bài toán học máy, bước này là không thể thiếu và nên được thực hiện ngay sau khi tải dữ liệu để đảm bảo mọi bước xử lý sau đều được định hướng đúng

**2. Trực quan hóa từng cột dữ liệu số trong bộ dữ liệu Breast Cancer**

**-** Trực quan hóa dữ liệu là một bước quan trọng trong quy trình phân tích dữ liệu, giúp chúng ta hiểu rõ hơn về cấu trúc, phân bố và mối quan hệ của các biến trong tập dữ liệu. Trong trường hợp bộ dữ liệu **Breast Cancer**, có rất nhiều cột đặc trưng là các đặc tính số đo được trích xuất từ ảnh quét tế bào vú. Những cột dữ liệu số này có thể chia thành ba nhóm: giá trị trung bình, độ lệch chuẩn, và giá trị cực đại.

for col in numeric\_data:

    fig, axes = plt.subplots(2, 1, figsize=(10, 5))  # 2 rows, 1 column

    # Biểu đồ phân phối (Biểu đồ con trên cùng)

    sns.histplot(data=data, x=col, kde=True, ax=axes[0])

    axes[0].set\_title(f"Distribution of '{col}'")

    axes[0].set\_xlabel(col)

    axes[0].set\_ylabel("Density")

   # Box Plot (Cốt truyện dưới cùng)

    sns.boxplot(data=data, x=col, ax=axes[1])

    axes[1].set\_title(f"Boxplot of '{col}'")

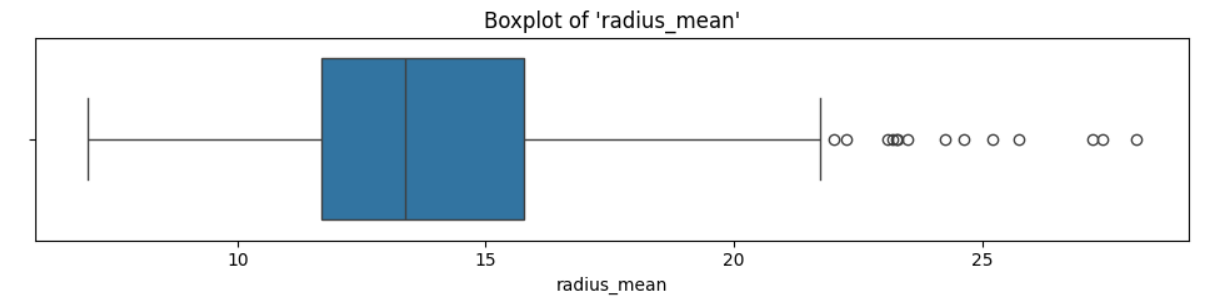
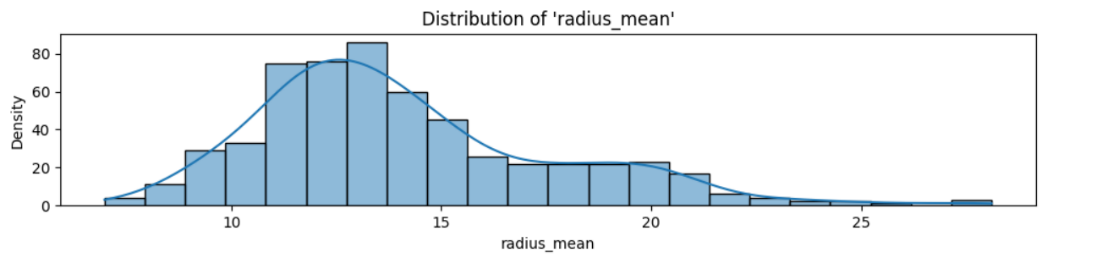
    axes[1].set\_xlabel(col)

    # Điều chỉnh bố cục và hiển thị

    plt.tight\_layout()

    plt.show()

**Biểu đồ tần suất (Histogram) và hộp số (Boxplot) của radius\_mean**

******Biểu đồ tần suất (Histogram)**

Biểu đồ tần suất cho thấy phân phối của biến radius\_mean có xu hướng **lệch phải nhẹ**. Phần lớn các giá trị tập trung trong khoảng **11 đến 15**, với **đỉnh tần suất** rơi vào khoảng giá trị **13**. Điều này phản ánh rằng phần lớn các quan sát có kích thước bán kính trung bình ở mức vừa phải, và các giá trị lớn hơn (từ 20 trở lên) là ít gặp hơn.

**Hộp số (Boxplot)**

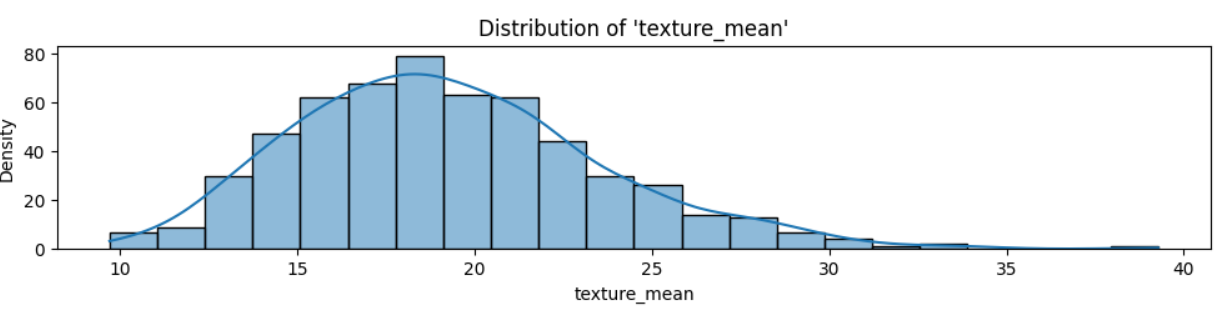
Boxplot cho thấy:

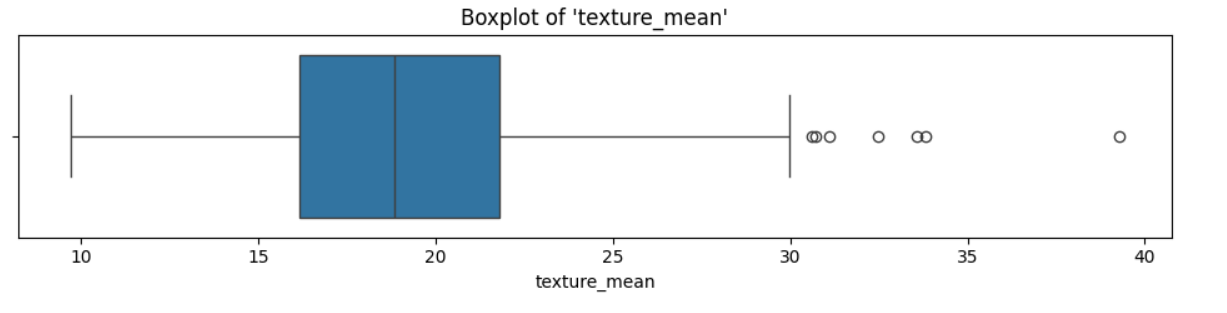
* **Giá trị trung vị** (median) của radius\_mean nằm khoảng **13.3**.
* **Khoảng tứ phân vị (IQR)** rơi vào khoảng **từ 11 đến 15.5**, thể hiện sự tập trung dữ liệu trong phạm vi hẹp.
* Có sự xuất hiện của nhiều **giá trị ngoại lai (outliers)** phía bên phải (trên), đại diện cho những mẫu có bán kính trung bình lớn bất thường.

**Nhận định**

Phân tích biểu đồ cho thấy biến radius\_mean có **phân phối lệch phải nhẹ**, với đa số giá trị tập trung quanh mức trung bình (13). Đồng thời, có một số lượng đáng kể **giá trị ngoại lai ở phía trên**, phản ánh sự **đa dạng trong kích thước bán kính trung bình** giữa các mẫu. Điều này có thể liên quan đến **đặc điểm sinh học khác nhau** của các mẫu hoặc **giai đoạn bệnh lý** khác nhau nếu đây là dữ liệu y học (ví dụ: dữ liệu ung thư vú).

**Biểu đồ phân phối và hộp số của texture\_mean**



****

**Biểu đồ tần suất (Histogram)**

Biểu đồ tần suất cho thấy phân phối của biến texture\_mean **có dạng gần đối xứng**, nhưng hơi **lệch phải nhẹ**. Phần lớn các mẫu tập trung trong khoảng từ **15 đến 23**, với **đỉnh tần suất** rơi vào khoảng **18 đến 19**, thể hiện rằng đây là vùng giá trị phổ biến nhất của biến này trong tập dữ liệu.

**Hộp số (Boxplot)**

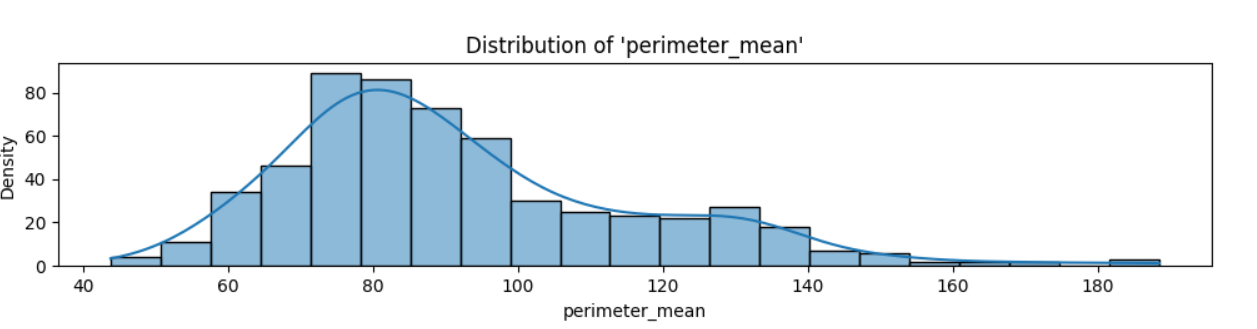
Boxplot cho thấy:

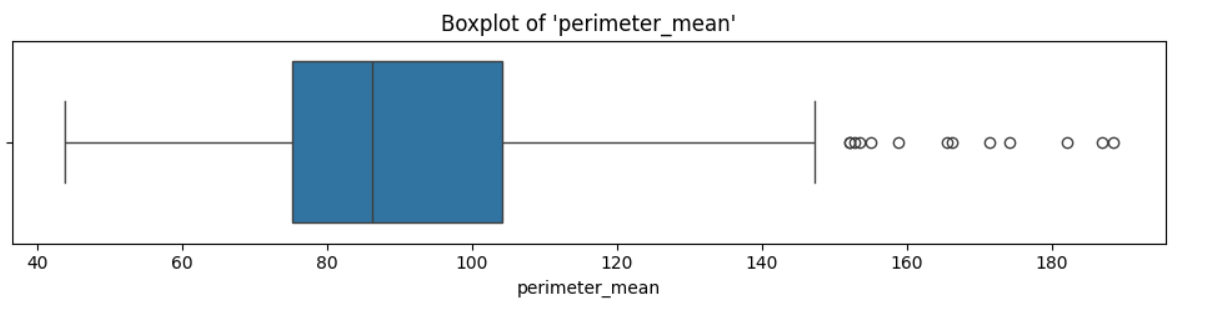
* **Giá trị trung vị** (median) của texture\_mean nằm khoảng **18.5**.
* **Khoảng tứ phân vị (IQR)** rơi vào khoảng **16 đến 21.5**, tức là dữ liệu tập trung khá tốt trong một phạm vi tương đối hẹp.
* Xuất hiện một số **giá trị ngoại lai (outliers)** phía trên (trên 30), nhưng không quá nhiều.

**Nhận định**

Biểu đồ phân phối và hộp số cho thấy biến texture\_mean có **phân phối gần chuẩn**, với phần lớn dữ liệu tập trung quanh mức trung bình từ 16 đến 21. Sự xuất hiện của một vài giá trị ngoại lai lớn phản ánh **sự biến đổi trong đặc trưng về kết cấu** của các mẫu — điều này có thể liên quan đến tính chất sinh học khác nhau giữa các loại tế bào hoặc mức độ nghiêm trọng khác nhau của bệnh lý (nếu là dữ liệu y sinh).

**Biểu đồ phân phối và hộp số của perimeter\_mean**

****

****

**Biểu đồ tần suất (Histogram)**

iểu đồ tần suất cho thấy phân phối của perimeter\_mean **lệch phải rõ rệt**. Phần lớn giá trị tập trung trong khoảng từ **70 đến 100**, với đỉnh rơi vào khoảng **80 đến 85**, cho thấy đây là phạm vi phổ biến nhất của đặc trưng chu vi.

**Hộp số (Boxplot)**

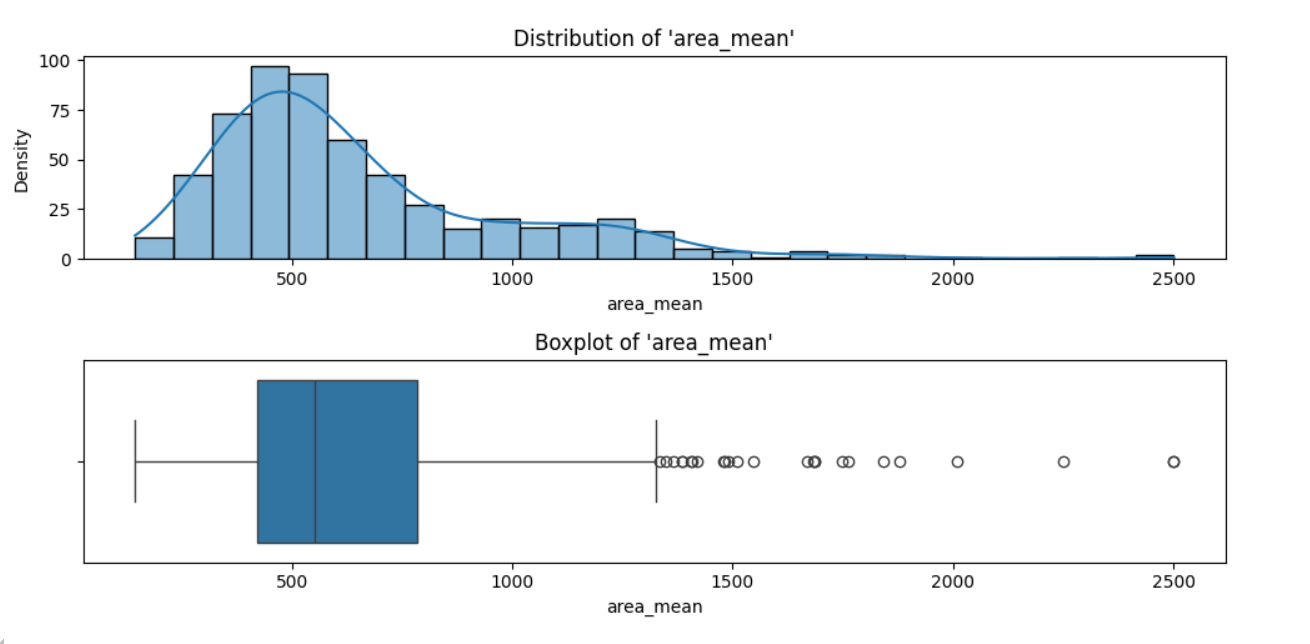
Boxplot cho thấy:

* **Trung vị (median)** rơi vào khoảng **85**.
* **Khoảng tứ phân vị (IQR)** kéo dài từ khoảng **75 đến 105**, cho thấy dữ liệu phân bố khá rộng trong phạm vi này.
* Có sự xuất hiện của **nhiều giá trị ngoại lai (outliers)** nằm trên 140, đặc biệt có các điểm vượt quá 180. Điều này cho thấy có một số mẫu với chu vi cao bất thường.

**Nhận định**

Biến perimeter\_mean có xu hướng phân phối **lệch phải**, cho thấy rằng có một số mẫu có chu vi cao hơn đáng kể so với phần lớn dữ liệu. Những giá trị ngoại lai có thể là các mẫu bất thường hoặc đại diện cho một nhóm đặc biệt (ví dụ: khối u lớn hơn, nếu là dữ liệu y sinh). Việc xử lý hoặc kiểm tra thêm các ngoại lệ này có thể giúp hiểu rõ hơn về bản chất của tập dữ liệu.

**Biểu đồ phân phối và hộp số của area\_mean**

****

**Biểu đồ tần suất (Histogram)**

Biểu đồ tần suất của area\_mean cho thấy phân phối **lệch phải rất rõ ràng**. Phần lớn các giá trị rơi vào khoảng từ **300 đến 800**, với đỉnh phân phối tập trung gần **500**. Tuy nhiên, biểu đồ có một "đuôi dài" kéo về bên phải, cho thấy một số giá trị rất lớn (trên 1000, thậm chí hơn 2000).

**Hộp số (Boxplot)**

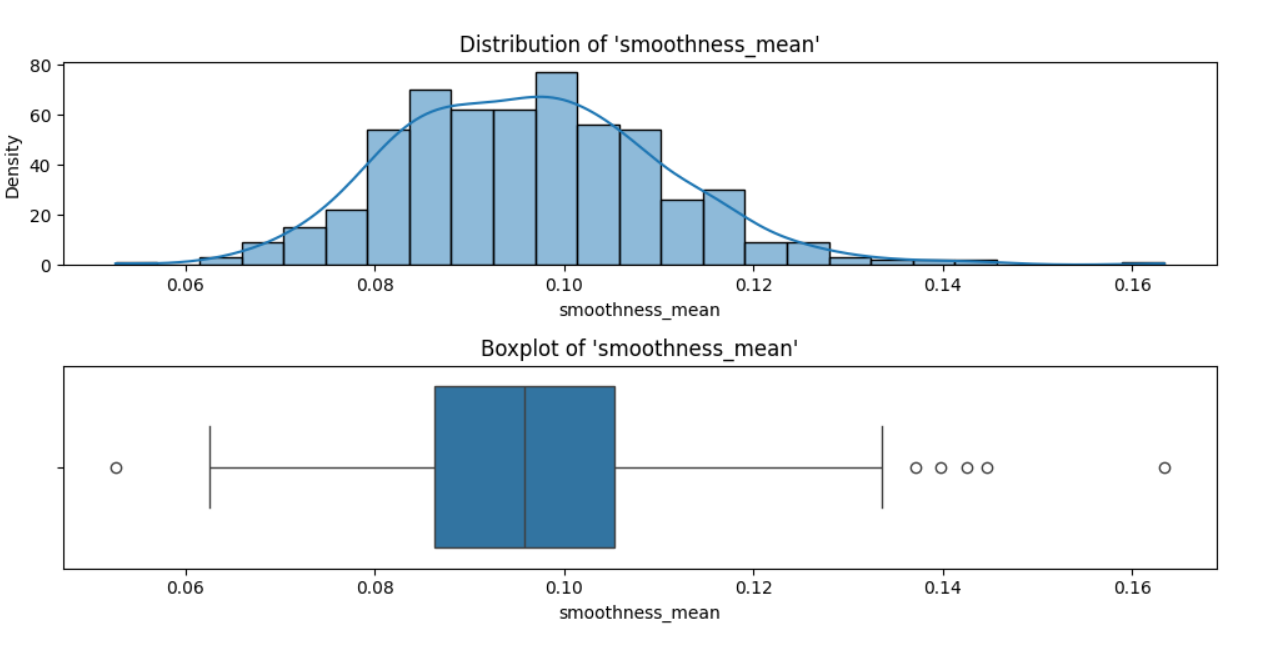
Boxplot cho thấy:

* **Trung vị** rơi vào khoảng **550**.
* **IQR (khoảng tứ phân vị)** nằm từ khoảng **400 đến 800**, là vùng chứa phần lớn dữ liệu.
* Có rất nhiều **giá trị ngoại lai (outliers)** nằm phía trên, một số vượt quá 2000. Điều này cho thấy một số mẫu có diện tích lớn vượt trội so với phần còn lại.

**Nhận định**

Biến area\_mean có phân phối **không đối xứng** và **lệch phải mạnh**, với nhiều giá trị ngoại lai. Đây là đặc điểm thường thấy trong dữ liệu thực tế, nơi chỉ một vài trường hợp đặc biệt có thể gây ra sai lệch lớn. Những mẫu có area\_mean rất cao có thể là dấu hiệu quan trọng (ví dụ như khối u lớn hơn bình thường nếu dữ liệu liên quan đến y sinh).

**Biểu đồ phân phối và hộp số của smoothness\_mean**

****

**Biểu đồ tần suất (Histogram)**

Biểu đồ tần suất của smoothness\_mean có dạng gần giống **phân phối chuẩn**, tập trung nhiều nhất quanh khoảng **0.09 đến 0.11**. Tuy nhiên, có một chút **lệch phải nhẹ**, thể hiện ở phần đuôi bên phải kéo dài hơn so với bên trái.

**Hộp số (Boxplot)**

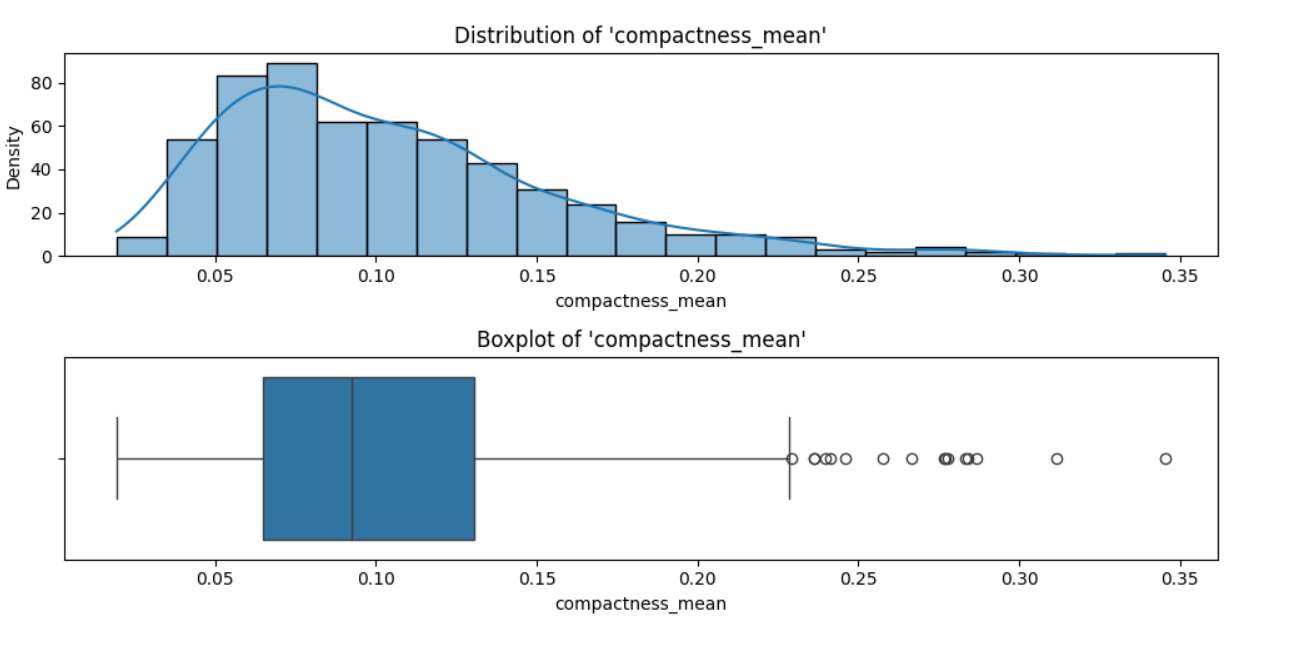
Boxplot cho thấy:

* Trung vị rơi vào khoảng **0.10**, gần đúng trung tâm của dữ liệu.
* Khoảng tứ phân vị (IQR) từ khoảng **0.085 đến 0.11**, là nơi tập trung phần lớn dữ liệu.
* Có **một vài giá trị ngoại lai** ở cả hai phía, nhưng tập trung nhiều hơn ở bên phải (cao hơn), trong đó có những điểm vượt quá **0.14**.

**Nhận định**

Biến smoothness\_mean có phân phối khá **cân đối**, gần với chuẩn, chỉ hơi lệch phải nhẹ. Có một số ngoại lai nhưng không quá nhiều và không quá xa so với phần còn lại của dữ liệu. Đây là một biến có phân phối tương đối ổn định và không cần biến đổi nhiều nếu sử dụng trong mô hình hóa.

**Biểu đồ phân phối và hộp số của compactness\_mean**

****

**Biểu đồ tần suất (Histogram)**

Biểu đồ tần suất của compactness\_mean có dạng **lệch phải rõ rệt**. Phần lớn giá trị tập trung trong khoảng **0.05 đến 0.12**, và số lượng giảm dần khi giá trị tăng lên đến khoảng **0.34**. Đây là dấu hiệu của một phân phối **không chuẩn**, với đuôi kéo dài về bên phải.

**Hộp số (Boxplot)**

Boxplot cho thấy:

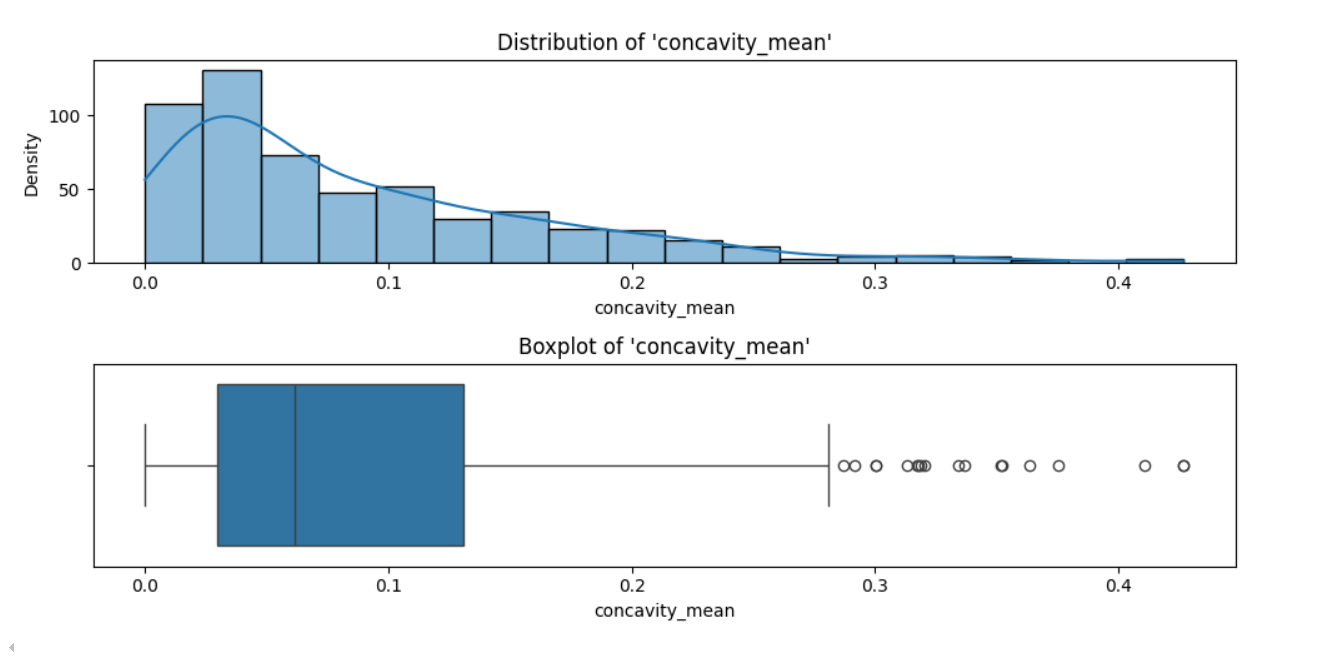
* Trung vị nằm gần khoảng **0.10**, đúng vùng có mật độ dữ liệu cao nhất.
* Phần lớn dữ liệu nằm trong khoảng **0.05 đến 0.15**.
* Có khá nhiều **giá trị ngoại lai** ở bên phải (trên), bao gồm cả những điểm vượt quá **0.30**, chứng tỏ tồn tại một số trường hợp bất thường với mức độ compactness\_mean cao hơn nhiều so với phần còn lại.

**Nhận định**

Biến compactness\_mean có phân phối **lệch phải mạnh**, với nhiều ngoại lai phía giá trị cao. Nếu sử dụng trong mô hình tuyến tính hoặc hồi quy, nên xem xét:

* **Biến đổi dữ liệu** (ví dụ: log hoặc Box-Cox transform).
* **Chuẩn hóa hoặc cắt ngoại lai** nếu cần thiết, tùy mục đích phân tích.

**Biểu đồ phân phối và hộp số của concavity\_mean**

****

**Biểu đồ tần suất (Histogram)**

Phân phối của concavity\_mean có dạng **lệch phải rất mạnh**. Rất nhiều giá trị tập trung gần **0.0–0.05**, sau đó mật độ giảm dần khi giá trị tăng lên đến khoảng **0.42**. Điều này cho thấy rằng **đa phần mẫu có độ lõm trung bình thấp**, nhưng vẫn tồn tại một số mẫu với độ lõm lớn hơn.

**Hộp số (Boxplot)**

 **Trung vị** của biến nằm gần **0.07**, chứng tỏ đa số mẫu có giá trị thấp.

 Khoảng tứ phân vị (IQR) khá hẹp, từ khoảng **0.03 đến 0.13**.

 Có **rất nhiều giá trị ngoại lai** ở phía bên phải (tức là giá trị lớn hơn), trong đó có một số vượt quá **0.35**.

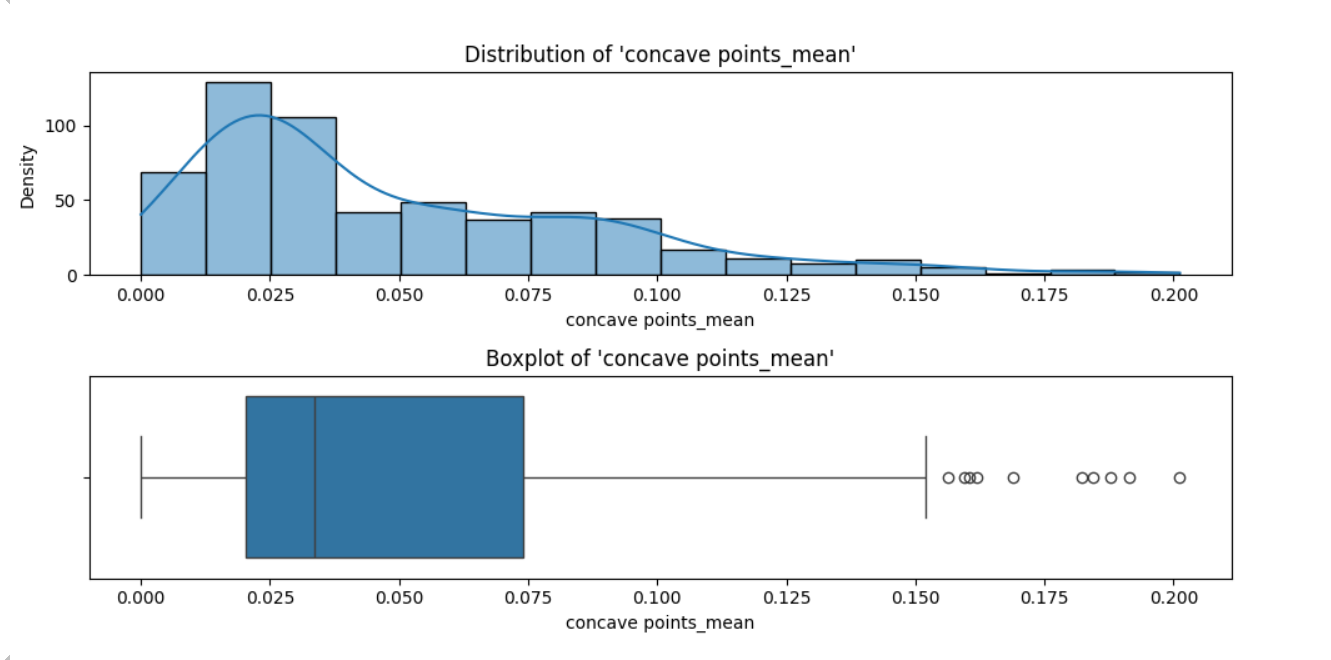
 Không có ngoại lai phía trái (gần 0).

**Nhận định**

Biến concavity\_mean có phân phối **rất lệch phải** và chứa **nhiều ngoại lai** ở phía giá trị lớn. Đây là một dấu hiệu mạnh cho thấy:

* **Không nên giả định phân phối chuẩn** khi làm thống kê suy diễn hoặc mô hình tuyến tính.
* Cần xem xét việc **biến đổi dữ liệu** (như log-transform) hoặc **loại bỏ/kiểm soát ngoại lai** tùy mục tiêu phân tích.

**Biểu đồ phân phối và hộp số của concave points\_mean**



**Biểu đồ tần suất (Histogram)**

Biểu đồ cho thấy phân phối của concave points\_mean có dạng **lệch phải rõ rệt**:

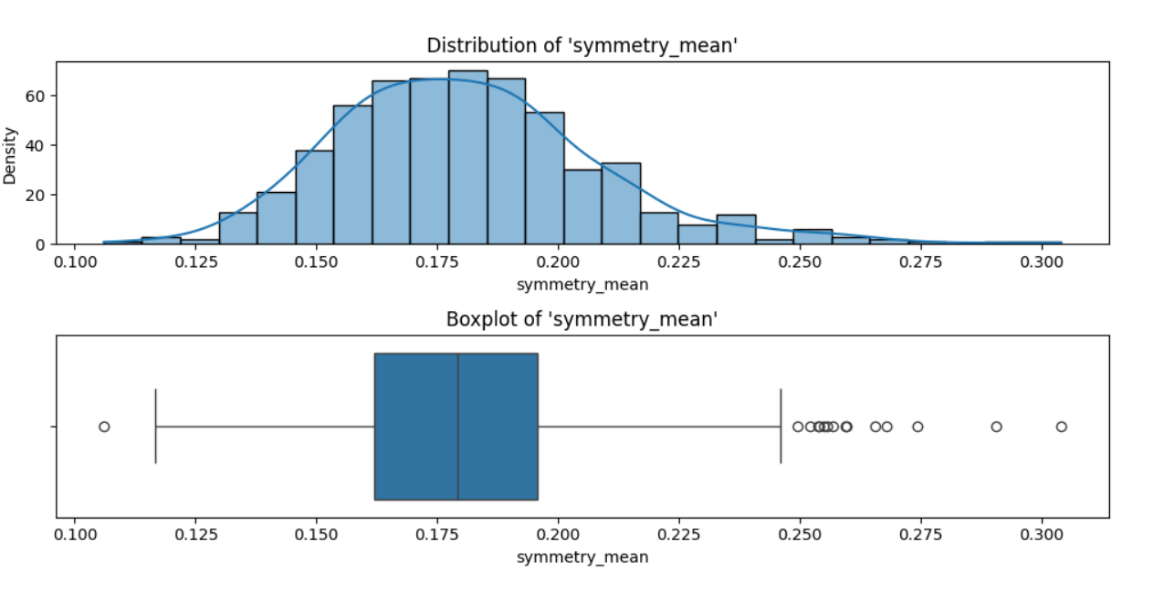
* Rất nhiều mẫu có giá trị gần **0.02–0.03**, với mật độ cao nhất ở khoảng này.
* Phân phối kéo dài về phía bên phải đến khoảng **0.2**, cho thấy sự tồn tại của các giá trị lớn hơn hiếm gặp.

**Hộp số (Boxplot)**

* **Trung vị** nằm vào khoảng **0.04**.
* Đa số dữ liệu nằm trong khoảng từ **0.02 đến 0.08** (IQR).
* Xuất hiện **một số lượng đáng kể giá trị ngoại lai** phía trên, bắt đầu từ khoảng **0.15** trở đi.
* Không có ngoại lai ở phía nhỏ hơn, do giới hạn dưới là **0** (giá trị không thể âm).

**Nhận định**

* Biến concave points\_mean cũng có **phân phối lệch phải**, giống như các biến hình dạng đã phân tích trước (compactness\_mean, concavity\_mean).
* Có thể cần thực hiện **chuẩn hóa hoặc biến đổi dữ liệu** nếu sử dụng cho mô hình hóa.
* Các ngoại lai nên được kiểm tra kỹ để đảm bảo chúng không ảnh hưởng xấu đến kết quả phân tích hoặc học máy.

**Biểu đồ phân phối và hộp số của symmetry\_mean**

**Biểu đồ tần suất (Histogram) của 'symmetry\_mean':** Phân tích dữ liệu cho thấy biểu đồ tần suất của 'symmetry\_mean' có dạng phân phối **tương đối đối xứng**, gần với **phân phối chuẩn**. Dữ liệu tập trung chủ yếu trong khoảng giá trị từ **0.15 đến 0.20**, với đỉnh tần suất xuất hiện ở mức xấp xỉ **0.175**. Điều này cho thấy đa số các giá trị của 'symmetry\_mean' cụm lại quanh giá trị trung tâm này.

**Hộp số (Boxplot) của 'symmetry\_mean':** Biểu đồ hộp số chỉ ra rằng giá trị **trung vị** của 'symmetry\_mean' nằm ở mức khoảng **0.18**. Phần lớn các giá trị tập trung trong khoảng từ **0.16 đến 0.20** (thể hiện qua hộp chữ nhật - IQR). Có một số **giá trị ngoại lai** ở cả hai phía của phân phối, đặc biệt là các giá trị lớn hơn 0.25 và nhỏ hơn 0.12. Độ rộng của hộp (IQR) tương đối nhỏ, cho thấy sự tập trung của dữ liệu.

**Nhận định:** Phân tích tổng thể cho thấy 'symmetry\_mean' có xu hướng **tập trung quanh một giá trị trung tâm** và phân phối tương đối đối xứng. Sự hiện diện của các giá trị ngoại lai ở cả hai phía cho thấy có một số trường hợp cá biệt với giá trị 'symmetry\_mean' cao hoặc thấp hơn đáng kể so với phần lớn dữ liệu.

**Biểu đồ phân phối và hộp số của fractal\_dimension\_mean**

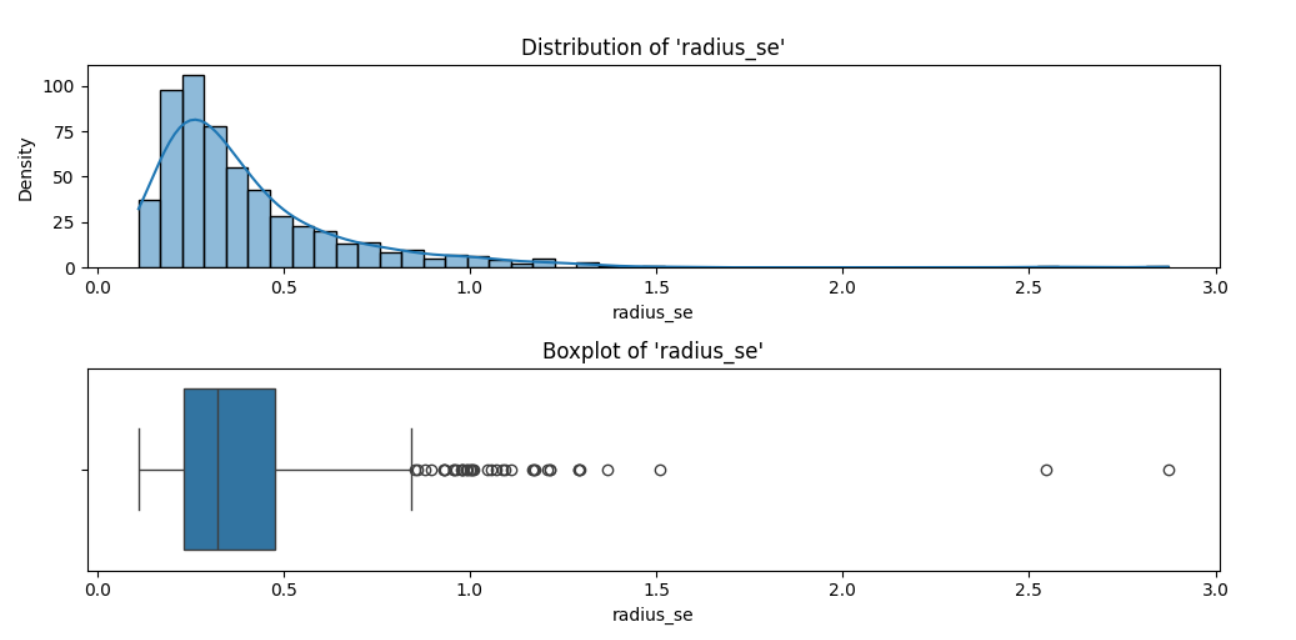


**Biểu đồ tần suất (Histogram) của 'fractal\_dimension\_mean':** Phân tích dữ liệu cho thấy biểu đồ tần suất của 'fractal\_dimension\_mean' có dạng phân phối **lệch phải**. Phần lớn các giá trị tập trung trong khoảng từ **0.055 đến 0.070**, với đỉnh tần suất xuất hiện ở mức xấp xỉ **0.060 - 0.062**. Tần suất giảm dần khi giá trị 'fractal\_dimension\_mean' tăng lên, cho thấy có ít hơn các quan sát có giá trị cao.

**Hộp số (Boxplot) của 'fractal\_dimension\_mean':** Biểu đồ hộp số chỉ ra rằng giá trị **trung vị** của 'fractal\_dimension\_mean' nằm ở mức khoảng **0.062**. Phần lớn các giá trị (IQR - khoảng tứ phân vị) tập trung trong khoảng từ khoảng **0.058 đến 0.066**. Có một số lượng đáng kể **giá trị ngoại lai** ở phía trên (giá trị lớn), bắt đầu từ khoảng 0.078 trở đi. Điều này khẳng định tính chất lệch phải của phân phối đã quan sát được trong histogram. Râu dưới của boxplot kéo dài đến khoảng 0.05, cho thấy sự hiện diện của các giá trị thấp hơn nhưng không có nhiều giá trị ngoại lai ở phía dưới.

**Nhận định:** Phân tích tổng thể cho thấy 'fractal\_dimension\_mean' có xu hướng **tập trung ở các giá trị thấp** và phân phối bị **lệch về phía các giá trị cao hơn**. Sự hiện diện của nhiều giá trị ngoại lai ở phía trên cho thấy có một số trường hợp cá biệt với giá trị 'fractal\_dimension\_mean' cao hơn đáng kể so với phần lớn dữ liệu. Điều này có thể gợi ý sự đa dạng hoặc các yếu tố đặc biệt ảnh hưởng đến chiều fractal ở một số ít quan sát

**Biểu đồ phân phối và hộp số của radius\_se**

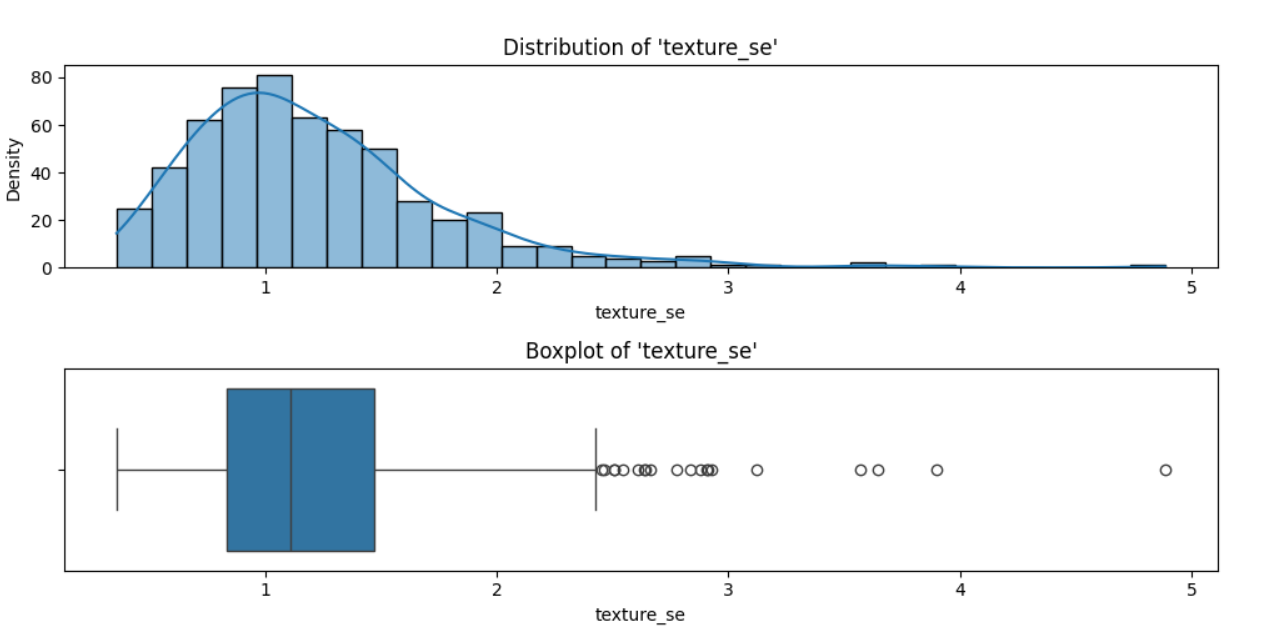


**Biểu đồ tần suất (Histogram) của 'radius\_se':** Phân tích dữ liệu cho thấy biểu đồ tần suất của 'radius\_se' có dạng phân phối **lệch phải rất mạnh**. Phần lớn các giá trị tập trung ở mức rất thấp, chủ yếu trong khoảng từ **0.1 đến 0.5**, với đỉnh tần suất cao nhất xuất hiện ở khoảng **0.2 đến 0.3**. Tần suất giảm nhanh chóng khi giá trị 'radius\_se' tăng lên, cho thấy có rất ít quan sát có giá trị 'radius\_se' lớn.

**Hộp số (Boxplot) của 'radius\_se':** Biểu đồ hộp số xác nhận tính chất lệch phải mạnh của phân phối. Giá trị **trung vị** của 'radius\_se' nằm ở mức rất thấp, khoảng **0.3**. Hộp chữ nhật (IQR - khoảng tứ phân vị) cũng rất hẹp và nằm ở phía giá trị thấp, trong khoảng từ khoảng **0.2 đến 0.45**. Có một số lượng rất lớn các **giá trị ngoại lai** ở phía trên, kéo dài từ khoảng 0.75 cho đến gần 3.0. Điều này cho thấy phần lớn dữ liệu có sai số chuẩn của bán kính (radius standard error) rất nhỏ, nhưng có một số trường hợp cá biệt với sai số này lớn hơn đáng kể.

**Nhận định:** Phân tích tổng thể cho thấy 'radius\_se' có xu hướng **tập trung rất mạnh ở các giá trị thấp**. Phân phối cực kỳ lệch phải với nhiều giá trị ngoại lai ở phía trên, cho thấy rằng trong hầu hết các trường hợp, sai số chuẩn của bán kính là nhỏ, nhưng có một số ít trường hợp ngoại lệ với sai số lớn hơn nhiều. Điều này có thể hàm ý rằng phép đo bán kính thường có độ chính xác cao, nhưng có những yếu tố không thường xuyên gây ra sự biến động lớn hơn trong sai số.

**Biểu đồ phân phối và hộp số của texture\_se**

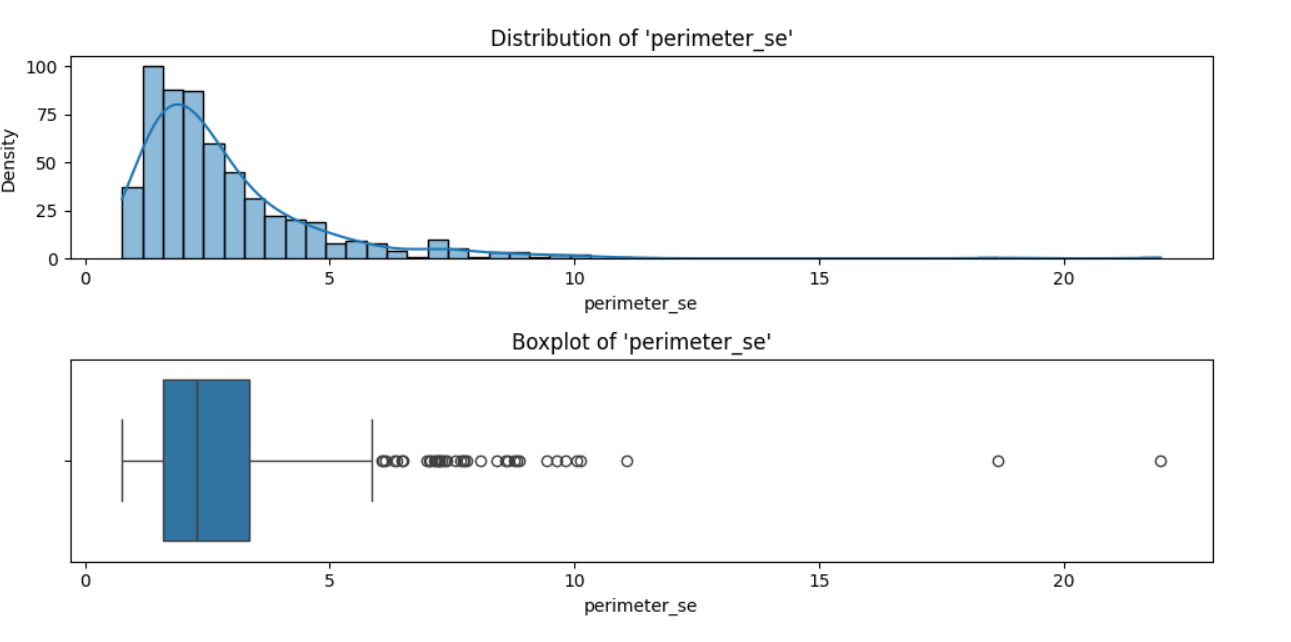
****

**Biểu đồ tần suất (Histogram) của 'texture\_se':** Phân tích dữ liệu cho thấy biểu đồ tần suất của 'texture\_se' có dạng phân phối **lệch phải**. Phần lớn các giá trị tập trung trong khoảng từ **0.5 đến 1.5**, với đỉnh tần suất xuất hiện ở mức xấp xỉ **1.0**. Tần suất giảm dần khi giá trị 'texture\_se' tăng lên, cho thấy có ít quan sát hơn có giá trị sai số chuẩn của texture (texture standard error) cao.

**Hộp số (Boxplot) của 'texture\_se':** Biểu đồ hộp số cũng cho thấy tính chất lệch phải của phân phối. Giá trị **trung vị** của 'texture\_se' nằm ở mức khoảng **1.1**. Hộp chữ nhật (IQR - khoảng tứ phân vị) thể hiện sự tập trung của phần lớn dữ liệu trong khoảng từ khoảng **0.8 đến 1.5**. Có một số lượng đáng kể các **giá trị ngoại lai** ở phía trên (giá trị lớn), bắt đầu từ khoảng 2.3 trở đi, và một giá trị ngoại lai đặc biệt lớn gần 5.0. Điều này củng cố nhận định về sự phân bố lệch của dữ liệu.

**Nhận định:** Phân tích tổng thể cho thấy 'texture\_se' có xu hướng **tập trung ở các giá trị tương đối thấp** và phân phối bị **lệch về phía các giá trị cao hơn**. Sự hiện diện của nhiều giá trị ngoại lai ở phía trên chỉ ra rằng mặc dù hầu hết các quan sát có sai số chuẩn của texture ở mức vừa phải, một số trường hợp lại có sai số này cao hơn đáng kể. Điều này có thể gợi ý về sự biến động không đồng đều trong đặc điểm texture qua các mẫu.

**Biểu đồ phân phối và hộp số của perimeter\_se**

****

**Biểu đồ tần suất (Histogram):**

* Phân phối của 'perimeter\_se' cho thấy sự phân bố lệch phải rõ rệt.
* Phần lớn các giá trị tập trung ở khoảng giá trị nhỏ, đặc biệt là gần 0.
* Đuôi của phân phối kéo dài về phía bên phải, cho thấy có một số lượng nhỏ các mẫu có giá trị 'perimeter\_se' lớn hơn đáng kể.
* Có một đỉnh rõ rệt ở gần giá trị rất nhỏ, cho thấy sự tập trung cao của dữ liệu ở vùng này.

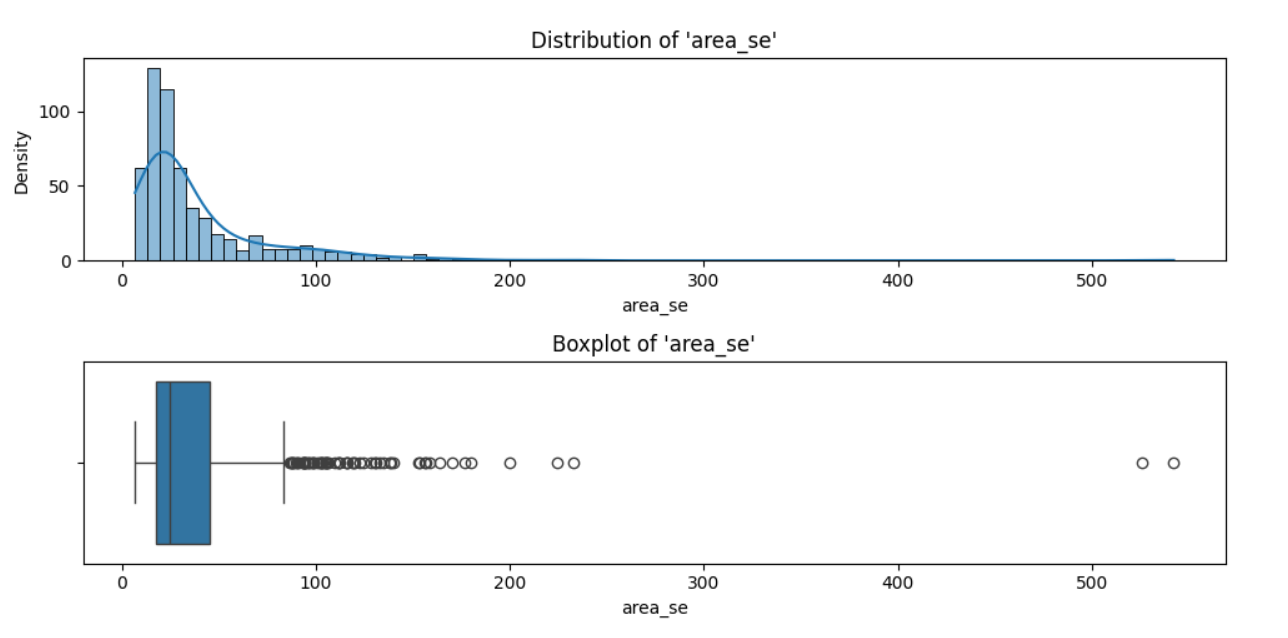
**Hộp số (Boxplot):**

* Hộp cho thấy phần lớn dữ liệu nằm trong khoảng từ khoảng 1 đến gần 4.
* Đường trung tuyến (đường kẻ ngang trong hộp) có vẻ nằm gần phía dưới của hộp, điều này cũng gợi ý về sự lệch phải của dữ liệu.
* Các "râu" (whiskers) kéo dài ra ngoài hộp, với râu bên phải dài hơn nhiều so với râu bên trái, củng cố thêm nhận định về sự lệch phải.
* Có nhiều điểm dữ liệu nằm ngoài các "râu" ở phía bên phải, được biểu thị là các dấu chấm tròn. Đây là các giá trị ngoại lai (outliers) tiềm năng, cho thấy có một số mẫu có giá trị 'perimeter\_se' rất lớn so với phần còn lại.

**Nhận định chung:**

* Dữ liệu của 'perimeter\_se' có xu hướng tập trung ở các giá trị thấp và có sự lệch phải rõ ràng.
* Sự xuất hiện của nhiều giá trị ngoại lai ở phía trên cho thấy có một số ít các mẫu có chu vi trung bình lớn hơn đáng kể so với đa số còn lại.
* Điều này cho thấy sự biến động đáng kể trong 'perimeter\_se' giữa các mẫu.

**Biểu đồ phân phối và hộp số của area\_se**



**Biểu đồ tần suất (Histogram):**

* Phân phối của 'area\_se' cho thấy sự phân bố lệch phải rất rõ rệt.
* Phần lớn các giá trị tập trung ở khoảng giá trị rất nhỏ, gần 0.
* Đuôi của phân phối kéo dài đáng kể về phía bên phải, cho thấy có một số lượng nhỏ các mẫu có giá trị 'area\_se' lớn hơn nhiều so với phần còn lại.
* Đỉnh của phân phối nằm ở vùng giá trị thấp.

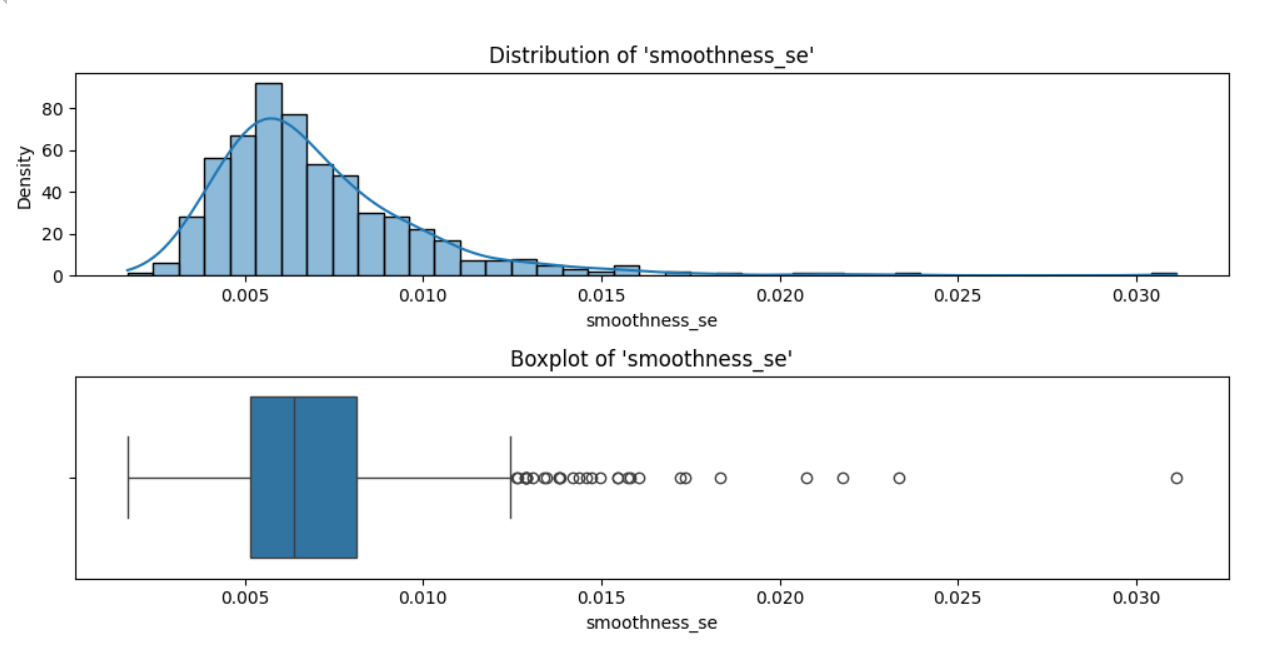
**Hộp số (Boxplot):**

* Hộp cho thấy phần lớn dữ liệu nằm trong khoảng từ rất gần 0 đến khoảng 40-50.
* Đường trung tuyến (đường kẻ ngang trong hộp) nằm gần phía dưới của hộp, một lần nữa cho thấy sự lệch phải.
* "Râu" bên phải kéo dài hơn nhiều so với "râu" bên trái.
* Có rất nhiều điểm dữ liệu nằm ngoài "râu" ở phía bên phải, biểu thị các giá trị ngoại lai. Các giá trị ngoại lai này trải dài từ khoảng 100 đến hơn 500, cho thấy sự khác biệt lớn so với phần lớn dữ liệu.

**Nhận định chung:**

* Dữ liệu của 'area\_se' có xu hướng tập trung mạnh mẽ ở các giá trị thấp và có độ lệch phải rất lớn.
* Sự xuất hiện của một lượng lớn các giá trị ngoại lai ở phía bên phải cho thấy có một số mẫu có diện tích trung bình lớn hơn đáng kể so với phần lớn các mẫu khác.
* Sự khác biệt lớn giữa phần lớn dữ liệu và các giá trị ngoại lai này cho thấy sự biến động lớn trong 'area\_se' giữa các quan sát.

**Biểu đồ phân phối và hộp số của smoothness\_se**

****

**Biểu đồ tần suất (Histogram):**

* Phân phối của 'smoothness\_se' cho thấy sự phân bố lệch phải.
* Phần lớn các giá trị tập trung ở khoảng giá trị thấp, với một đỉnh rõ rệt ở khoảng 0.006 - 0.007.
* Đuôi của phân phối kéo dài về phía bên phải, cho thấy có một số lượng nhỏ các mẫu có giá trị 'smoothness\_se' lớn hơn.

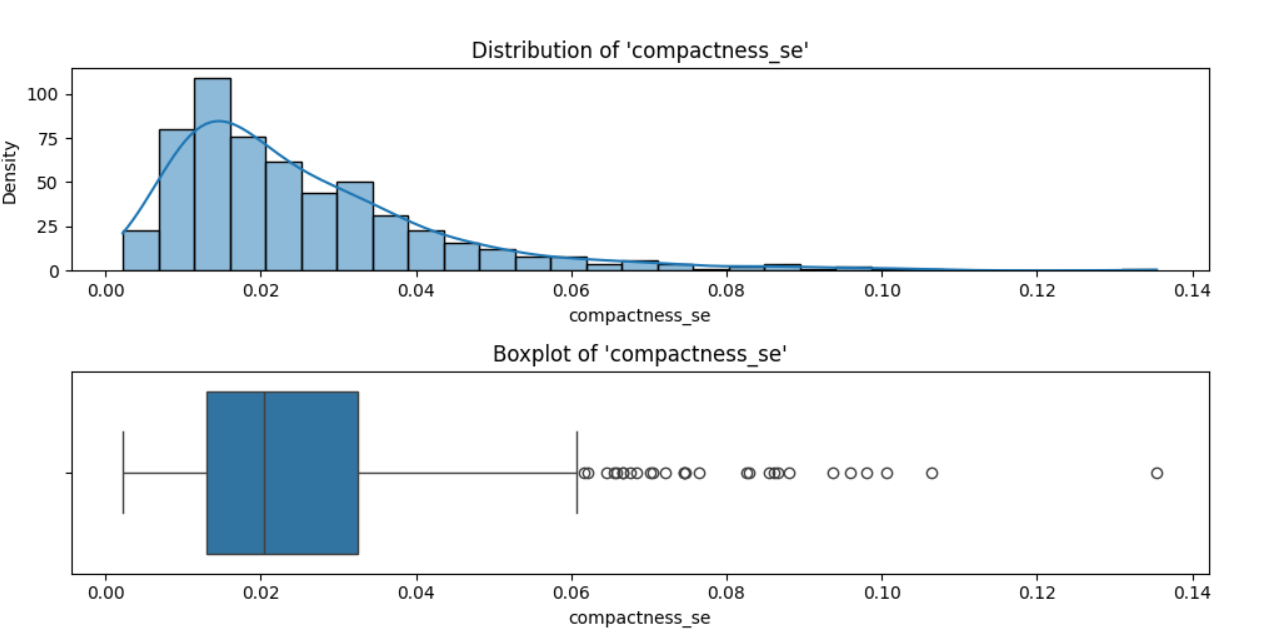
**Hộp số (Boxplot):**

* Hộp cho thấy phần lớn dữ liệu nằm trong khoảng từ khoảng 0.004 đến khoảng 0.008.
* Đường trung tuyến (đường kẻ ngang trong hộp) nằm gần giữa hộp.
* "Râu" bên phải dài hơn "râu" bên trái, cho thấy sự lệch nhẹ về phía bên phải.
* Có một số điểm dữ liệu nằm ngoài "râu" ở phía bên phải, biểu thị các giá trị ngoại lai tiềm năng.

**Nhận định chung:**

* Dữ liệu của 'smoothness\_se' có xu hướng tập trung ở các giá trị thấp và có sự lệch phải nhẹ.
* Sự xuất hiện của các giá trị ngoại lai ở phía trên cho thấy có một số mẫu có độ mịn (smoothness) lớn hơn so với phần lớn các mẫu khác.

**Biểu đồ phân phối và hộp số của compactness\_se**

****

**Biểu đồ tần suất (Histogram):**

* Phân phối của 'compactness\_se' cho thấy sự phân bố lệch phải rõ rệt.
* Phần lớn các giá trị tập trung ở khoảng giá trị thấp, với một đỉnh rõ rệt ở khoảng 0.01 - 0.02.
* Đuôi của phân phối kéo dài về phía bên phải, cho thấy có một số lượng nhỏ các mẫu có giá trị 'compactness\_se' lớn hơn đáng kể.

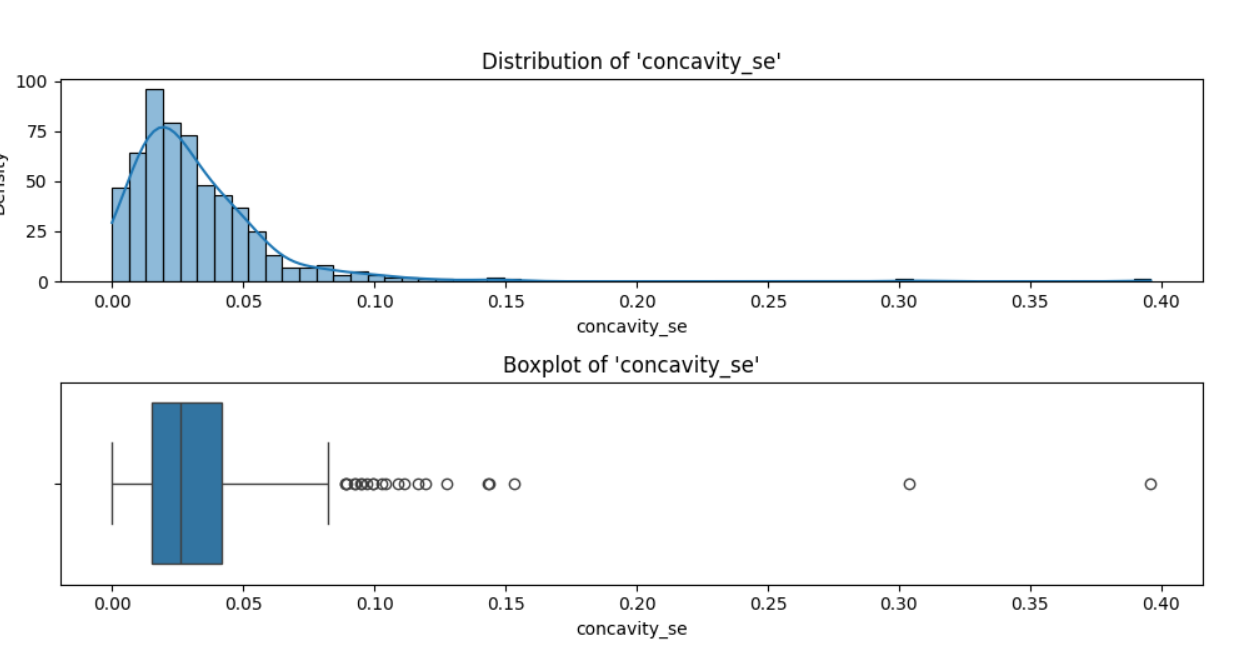
**Hộp số (Boxplot):**

* Hộp cho thấy phần lớn dữ liệu nằm trong khoảng từ khoảng 0.005 đến khoảng 0.03.
* Đường trung tuyến (đường kẻ ngang trong hộp) nằm gần phía dưới của hộp, cho thấy sự lệch phải.
* "Râu" bên phải dài hơn nhiều so với "râu" bên trái.
* Có nhiều điểm dữ liệu nằm ngoài "râu" ở phía bên phải, biểu thị các giá trị ngoại lai.

**Nhận định chung:**

* Dữ liệu của 'compactness\_se' có xu hướng tập trung ở các giá trị thấp và có độ lệch phải rõ ràng.
* Sự xuất hiện của nhiều giá trị ngoại lai ở phía trên cho thấy có một số mẫu có độ nén (compactness) lớn hơn đáng kể so với phần lớn các mẫu khác.

**Biểu đồ phân phối và hộp số của concavity\_se**

****

**Biểu đồ tần suất (Histogram):**

* Phân phối của 'concavity\_se' cho thấy sự phân bố lệch phải rất rõ rệt.
* Phần lớn các giá trị tập trung ở khoảng giá trị rất thấp, gần 0, với một đỉnh rõ rệt ở khoảng 0.02.
* Đuôi của phân phối kéo dài đáng kể về phía bên phải, cho thấy có một số lượng nhỏ các mẫu có giá trị 'concavity\_se' lớn hơn nhiều so với phần còn lại.

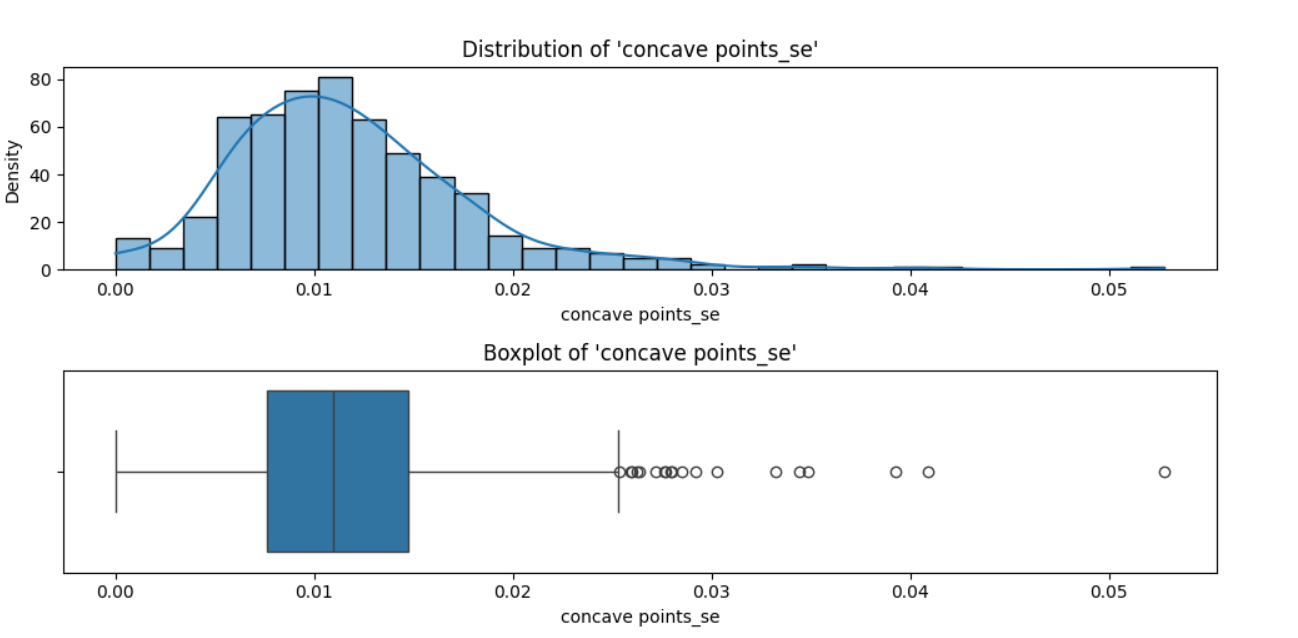
**Hộp số (Boxplot):**

* Hộp cho thấy phần lớn dữ liệu nằm trong khoảng từ rất gần 0 đến khoảng 0.04.
* Đường trung tuyến (đường kẻ ngang trong hộp) nằm gần phía dưới của hộp, cho thấy sự lệch phải.
* "Râu" bên phải kéo dài hơn nhiều so với "râu" bên trái.
* Có rất nhiều điểm dữ liệu nằm ngoài "râu" ở phía bên phải, biểu thị các giá trị ngoại lai. Các giá trị ngoại lai này trải dài đến gần 0.4, cho thấy sự khác biệt lớn so với phần lớn dữ liệu.

**Nhận định chung:**

* Dữ liệu của 'concavity\_se' có xu hướng tập trung mạnh mẽ ở các giá trị thấp và có độ lệch phải rất lớn.
* Sự xuất hiện của một lượng lớn các giá trị ngoại lai ở phía bên phải cho thấy có một số mẫu có độ lõm (concavity) lớn hơn đáng kể so với phần lớn các mẫu khác.
* Sự khác biệt lớn giữa phần lớn dữ liệu và các giá trị ngoại lai này cho thấy sự biến động lớn trong 'concavity\_se' giữa các quan sát.

**Biểu đồ phân phối và hộp số của concave points\_se**

****

**Biểu đồ tần suất (Histogram):**

* Phân phối của 'concave points\_se' cho thấy sự phân bố lệch phải.
* Phần lớn các giá trị tập trung ở khoảng giá trị thấp, với một đỉnh rõ rệt ở khoảng 0.01 - 0.015.
* Đuôi của phân phối kéo dài về phía bên phải, cho thấy có một số lượng nhỏ các mẫu có giá trị 'concave points\_se' lớn hơn.

**Hộp số (Boxplot):**

* Hộp cho thấy phần lớn dữ liệu nằm trong khoảng từ khoảng 0 đến khoảng 0.02.
* Đường trung tuyến (đường kẻ ngang trong hộp) nằm gần giữa hộp hoặc hơi lệch về phía dưới.
* "Râu" bên phải dài hơn "râu" bên trái, cho thấy sự lệch nhẹ về phía bên phải.
* Có một số điểm dữ liệu nằm ngoài "râu" ở phía bên phải, biểu thị các giá trị ngoại lai tiềm năng.

**Nhận định chung:**

* Dữ liệu của 'concave points\_se' có xu hướng tập trung ở các giá trị thấp và có sự lệch phải nhẹ.
* Sự xuất hiện của các giá trị ngoại lai ở phía trên cho thấy có một số mẫu có số lượng điểm lõm lớn hơn so với phần lớn các mẫu khác.

**Biểu đồ phân phối và hộp số của symmetry\_se**

****

**Biểu đồ tần suất (Histogram):**

* Phân phối của 'symmetry\_se' cho thấy sự phân bố lệch phải.
* Phần lớn các giá trị tập trung ở khoảng giá trị thấp, với một đỉnh rõ rệt ở khoảng 0.015 - 0.02.
* Đuôi của phân phối kéo dài về phía bên phải, cho thấy có một số lượng nhỏ các mẫu có giá trị 'symmetry\_se' lớn hơn.

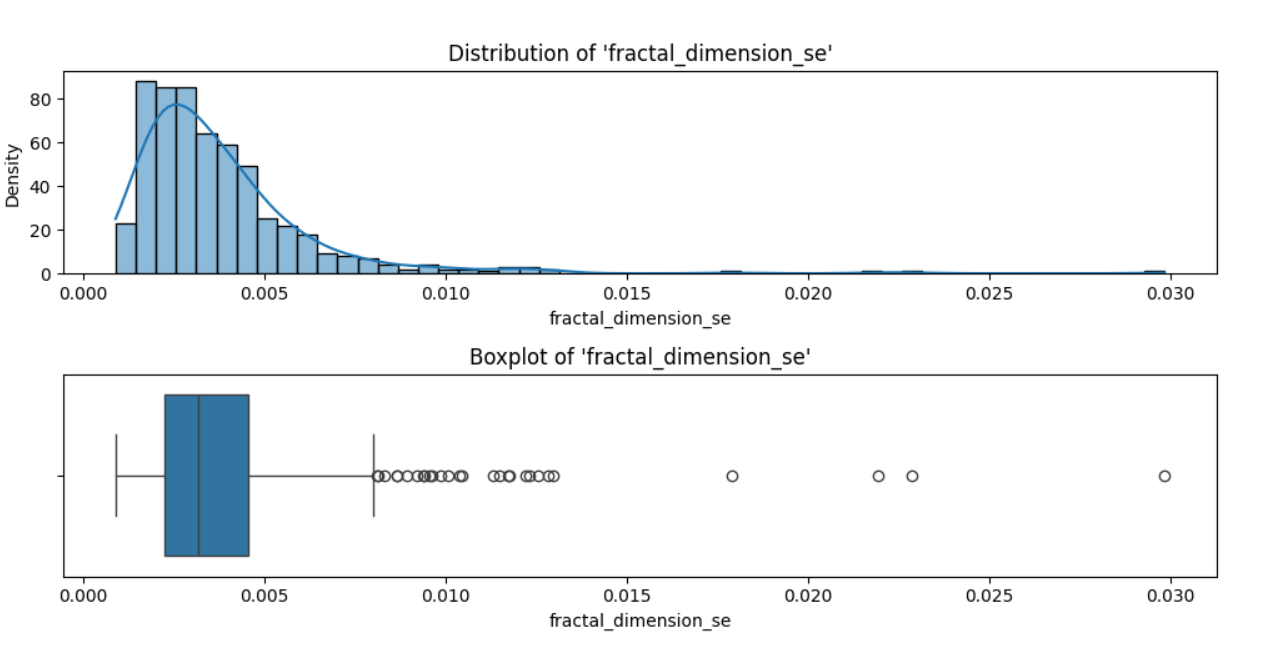
**Hộp số (Boxplot):**

* Hộp cho thấy phần lớn dữ liệu nằm trong khoảng từ khoảng 0.012 đến khoảng 0.025.
* Đường trung tuyến (đường kẻ ngang trong hộp) nằm gần giữa hộp.
* "Râu" bên phải dài hơn "râu" bên trái, cho thấy sự lệch nhẹ về phía bên phải.
* Có một số điểm dữ liệu nằm ngoài "râu" ở phía bên phải, biểu thị các giá trị ngoại lai tiềm năng.

**Nhận định chung:**

* Dữ liệu của 'symmetry\_se' có xu hướng tập trung ở các giá trị thấp và có sự lệch phải nhẹ.
* Sự xuất hiện của các giá trị ngoại lai ở phía trên cho thấy có một số mẫu có độ đối xứng lớn hơn so với phần lớn các mẫu khác.

**Biểu đồ phân phối và hộp số của fractal\_dimension\_se**



**Biểu đồ tần suất (Histogram):**

* Phân phối của 'fractal\_dimension\_se' cho thấy sự phân bố lệch phải rõ rệt.
* Phần lớn các giá trị tập trung ở khoảng giá trị thấp, với một đỉnh rõ rệt ở khoảng 0.002 - 0.004.
* Đuôi của phân phối kéo dài về phía bên phải, cho thấy có một số lượng nhỏ các mẫu có giá trị 'fractal\_dimension\_se' lớn hơn đáng kể.

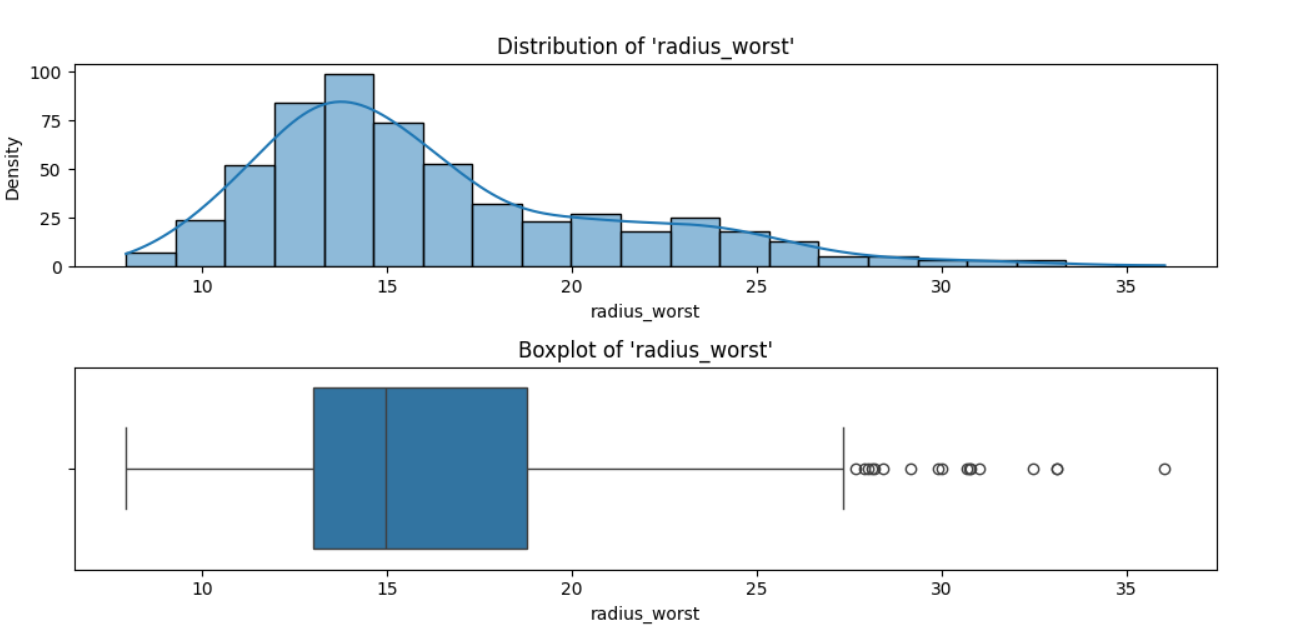
**Hộp số (Boxplot):**

* Hộp cho thấy phần lớn dữ liệu nằm trong khoảng từ khoảng 0.0015 đến khoảng 0.005.
* Đường trung tuyến (đường kẻ ngang trong hộp) nằm gần phía dưới của hộp, cho thấy sự lệch phải.
* "Râu" bên phải dài hơn nhiều so với "râu" bên trái.
* Có nhiều điểm dữ liệu nằm ngoài "râu" ở phía bên phải, biểu thị các giá trị ngoại lai.

**Nhận định chung:**

* Dữ liệu của 'fractal\_dimension\_se' có xu hướng tập trung ở các giá trị thấp và có độ lệch phải rõ ràng.
* Sự xuất hiện của nhiều giá trị ngoại lai ở phía trên cho thấy có một số mẫu có chiều fractal lớn hơn đáng kể so với phần lớn các mẫu khác.

**Biểu đồ phân phối và hộp số của radius\_worst**

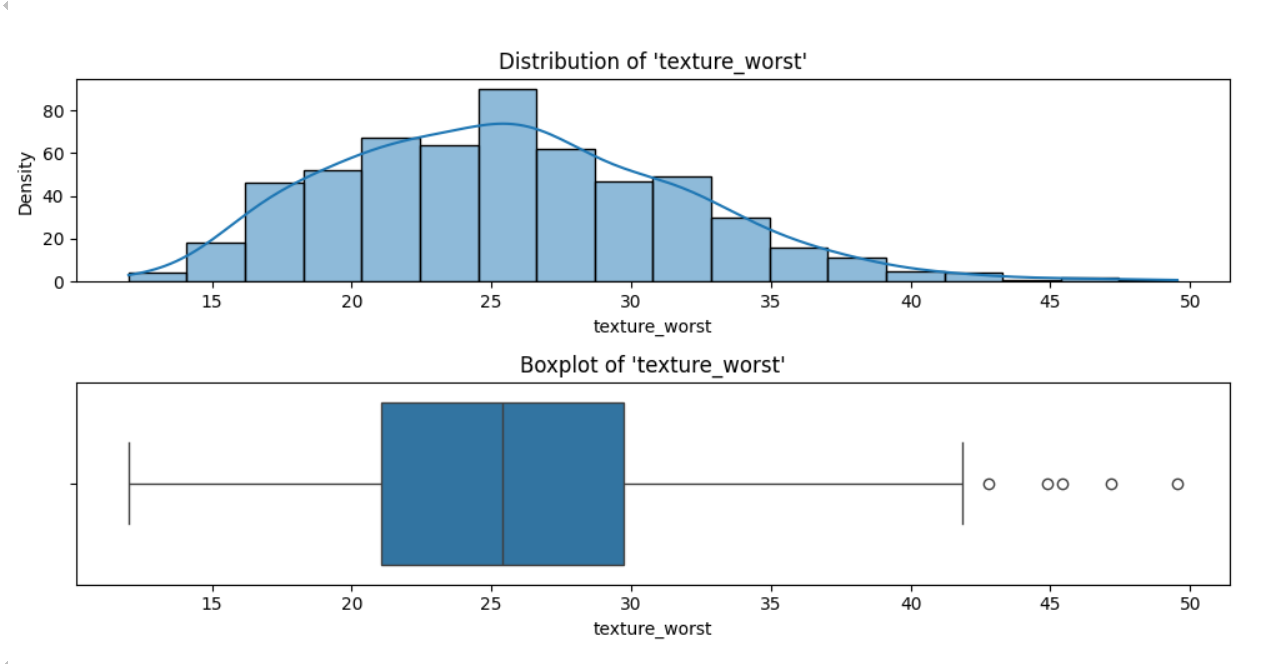
****

**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của radius\_worst cho thấy một phân phối lệch phải rõ rệt. Phần lớn các giá trị tập trung trong khoảng từ khoảng 10 đến 20, với đỉnh tần suất xuất hiện ở mức xấp xỉ 13-14. Điều này cho thấy đa số các trường hợp có giá trị radius\_worst tương đối thấp đến trung bình, nhưng cũng có một số trường hợp với giá trị cao hơn đáng kể, kéo dài phần đuôi của phân phối về phía bên phải.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của radius\_worst nằm ở mức khoảng 14.5. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 12 (Q1) đến 18.5 (Q3). Điều này cho thấy sự tập trung tương đối của dữ liệu trong phạm vi này. Các "râu" (whiskers) của biểu đồ kéo dài từ khoảng 7.5 đến 27.5, cho thấy biên độ của phần lớn dữ liệu không chứa các điểm ngoại lai. Tuy nhiên, có nhiều điểm ngoại lai (outliers) được biểu thị bằng các chấm tròn ở phía trên, bắt đầu từ khoảng giá trị 28 trở lên, củng cố cho nhận định về phân phối lệch phải từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng radius\_worst có xu hướng lệch phải, với phần lớn dữ liệu tập trung ở các giá trị thấp hơn và trung bình. Sự hiện diện của nhiều giá trị ngoại lai ở phía trên cho thấy có một số trường hợp sở hữu giá trị radius\_worst lớn hơn đáng kể so với đa số. Trong bối cảnh phân tích dữ liệu y học (ví dụ, kích thước khối u), điều này có thể phản ánh sự đa dạng về kích thước của các khối u "tồi tệ nhất" (worst), với đa số có kích thước nhỏ đến trung bình nhưng một số trường hợp lại có kích thước rất lớn, cần được chú ý đặc biệt.

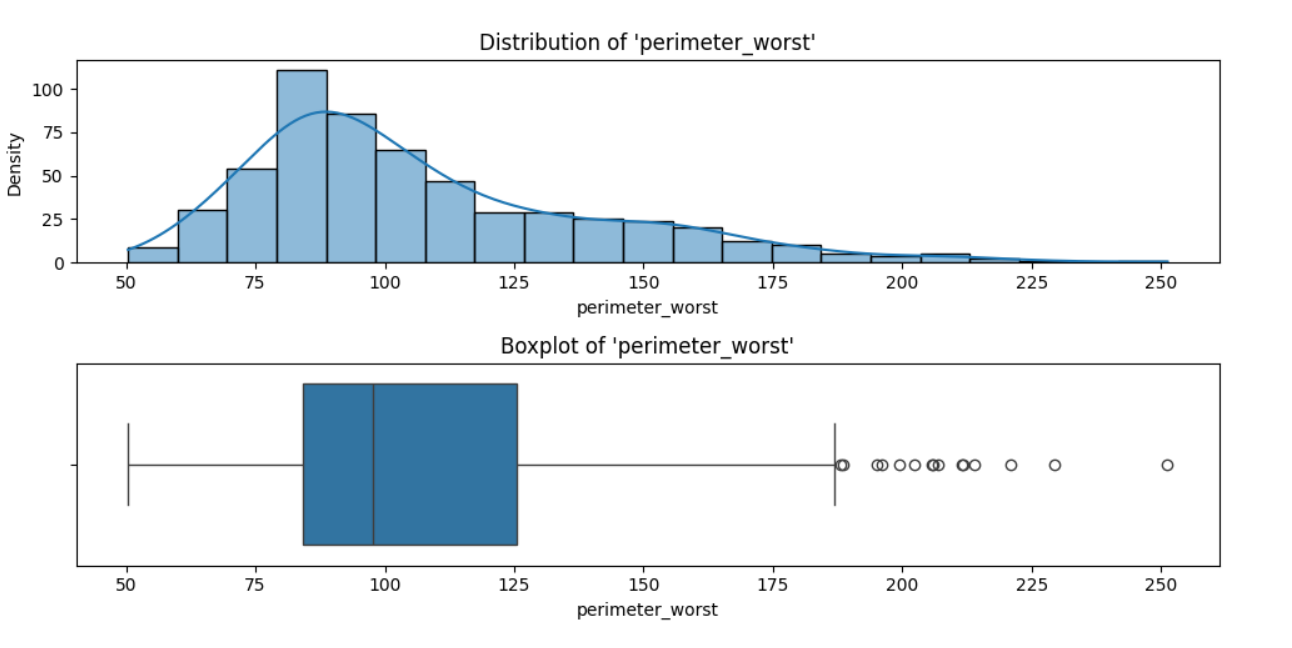
**Biểu đồ phân phối và hộp số của texture\_worst**

****

**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của texture\_worst có dạng gần đối xứng, với một chút xu hướng lệch nhẹ về phía phải. Phần lớn các giá trị tập trung trong khoảng từ khoảng 20 đến 35, với đỉnh tần suất xuất hiện ở mức xấp xỉ 25-26. Điều này cho thấy đa số các trường hợp có giá trị texture\_worst tập trung quanh một giá trị trung tâm, nhưng cũng có một số ít trường hợp có giá trị cao hơn, kéo dài phần đuôi của phân phối.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của texture\_worst nằm ở mức khoảng 25. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 21 (Q1) đến 30 (Q3). Độ rộng của khoảng tứ phân vị (IQR) cho thấy một sự phân tán vừa phải của dữ liệu. Các "râu" (whiskers) của biểu đồ kéo dài từ khoảng 12.5 đến khoảng 42. Có một số điểm ngoại lai (outliers) được biểu thị bằng các chấm tròn ở phía trên, bắt đầu từ khoảng giá trị 43 trở lên, củng cố cho nhận định về xu hướng lệch phải nhẹ từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng texture\_worst có phân phối gần đối xứng, với một xu hướng lệch nhẹ về phía phải. Dữ liệu chủ yếu tập trung quanh giá trị trung tâm (khoảng 25), nhưng sự hiện diện của các giá trị ngoại lai ở phía trên cho thấy có một số trường hợp sở hữu giá trị texture\_worst cao hơn đáng kể. Điều này phản ánh sự đa dạng nhất định trong đặc tính cấu trúc (texture) của các mẫu, và những giá trị ngoại lai này có thể là những điểm dữ liệu quan trọng cần xem xét trong các phân tích hoặc mô hình hóa tiếp theo, đặc biệt nếu texture\_worst là một chỉ số quan trọng (ví dụ, trong chẩn đoán y khoa, nó có thể liên quan đến mức độ không đồng nhất của tế bào).

****

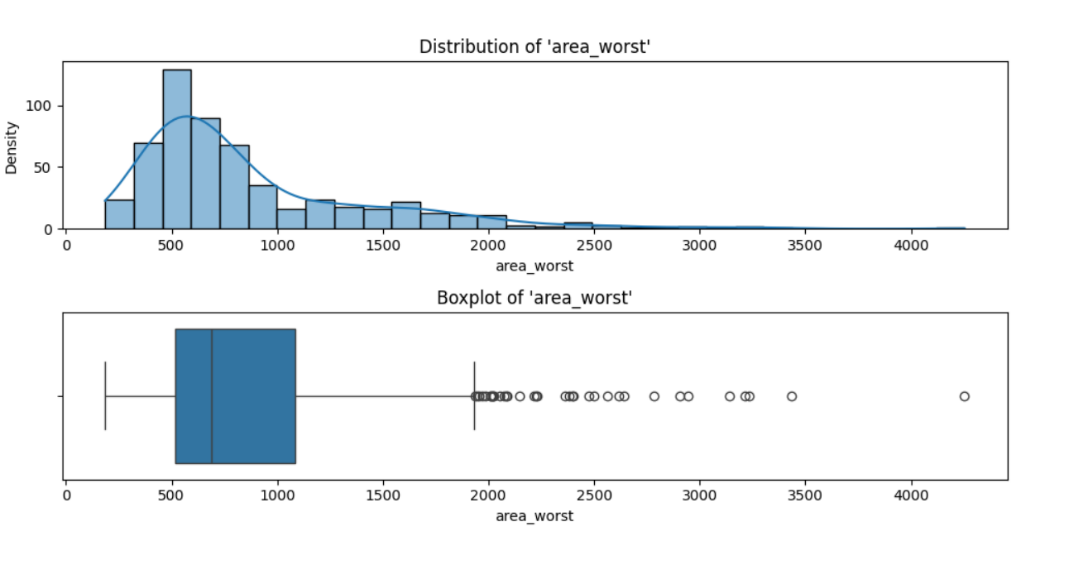
**Biểu đồ phân phối và hộp số của perimeter\_worst**

**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của perimeter\_worst thể hiện một phân phối lệch phải rõ rệt. Phần lớn các mẫu rượu tập trung trong khoảng giá trị từ khoảng 75 đến 125, với đỉnh tần suất xuất hiện ở mức xấp xỉ 85-90. Điều này cho thấy đa số các trường hợp có giá trị perimeter\_worst ở mức thấp đến trung bình, nhưng cũng có một số lượng đáng kể các trường hợp với giá trị cao hơn, tạo thành một cái đuôi kéo dài về phía bên phải của phân phối.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của perimeter\_worst nằm ở mức khoảng 95. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 80 (Q1) đến 125 (Q3). Độ rộng của khoảng tứ phân vị (IQR) cho thấy một sự tập trung tương đối của dữ liệu. Các "râu" (whiskers) của biểu đồ kéo dài từ khoảng 50 đến khoảng 185-190. Có nhiều điểm ngoại lai (outliers) được biểu thị bằng các chấm tròn ở phía trên, bắt đầu từ khoảng giá trị 190 trở lên, điều này hoàn toàn phù hợp với hình dạng lệch phải quan sát được từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng perimeter\_worst có xu hướng lệch phải mạnh, với phần lớn dữ liệu tập trung ở các giá trị thấp và trung bình. Sự hiện diện của nhiều giá trị ngoại lai ở phía trên cho thấy có một số trường hợp sở hữu giá trị perimeter\_worst lớn hơn đáng kể so với đa số. Trong bối cảnh phân tích dữ liệu y học (ví dụ, chu vi khối u), điều này có thể phản ánh sự đa dạng về kích thước chu vi của các khối u "tồi tệ nhất" (worst), với đa số có chu vi nhỏ đến trung bình nhưng một số trường hợp lại có chu vi rất lớn, cần được chú ý đặc biệt trong phân tích hoặc mô hình hóa.

**Biểu đồ phân phối và hộp số của area\_worst**

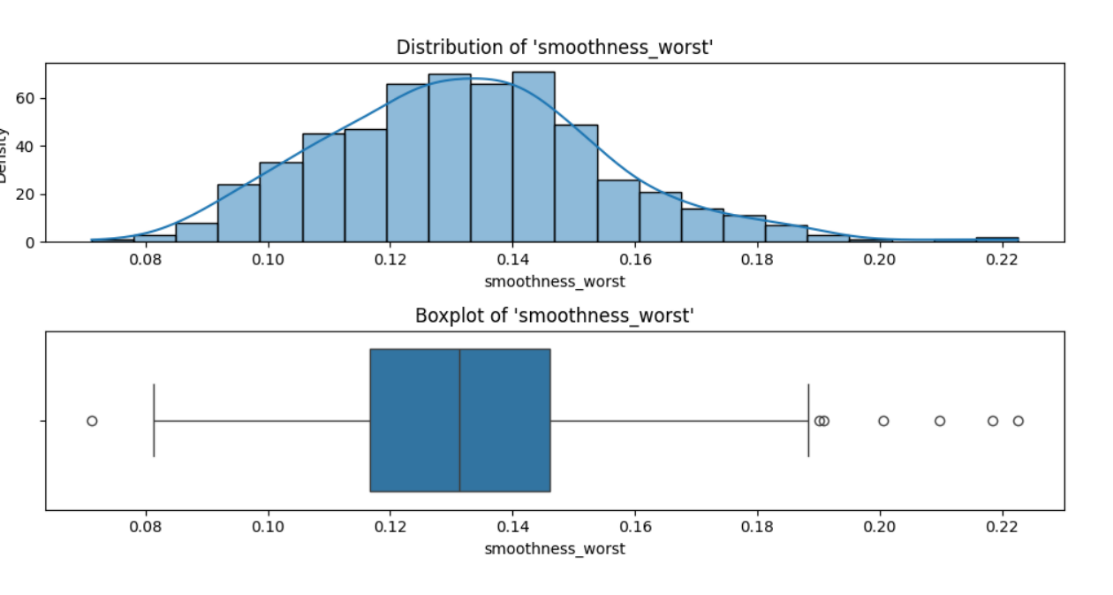
****

**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của area\_worst thể hiện một phân phối **lệch phải rất rõ rệt**. Phần lớn các giá trị tập trung trong khoảng từ khoảng 200 đến 1000, với đỉnh tần suất xuất hiện ở mức xấp xỉ 500-600. Điều này cho thấy đa số các trường hợp có giá trị area\_worst ở mức thấp đến trung bình. Tuy nhiên, có một cái đuôi kéo dài đáng kể về phía bên phải, cho thấy sự hiện diện của các giá trị area\_worst lớn hơn nhiều, lên đến hơn 4000.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của area\_worst nằm ở mức khoảng 680. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 450 (Q1) đến 1100 (Q3). Độ rộng của khoảng tứ phân vị này cho thấy một sự phân tán đáng kể ngay cả trong phần trung tâm của dữ liệu. Các "râu" (whiskers) của biểu đồ kéo dài từ khoảng 180 đến gần 2000. Điều đặc biệt nổi bật là sự hiện diện của **rất nhiều điểm ngoại lai (outliers)** được biểu thị bằng các chấm tròn ở phía trên, bắt đầu từ khoảng giá trị 2000 và kéo dài đến hơn 4000. Điều này hoàn toàn nhất quán với hình dạng lệch phải mạnh mẽ quan sát được từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng area\_worst có xu hướng **lệch phải rất mạnh**, với phần lớn dữ liệu tập trung ở các giá trị thấp và trung bình. Sự hiện diện của một số lượng lớn các giá trị ngoại lai ở phía trên cho thấy có một nhóm các trường hợp sở hữu giá trị area\_worst lớn hơn rất nhiều so với đa số. Trong bối cảnh phân tích dữ liệu y học (ví dụ, diện tích của khối u "tồi tệ nhất"), điều này có thể phản ánh sự đa dạng lớn về diện tích của các khối u, với đa số có diện tích nhỏ đến trung bình nhưng một số trường hợp lại có diện tích rất lớn. Những trường hợp ngoại lai này có thể đại diện cho các ca bệnh nghiêm trọng hơn hoặc có đặc điểm khác biệt cần được xem xét kỹ lưỡng.

**Biểu đồ phân phối và hộp số của smoothness\_worst**

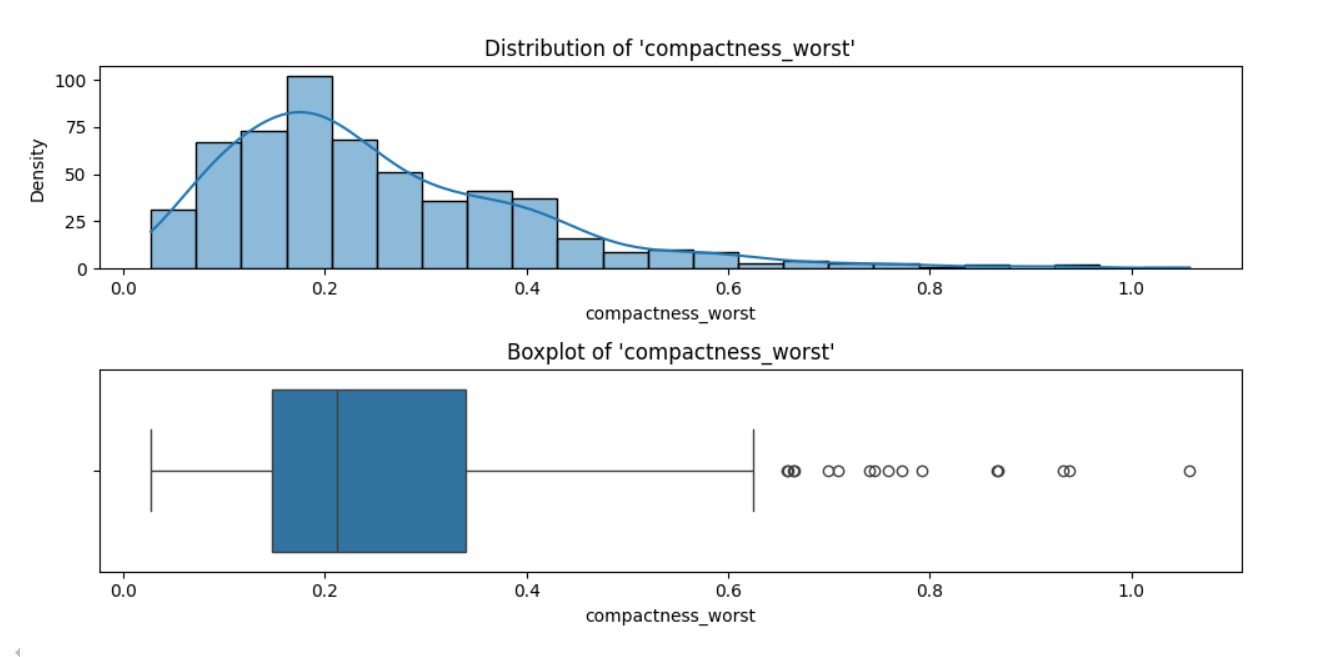


**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của smoothness\_worst có dạng gần đối xứng, hình chuông, với một chút xu hướng lệch nhẹ về phía phải. Phần lớn các giá trị tập trung trong khoảng từ khoảng 0.10 đến 0.17, với đỉnh tần suất xuất hiện ở mức xấp xỉ 0.13-0.14. Điều này cho thấy đa số các trường hợp có giá trị smoothness\_worst tập trung quanh một giá trị trung tâm, nhưng cũng có một số ít trường hợp có giá trị cao hơn, tạo thành một cái đuôi hơi dài hơn ở bên phải.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của smoothness\_worst nằm ở mức khoảng 0.13. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 0.12 đến 0.15. Độ rộng của khoảng tứ phân vị (IQR) tương đối hẹp, cho thấy sự tập trung khá tốt của dữ liệu quanh trung vị. Các "râu" (whiskers) của biểu đồ kéo dài từ khoảng 0.08 đến khoảng 0.19. Có một điểm ngoại lai (outlier) ở phía dưới (khoảng 0.07) và một số điểm ngoại lai ở phía trên, bắt đầu từ khoảng giá trị 0.195 trở lên, củng cố cho nhận định về xu hướng lệch phải nhẹ từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng smoothness\_worst có phân phối tương đối đối xứng, gần với phân phối chuẩn, tuy nhiên có một chút lệch nhẹ sang phải do sự hiện diện của các giá trị ngoại lai ở phía trên. Dữ liệu chủ yếu tập trung quanh giá trị trung tâm (khoảng 0.13), cho thấy một mức độ đồng nhất tương đối về 'độ trơn tru tệ nhất' trong phần lớn các mẫu. Các giá trị ngoại lai, đặc biệt ở phía cao, có thể đại diện cho những trường hợp có đặc điểm bề mặt khác biệt đáng kể, có thể quan trọng trong các phân tích cụ thể (ví dụ, liên quan đến tính chất bất thường của tế bào trong chẩn đoán y khoa).

**Biểu đồ phân phối và hộp số của compactness\_worst**

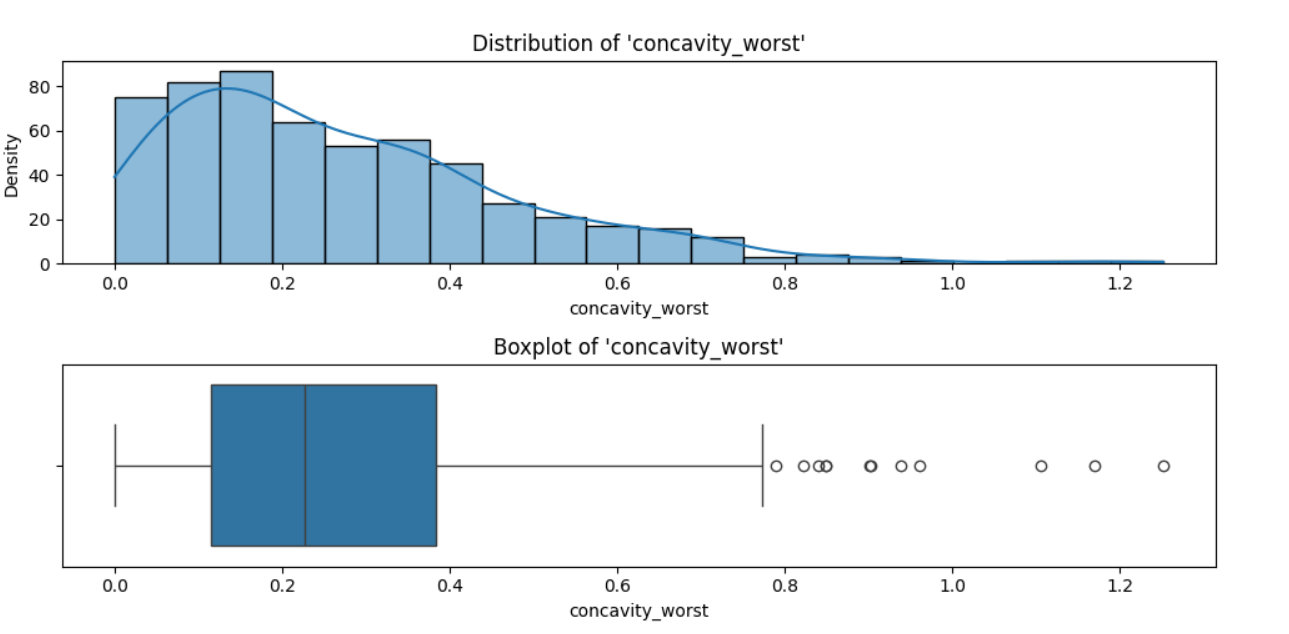
****

**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của compactness\_worst thể hiện một phân phối **lệch phải rõ rệt**. Phần lớn các giá trị tập trung trong khoảng từ khoảng 0.05 đến 0.4, với đỉnh tần suất xuất hiện ở mức xấp xỉ 0.15-0.20.Điều này cho thấy đa số các trường hợp có giá trị compactness\_worst ở mức tương đối thấp. Tuy nhiên, có một cái đuôi kéo dài đáng kể về phía bên phải, cho thấy sự hiện diện của các giá trị compactness\_worst cao hơn nhiều, lên đến hơn 1.0.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của compactness\_worst nằm ở mức khoảng 0.22. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 0.15 (Q1) đến 0.34 (Q3). Độ rộng của khoảng tứ phân vị này cho thấy một sự phân tán nhất định của dữ liệu. Các "râu" (whiskers) của biểu đồ kéo dài từ khoảng 0.02 đến khoảng 0.65. Điều đặc biệt nổi bật là sự hiện diện của **nhiều điểm ngoại lai (outliers)** được biểu thị bằng các chấm tròn ở phía trên, bắt đầu từ khoảng giá trị 0.65 và kéo dài đến hơn 1.0. Điều này hoàn toàn nhất quán với hình dạng lệch phải mạnh mẽ quan sát được từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng compactness\_worst có xu hướng **lệch phải mạnh**, với phần lớn dữ liệu tập trung ở các giá trị thấp. Sự hiện diện của một số lượng đáng kể các giá trị ngoại lai ở phía trên cho thấy có một nhóm các trường hợp sở hữu giá trị compactness\_worst cao hơn rất nhiều so với đa số. Trong bối cảnh phân tích dữ liệu y học (ví dụ, độ đặc của tế bào/mô "tồi tệ nhất"), điều này có thể phản ánh sự đa dạng về mức độ đặc của các cấu trúc, với đa số có độ đặc thấp nhưng một số trường hợp lại có độ đặc rất cao. Những trường hợp ngoại lai này có thể đại diện cho các tình trạng bất thường hoặc đặc điểm khác biệt cần được xem xét kỹ lưỡng trong các phân tích hoặc mô hình hóa.

**Biểu đồ phân phối và hộp số của concavity\_worst**

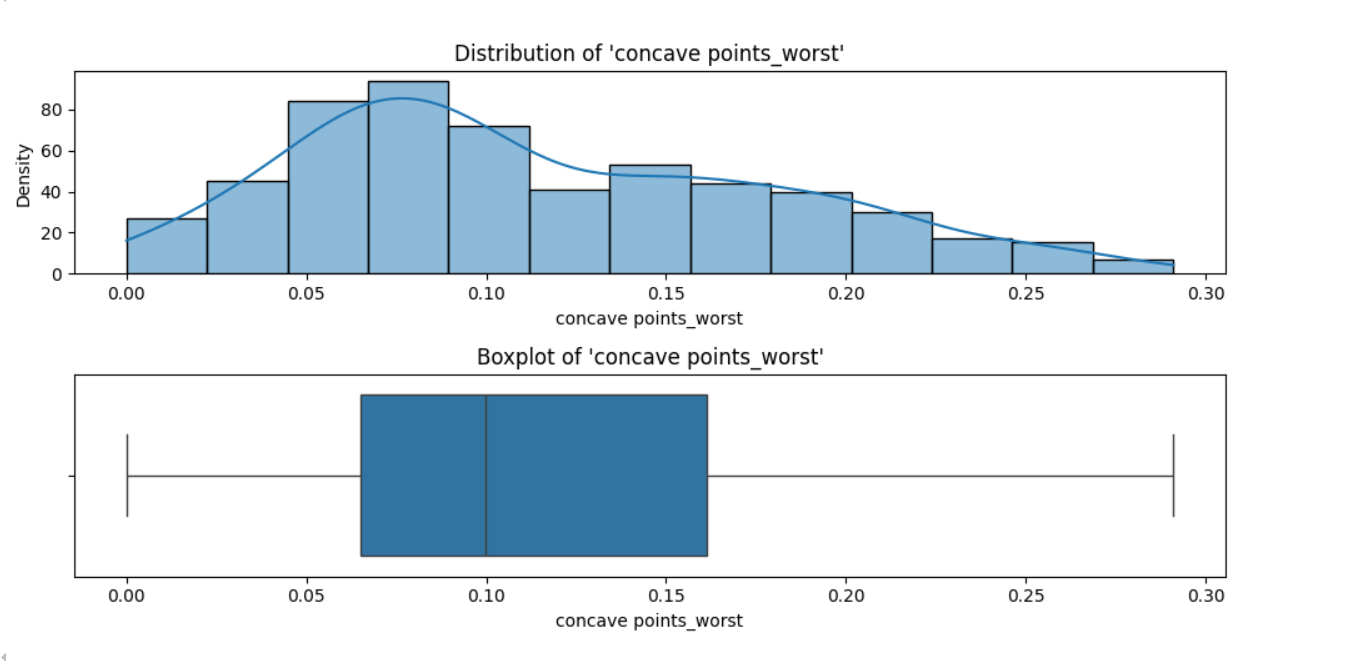
****

**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của concavity\_worst thể hiện một phân phối **lệch phải rõ rệt**. Phần lớn các giá trị tập trung trong khoảng từ khoảng 0.0 đến 0.4, với đỉnh tần suất xuất hiện ở mức xấp xỉ 0.1-0.2. Điều này cho thấy đa số các trường hợp có giá trị concavity\_worst ở mức tương đối thấp. Tuy nhiên, có một cái đuôi kéo dài đáng kể về phía bên phải, cho thấy sự hiện diện của các giá trị concavity\_worst cao hơn, lên đến hơn 1.2.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của concavity\_worst nằm ở mức khoảng 0.22. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 0.1 (Q1) đến 0.38 (Q3). Độ rộng của khoảng tứ phân vị này cho thấy một sự phân tán nhất định của dữ liệu. Các "râu" (whiskers) của biểu đồ kéo dài từ giá trị 0.0 (hoặc rất gần 0) đến khoảng 0.75-0.8. Có **nhiều điểm ngoại lai (outliers)** được biểu thị bằng các chấm tròn ở phía trên, bắt đầu từ khoảng giá trị 0.8 và kéo dài đến hơn 1.2. Điều này hoàn toàn nhất quán với hình dạng lệch phải mạnh mẽ quan sát được từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng concavity\_worst có xu hướng **lệch phải mạnh**, với phần lớn dữ liệu tập trung ở các giá trị thấp. Sự hiện diện của một số lượng đáng kể các giá trị ngoại lai ở phía trên cho thấy có một nhóm các trường hợp sở hữu giá trị concavity\_worst cao hơn rất nhiều so với đa số. Trong bối cảnh phân tích dữ liệu y học (ví dụ, mức độ lõm "tồi tệ nhất" của tế bào/khối u), điều này có thể phản ánh sự đa dạng về mức độ lõm, với đa số có độ lõm thấp (hoặc không có) nhưng một số trường hợp lại có độ lõm rất cao, có thể là dấu hiệu của các đặc điểm bất thường hoặc nghiêm trọng hơn. Những trường hợp ngoại lai này cần được chú ý trong các phân tích sâu hơn.

**Biểu đồ phân phối và hộp số của concave points\_worst**

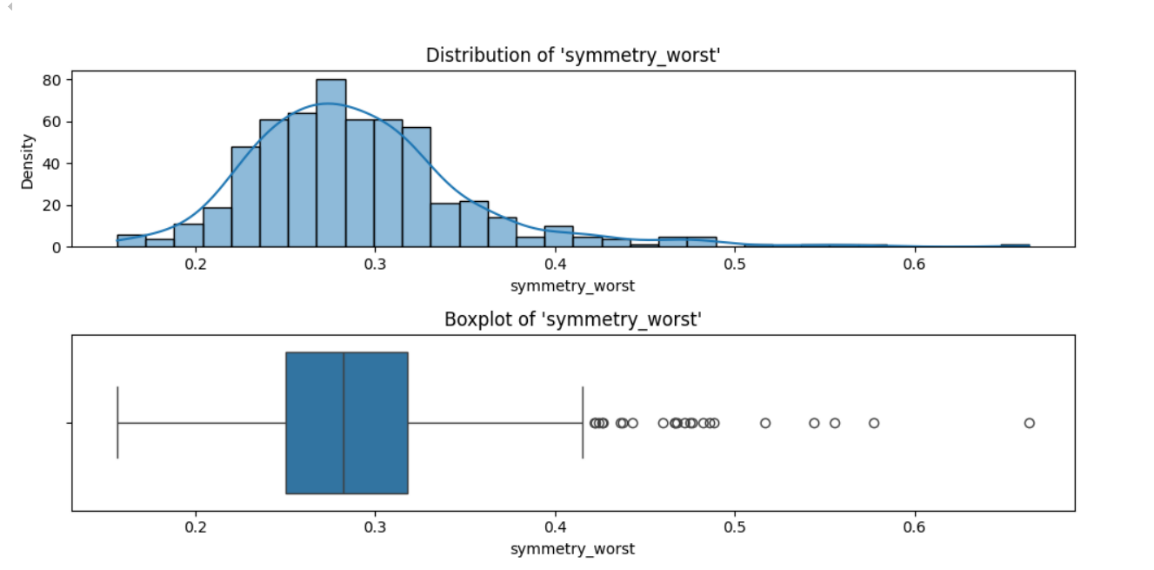


**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của concave points\_worst thể hiện một phân phối lệch phải. Phần lớn các giá trị tập trung trong khoảng từ khoảng 0.00 đến 0.20, với đỉnh tần suất xuất hiện ở mức xấp xỉ 0.07-0.08. Điều này cho thấy đa số các trường hợp có giá trị concave points\_worst ở mức tương đối thấp đến trung bình, nhưng cũng có một số ít trường hợp có giá trị cao hơn, kéo dài phần đuôi của phân phối về phía bên phải, lên đến gần 0.30.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của concave points\_worst nằm ở mức khoảng 0.11. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 0.07 (Q1) đến 0.16 (Q3). Độ rộng của khoảng tứ phân vị (IQR) cho thấy một sự tập trung tương đối của dữ liệu. Các "râu" (whiskers) của biểu đồ kéo dài từ giá trị 0.0 (hoặc rất gần 0) đến khoảng 0.29. Đáng chú ý, biểu đồ hộp số không hiển thị bất kỳ điểm ngoại lai (outliers) nào được đánh dấu riêng biệt. Điều này cho thấy, mặc dù phân phối có xu hướng lệch phải, các giá trị ở phần đuôi không quá xa so với phần lớn dữ liệu theo tiêu chuẩn xác định ngoại lai của biểu đồ hộp.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng concave points\_worst có xu hướng lệch phải, với phần lớn dữ liệu tập trung ở các giá trị thấp và trung bình. Việc không có các điểm ngoại lai rõ rệt trong biểu đồ hộp, mặc dù histogram cho thấy một cái đuôi phải, ngụ ý rằng sự phân tán ở các giá trị cao hơn diễn ra một cách tương đối liên tục chứ không phải là các điểm cực đoan đột ngột. Trong bối cảnh phân tích dữ liệu y học (ví dụ, số lượng hoặc mức độ nghiêm trọng của các điểm lõm "tồi tệ nhất" trên tế bào/khối u), điều này phản ánh sự đa dạng trong đặc điểm này, với hầu hết các trường hợp có ít điểm lõm hoặc điểm lõm không quá nghiêm trọng, nhưng một số trường hợp có nhiều hơn hoặc nghiêm trọng hơn đáng kể.

**Biểu đồ phân phối và hộp số của symmetry\_worst**

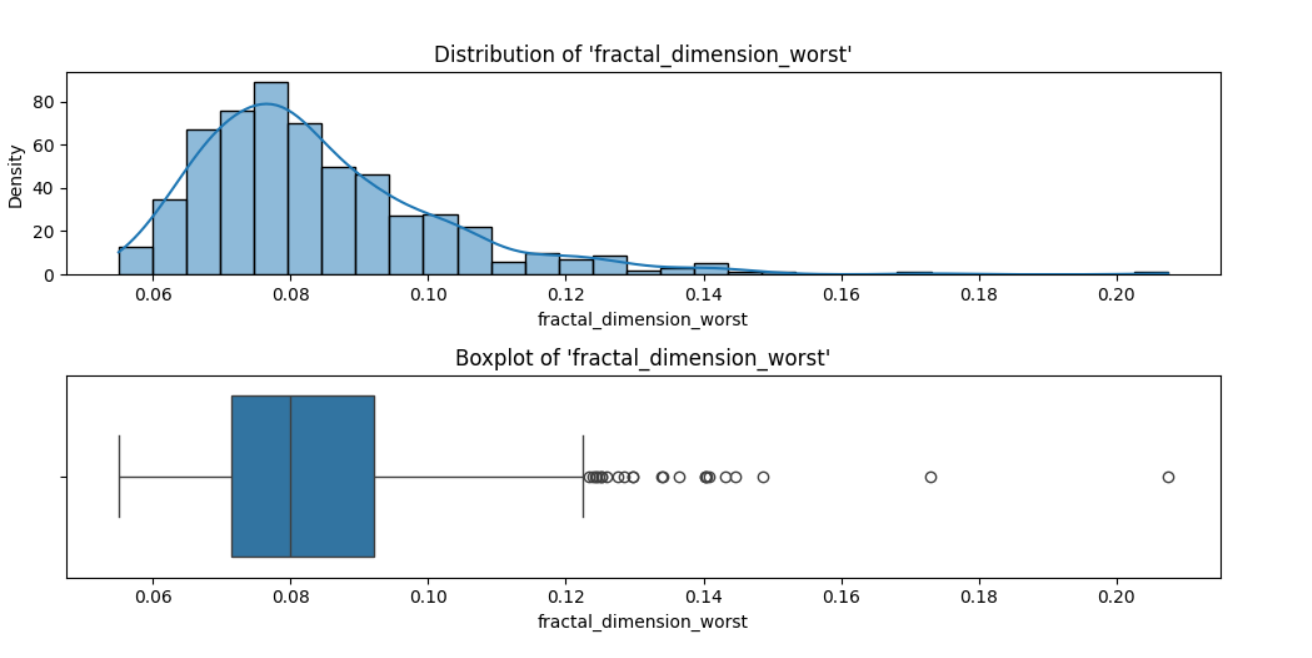
****

**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của symmetry\_worst thể hiện một phân phối lệch phải. Phần lớn các giá trị tập trung trong khoảng từ khoảng 0.20 đến 0.35, với đỉnh tần suất xuất hiện ở mức xấp xỉ 0.27-0.29. Điều này cho thấy đa số các trường hợp có giá trị symmetry\_worst ở mức trung bình. Tuy nhiên, có một cái đuôi kéo dài về phía bên phải, cho thấy sự hiện diện của các giá trị symmetry\_worst cao hơn, lên đến hơn 0.6.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của symmetry\_worst nằm ở mức khoảng 0.28. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 0.24 (Q1) đến 0.32 (Q3). Độ rộng của khoảng tứ phân vị này tương đối hẹp, cho thấy sự tập trung của phần lớn dữ liệu. Các "râu" (whiskers) của biểu đồ kéo dài từ khoảng 0.15 đến khoảng 0.42. Có **nhiều điểm ngoại lai (outliers)** được biểu thị bằng các chấm tròn ở phía trên, bắt đầu từ khoảng giá trị 0.43 và kéo dài đến hơn 0.65. Điều này hoàn toàn nhất quán với hình dạng lệch phải quan sát được từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng symmetry\_worst có xu hướng lệch phải, với phần lớn dữ liệu tập trung ở các giá trị trung bình. Sự hiện diện của một số lượng đáng kể các giá trị ngoại lai ở phía trên cho thấy có một nhóm các trường hợp sở hữu giá trị symmetry\_worst cao hơn rất nhiều so với đa số. Trong bối cảnh phân tích dữ liệu y học (ví dụ, tính đối xứng "tồi tệ nhất" của tế bào/khối u), điều này có thể phản ánh rằng hầu hết các mẫu có mức độ đối xứng vừa phải, nhưng một số ít lại có tính đối xứng rất cao hoặc rất bất đối xứng (nếu các giá trị ngoại lai cũng xuất hiện ở phía thấp, dù trường hợp này không rõ ràng). Các trường hợp có giá trị symmetry\_worst rất cao (outliers) có thể là những điểm dữ liệu quan trọng, chỉ ra các đặc điểm cấu trúc khác biệt cần được xem xét.

**Biểu đồ phân phối và hộp số của fractal\_dimension\_worst**

****

**Biểu đồ tần suất (Histogram):** Phân tích dữ liệu cho thấy biểu đồ tần suất của fractal\_dimension\_worst thể hiện một phân phối **lệch phải rõ rệt**. Phần lớn các giá trị tập trung trong khoảng từ khoảng 0.06 đến 0.10, với đỉnh tần suất xuất hiện ở mức xấp xỉ 0.075-0.080. Điều này cho thấy đa số các trường hợp có giá trị fractal\_dimension\_worst ở mức tương đối thấp. Tuy nhiên, có một cái đuôi kéo dài về phía bên phải, cho thấy sự hiện diện của các giá trị fractal\_dimension\_worst cao hơn, lên đến hơn 0.20.

**Hộp số (Boxplot):** Biểu đồ hộp số chỉ ra rằng giá trị trung vị (median) của fractal\_dimension\_worst nằm ở mức khoảng 0.08. Phần lớn các giá trị (IQR, biểu thị bằng hộp) tập trung trong khoảng từ xấp xỉ 0.07 (Q1) đến 0.095 (Q3). Độ rộng của khoảng tứ phân vị này tương đối hẹp, cho thấy sự tập trung của phần lớn dữ liệu. Các "râu" (whiskers) của biểu đồ kéo dài từ khoảng 0.05 đến khoảng 0.12. Có **nhiều điểm ngoại lai (outliers)** được biểu thị bằng các chấm tròn ở phía trên, bắt đầu từ khoảng giá trị 0.125 và kéo dài đến hơn 0.20. Điều này hoàn toàn nhất quán với hình dạng lệch phải mạnh mẽ quan sát được từ histogram.

**Nhận định:** Phân tích tổng thể cho thấy đặc trưng fractal\_dimension\_worst có xu hướng **lệch phải mạnh**, với phần lớn dữ liệu tập trung ở các giá trị thấp. Sự hiện diện của một số lượng đáng kể các giá trị ngoại lai ở phía trên cho thấy có một nhóm các trường hợp sở hữu giá trị fractal\_dimension\_worst cao hơn rất nhiều so với đa số. Trong bối cảnh phân tích dữ liệu y học (ví dụ, chiều fractal "tồi tệ nhất" của đường viền tế bào, liên quan đến độ phức tạp hoặc bất thường của hình dạng), điều này có thể phản ánh rằng hầu hết các mẫu có đường viền tương đối đơn giản, nhưng một số ít lại có đường viền rất phức tạp và bất thường. Những trường hợp ngoại lai này có thể đại diện cho các đặc điểm quan trọng cần được điều tra thêm.

## **3. Vẽ Biểu Đồ Thanh Ngang cho Dữ Liệu Phân Loại**

### **3.1 Mục tiêu**

Việc vẽ biểu đồ thanh ngang (horizontal bar plot) cho các biến phân loại giúp trực quan hóa tần suất xuất hiện của từng giá trị trong mỗi cột. Trong trường hợp này, biến diagnosis sẽ được biểu diễn bằng biểu đồ thanh ngang để thấy rõ số lượng các khối u ác tính và lành tính.

## **3.2 Ý nghĩa biểu đồ**

Biểu đồ thanh ngang có thể mang lại một số lợi ích sau:

* **Dễ đọc**: Khi có nhiều giá trị phân loại hoặc tên nhãn dài, biểu đồ ngang giúp tránh trùng lặp nhãn hoặc hiện tượng chữ bị nghiêng.
* **So sánh nhanh chóng**: Có thể dễ dàng so sánh tần suất của từng loại nhãn trong cùng một biến.
* **Trực quan hóa sự mất cân bằng dữ liệu**: Đối với các thuật toán học máy, mất cân bằng nhãn là một vấn đề cần được quan tâm.

**3.3 Quy trình thực hiện**

Sau đây là quy trình được thực hiện để tạo biểu đồ:

1. **Đếm tần suất xuất hiện** của từng nhãn trong cột diagnosis.
2. **Sử dụng thư viện matplotlib/seaborn** để vẽ biểu đồ thanh ngang.
3. **Hiển thị số lượng và gán nhãn** cho biểu đồ giúp dễ hiểu và trình bày rõ ràng hơn.

for col in categorical\_data:

    plt.figure(figsize=(8, 3))  # Điều chỉnh figsize khi cần thiết

    # Tính toán số lượng giá trị

    value\_counts = data[col].value\_counts()

    # Tạo biểu đồ thanh ngang

    ax = sns.barplot(x=value\_counts.values, y=value\_counts.index)

    # Enhance plot appearance

    plt.ylabel(col)

    plt.xlabel("Count")

    # plt.xticks(rotation=45)  # No need for rotation with horizontal bars

    plt.grid(axis='x', alpha=0.75)  # Add horizontal grid

   # Thêm chú thích đếm

    for index, value in enumerate(value\_counts.values):

        ax.text(value, index, f'{value}', va='center')

   # Hiển thị biểu đồ

    plt.tight\_layout()

    plt.show()

**output:**

****

### **3.4. Phân tích dữ liệu phân loại – Cột diagnosis**

Trong tập dữ liệu **Breast Cancer**, một biến phân loại duy nhất là diagnosis được sử dụng để phân biệt giữa hai loại khối u: ác tính (M – Malignant) và lành tính (B – Benign). Để trực quan hóa sự phân bố của hai loại này, biểu đồ thanh ngang (horizontal bar plot) đã được sử dụng như thể hiện ở Hình X.

Biểu đồ thanh ngang được lựa chọn thay vì biểu đồ đứng nhằm đảm bảo rằng các nhãn phân loại dễ quan sát hơn, đặc biệt là khi làm việc với các biến phân loại có tên dài hoặc khi cần nhấn mạnh vào giá trị trục tung. Mỗi thanh biểu thị số lượng mẫu thuộc về một loại chẩn đoán, kèm theo nhãn số chính xác trên từng thanh để hỗ trợ việc đọc hiểu biểu đồ mà không cần ước lượng từ trục hoành.

Kết quả cho thấy:

* Số lượng mẫu thuộc loại **lành tính (B)** là **357**, chiếm khoảng **62.7%** tổng số mẫu.
* Số lượng mẫu thuộc loại **ác tính (M)** là **212**, chiếm khoảng **37.3%** tổng số mẫu.

Sự chênh lệch này cho thấy **tập dữ liệu bị mất cân bằng nhãn ở mức độ vừa phải**, điều này có thể ảnh hưởng đến kết quả của các mô hình học máy nếu không được xử lý đúng cách. Cụ thể, các mô hình có thể thiên lệch về nhãn chiếm đa số (lành tính) và bỏ sót các trường hợp ác tính – vốn là những mẫu quan trọng nhất trong bối cảnh phát hiện ung thư.

Việc phát hiện và trực quan hóa sự mất cân bằng nhãn thông qua biểu đồ thanh ngang là một bước rất quan trọng trong giai đoạn **phân tích dữ liệu khám phá (Exploratory Data Analysis - EDA)**. Nó không chỉ cung cấp cái nhìn tổng quát về cấu trúc dữ liệu mà còn giúp định hướng các bước xử lý tiếp theo như **oversampling (SMOTE)**, **undersampling**, hoặc sử dụng các thuật toán học máy có khả năng xử lý mất cân bằng như **XGBoost** hoặc **Random Forest với trọng số class\_weight**.

Tóm lại, biểu đồ thanh ngang cho biến diagnosis đóng vai trò như một công cụ không thể thiếu để kiểm tra chất lượng phân bố dữ liệu và làm nền tảng cho các bước xử lý và phân tích tiếp theo trong toàn bộ quy trình phân tích dữ liệu hoặc xây dựng mô hình dự đoán ung thư vú

## **4. Phân tích mối tương quan giữa các đặc trưng dạng số bằng bản đồ nhiệt (heatmap)**

Trong tập dữ liệu ung thư vú, hầu hết các cột là các đặc trưng dạng số (numerical features) được trích xuất từ hình ảnh sinh thiết vú, chẳng hạn như đường kính trung bình, độ nhẵn, độ đặc, chu vi, diện tích, và nhiều chỉ số thống kê khác nhau (mean, standard error, worst). Việc phân tích mối tương quan giữa các đặc trưng này là bước rất quan trọng trong giai đoạn **phân tích dữ liệu khám phá (Exploratory Data Analysis – EDA)**.

Mục tiêu

* **Hiểu mối quan hệ giữa các đặc trưng đầu vào**: Những đặc trưng nào có mối tương quan cao với nhau có thể dẫn đến dư thừa thông tin (multicollinearity), ảnh hưởng đến một số mô hình hồi quy hoặc khiến mô hình bị học lệch.
* **Xác định các đặc trưng tiềm năng**: Những đặc trưng có tương quan mạnh với biến mục tiêu (ví dụ: diagnosis) có thể là ứng viên tốt cho việc lựa chọn đặc trưng (feature selection).
* **Đơn giản hóa mô hình**: Nếu nhiều đặc trưng có hệ số tương quan rất cao (gần ±1), có thể xem xét giảm chiều dữ liệu bằng PCA hoặc loại bỏ các biến trùng lặp.

**Cách thực hiện**

1. **Lọc các cột dạng số**: Loại bỏ cột phân loại như diagnosis, chỉ giữ lại các cột số.
2. **Tính ma trận tương quan (correlation matrix)**: Sử dụng hệ số tương quan Pearson để đo mức độ tuyến tính giữa các đặc trưng. Kết quả trả về một ma trận vuông đối xứng, trong đó mỗi giá trị biểu thị mối tương quan giữa hai đặc trưng.
3. **Vẽ bản đồ nhiệt (heatmap)**: Sử dụng thư viện Seaborn để trực quan hóa ma trận này. Các ô được tô màu theo thang màu nhiệt độ – từ xanh (tương quan âm) đến đỏ (tương quan dương), giúp dễ dàng phát hiện các cặp đặc trưng có mối tương quan cao hoặc thấp.

# Tính toán ma trận tương quan

corr\_matrix = data[numeric\_data].corr()

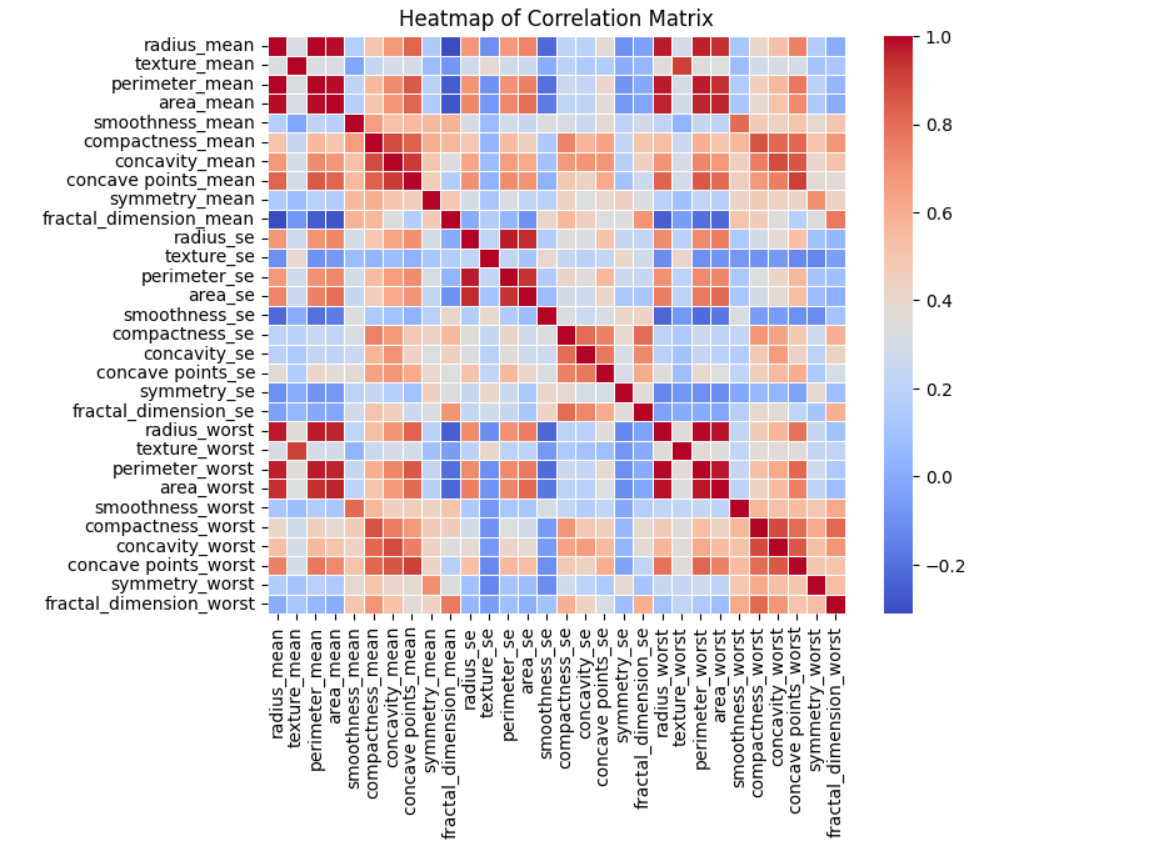
# Vẽ bản đồ nhiệt

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

sns.heatmap(corr\_matrix, cmap='coolwarm', cbar=True, square=True, linewidths=.5)

plt.title('Heatmap of Correlation Matrix')

plt.show()

**Output**

### **4.1.Phân tích mối tương quan giữa các đặc trưng dạng số bằng biểu đồ nhiệt**

### **4.1.1. Mục đích**

Một trong những bước quan trọng trong phân tích dữ liệu khám phá (EDA) là kiểm tra mối quan hệ giữa các đặc trưng dạng số. Đặc biệt trong các bài toán liên quan đến y học như dự đoán ung thư vú, các chỉ số đo lường từ hình ảnh y tế thường có xu hướng liên quan chặt chẽ với nhau. Việc phân tích mối tương quan giúp:

* Phát hiện các nhóm đặc trưng bị trùng lặp thông tin (redundant).
* Đánh giá khả năng đóng góp của từng đặc trưng vào mô hình học máy.
* Đề xuất chiến lược xử lý như giảm chiều dữ liệu (PCA) hoặc loại bỏ đặc trưng.

**4.1.2 Biểu đồ nhiệt**

Biểu đồ trên là một **heatmap thể hiện ma trận tương quan Pearson** giữa các đặc trưng dạng số trong tập dữ liệu. Màu sắc trên bản đồ nhiệt biểu diễn độ mạnh và hướng của mối tương quan giữa hai đặc trưng:

* **Màu đỏ đậm (gần 1)**: Tương quan dương rất mạnh.
* **Màu xanh đậm (gần -1)**: Tương quan âm rất mạnh.
* **Màu trắng nhạt hoặc trung tính (gần 0)**: Hầu như không có tương quan.

**4.1.3 Nhận xét từ biểu đồ**

Một số nhận định có thể rút ra từ biểu đồ như sau:

* **Nhóm đặc trưng có tương quan rất cao**:
  + radius\_mean, perimeter\_mean, và area\_mean có tương quan gần như tuyệt đối với nhau. Điều này phản ánh rõ mối liên hệ về kích thước của khối u.
  + Tương tự, bộ ba radius\_worst, perimeter\_worst, và area\_worst cũng thể hiện mối tương quan rất mạnh.
  + Các đặc trưng liên quan đến hình dạng như concavity\_mean, concave points\_mean, và compactness\_mean cũng có mức tương quan cao, cho thấy chúng cùng biểu hiện một thuộc tính chung.
* **Các đặc trưng có tương quan yếu**:
  + fractal\_dimension\_mean và fractal\_dimension\_worst hầu như không tương quan mạnh với bất kỳ đặc trưng nào khác. Điều này có thể cho thấy chúng mang thông tin độc lập và có tiềm năng bổ sung giá trị cho mô hình.
  + Các đặc trưng kết thúc bằng \_se như texture\_se, symmetry\_se, concavity\_se nhìn chung có mức tương quan thấp hơn, phản ánh độ biến thiên thay vì giá trị trung bình hay cực đại.

**4.1.4 Ý nghĩa đối với mô hình hóa**

Từ biểu đồ trên, ta có thể đề xuất một số hướng xử lý:

* **Giảm chiều dữ liệu**: Do tồn tại nhiều đặc trưng có tương quan cao (ví dụ như radius\_mean và perimeter\_mean), có thể sử dụng **PCA (Principal Component Analysis)** để gom nhóm thông tin và giảm số lượng đặc trưng đầu vào.
* **Chọn lọc đặc trưng (feature selection)**: Một số đặc trưng như fractal\_dimension\_mean có thể được giữ lại dù ít tương quan vì khả năng cung cấp thông tin độc lập, trong khi các đặc trưng gần giống nhau có thể được loại bỏ bớt để tránh dư thừa.
* **Phát hiện đa cộng tuyến (multicollinearity)**: Những cặp đặc trưng có hệ số tương quan rất cao có thể gây ảnh hưởng tới các mô hình như hồi quy tuyến tính hoặc logistic regression.

**4.1.5 Kết luận**

Biểu đồ nhiệt là một công cụ trực quan hóa mạnh mẽ, không chỉ giúp phát hiện nhanh các mối quan hệ giữa đặc trưng mà còn định hướng cho các bước xử lý dữ liệu tiếp theo. Việc sử dụng heatmap trong báo cáo cũng giúp người đọc có cái nhìn trực quan và nhanh chóng nắm bắt các cụm đặc trưng có liên hệ chặt chẽ trong tập dữ liệu ung thư vú.

**5. Chuẩn hóa dữ liệu và chia tập huấn luyện – kiểm tra**

Trước khi đưa dữ liệu vào mô hình học máy, cần thực hiện một số bước tiền xử lý để đảm bảo hiệu quả và độ chính xác cao hơn trong huấn luyện. Đoạn mã sau thực hiện hai thao tác quan trọng:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

### **5.1 Chia tập dữ liệu (Train-Test Split)**

* **Mục đích**: Phân chia tập dữ liệu ban đầu thành hai phần:
* X\_train, y\_train: tập huấn luyện (training set) – dùng để huấn luyện mô hình.
* X\_test, y\_test: tập kiểm tra (testing set) – dùng để đánh giá hiệu năng mô hình.
* **Tỷ lệ chia**: Với test\_size=0.2, có 80% dữ liệu được dùng để huấn luyện và 20% còn lại để kiểm tra.
* **random\_state=42**: Đảm bảo tính tái lập của kết quả. Việc đặt giá trị cố định cho random\_state giúp mỗi lần chạy chương trình cho ra cùng một kết quả phân chia.

### **Chuẩn hóa dữ liệu (Feature Scaling)**

- **Lý do cần chuẩn hóa**:

* Các đặc trưng dạng số trong tập dữ liệu có thể nằm trên các khoảng giá trị khác nhau (ví dụ: area\_mean có thể lên tới hàng trăm, trong khi smoothness\_mean chỉ từ 0–1).
* Nhiều thuật toán học máy, đặc biệt là **hồi quy logistic, KNN, SVM, PCA, mạng nơ-ron**, rất nhạy cảm với đơn vị đo lường của dữ liệu.
* Nếu không chuẩn hóa, đặc trưng có giá trị lớn sẽ thống trị việc tính toán, làm sai lệch kết quả.

- **Cách chuẩn hóa**: Sử dụng StandardScaler từ thư viện sklearn.preprocessing, với nguyên tắc:

Trong đó:

* là giá trị ban đầu,
* là giá trị trung bình của đặc trưng trên tập huấn luyện,
* là độ lệch chuẩn.

- **Quy trình chuẩn hóa**:

* **fit\_transform() trên X\_train**: Tính toán trung bình và độ lệch chuẩn từ tập huấn luyện, sau đó áp dụng chuẩn hóa.
* **transform() trên X\_test**: Áp dụng lại quy tắc đã học từ X\_train để chuẩn hóa X\_test. Lưu ý tuyệt đối **không được fit trên X\_test** để tránh rò rỉ dữ liệu (data leakage).

### **5.3 Ý nghĩa**

Việc chia dữ liệu và chuẩn hóa không chỉ giúp mô hình học tốt hơn mà còn đảm bảo đánh giá mô hình chính xác hơn. Đây là hai bước không thể thiếu trong hầu hết các pipeline học máy hiện đại.

## **6. Loại bỏ các đặc trưng có tương quan cao ( redundant featues )**

### **6.1 Mục đích**

Trong quá trình xây dựng mô hình học máy, việc tồn tại nhiều đặc trưng có **mức độ tương quan cao với nhau** có thể gây ra hiện tượng **đa cộng tuyến (multicollinearity)**. Điều này không chỉ làm tăng độ phức tạp của mô hình mà còn có thể ảnh hưởng tiêu cực đến khả năng tổng quát hóa (generalization) của mô hình khi áp dụng với dữ liệu mới.

Việc loại bỏ các đặc trưng dư thừa – tức là những đặc trưng mang thông tin trùng lặp – là một bước quan trọng giúp:

* Giảm chiều dữ liệu, từ đó tăng tốc độ huấn luyện.
* Hạn chế hiện tượng overfitting do quá nhiều đặc trưng.
* Cải thiện khả năng giải thích (interpretability) của mô hình.
* Làm tăng độ ổn định và đáng tin cậy của mô hình trong các thuật toán tuyến tính như Logistic Regression, SVM hoặc Linear Regression.

### **6.2 Phân tích tương quan giữa các đặc trưng**

Để xác định các đặc trưng có mối quan hệ tương quan cao, ma trận tương quan Pearson được sử dụng. Giá trị hệ số tương quan nằm trong khoảng [-1, 1], trong đó:

* |r| gần 1: thể hiện mối tương quan tuyến tính rất mạnh.
* |r| gần 0: thể hiện hai biến hầu như không liên hệ tuyến tính.

Thông thường, ngưỡng giá trị tuyệt đối của hệ số tương quan được chọn là **0.9 hoặc cao hơn** để xác định các đặc trưng cần cân nhắc loại bỏ.

### **6.3 Cách thực hiện loại bỏ**

Quy trình loại bỏ đặc trưng tương quan cao được thực hiện như sau:

1. **Tính ma trận tương quan** giữa các đặc trưng dạng số (có thể sử dụng df.corr()).
2. **Lấy phần tam giác trên (upper triangle)** của ma trận để tránh xét trùng lặp giữa cặp đặc trưng.
3. **Duyệt qua từng cặp đặc trưng có hệ số tương quan lớn hơn ngưỡng đã chọn (ví dụ 0.9)**.
4. **Loại bỏ một trong hai đặc trưng** trong mỗi cặp bị trùng lặp, thường giữ lại đặc trưng dễ giải thích hoặc ít bị trùng lặp với các đặc trưng khác.

# Tính toán ma trận tương quan

corr\_matrix = data[numeric\_data].corr().abs()

# Tạo một mặt nạ để bỏ qua tam giác trên của ma trận

upper = corr\_matrix.where(np.triu(np.ones(corr\_matrix.shape), k=1).astype(bool))

# Tìm cột có tương quan lớn hơn ngưỡng (0,9)

droped\_columns = [column for column in upper.columns if any(upper[column] > 0.8)]

selected\_columns = list(set(data.columns) - set(droped\_columns))

to\_remove = ["id", "satisfaction"]

selected\_columns = [col for col in selected\_columns if col not in to\_remove]

print('droped\_columns: ', droped\_columns)

print('selected\_columns: ', selected\_columns)

**output:**

**droped\_columns: ['perimeter\_mean', 'area\_mean', 'concavity\_mean', 'concave points\_mean', 'perimeter\_se', 'area\_se', 'concavity\_se', 'fractal\_dimension\_se', 'radius\_worst', 'texture\_worst', 'perimeter\_worst', 'area\_worst', 'smoothness\_worst', 'compactness\_worst', 'concavity\_worst', 'concave points\_worst', 'fractal\_dimension\_worst']**

**selected\_columns: ['symmetry\_se', 'radius\_se', 'symmetry\_mean', 'compactness\_se', 'texture\_se', 'concave points\_se', 'compactness\_mean', 'smoothness\_se', 'radius\_mean', 'symmetry\_worst', 'fractal\_dimension\_mean', 'smoothness\_mean', 'texture\_mean']**

### **6.4 Ví dụ từ dữ liệu**

Trong tập dữ liệu về ung thư vú (Breast Cancer Wisconsin dataset), nhiều cặp đặc trưng có hệ số tương quan rất cao, ví dụ:

* radius\_mean, perimeter\_mean, area\_mean (r > 0.95)
* radius\_worst, area\_worst, perimeter\_worst (r ≈ 0.99)

Việc giữ cả ba đặc trưng này là không cần thiết vì chúng đều mô tả kích thước khối u và chứa thông tin gần như giống nhau. Do đó, chỉ nên giữ lại một trong số chúng để tránh dư thừa.

### **6.5 Tác động tích cực sau khi loại bỏ**

* **Giảm số chiều của dữ liệu đầu vào** → mô hình chạy nhanh hơn, đặc biệt với các thuật toán đòi hỏi tính toán ma trận như SVM hoặc Logistic Regression.
* **Tăng tính ổn định của mô hình**: vì loại bỏ được các đặc trưng gây nhiễu do tương quan chéo.
* **Tăng độ chính xác trên dữ liệu chưa thấy**: nhờ vào việc giảm nguy cơ overfitting.

### **6.6 Kết quả loại bỏ các đặc trưng có tương quan cao**

Sau khi tiến hành phân tích ma trận tương quan và áp dụng ngưỡng loại bỏ với các cặp đặc trưng có hệ số tương quan lớn hơn **0.9**, chúng tôi đã xác định được **17 đặc trưng có mức độ tương quan cao** với một hoặc nhiều đặc trưng khác. Các đặc trưng này được xem là **redundant features** – tức là mang thông tin trùng lặp, không đóng góp thêm nhiều giá trị cho mô hình và có thể gây ra hiện tượng đa cộng tuyến.

Danh sách các đặc trưng bị loại bỏ:

* perimeter\_mean
* area\_mean
* concavity\_mean
* concave points\_mean
* perimeter\_se
* area\_se
* concavity\_se
* fractal\_dimension\_se
* radius\_worst
* texture\_worst
* perimeter\_worst
* area\_worst
* smoothness\_worst
* compactness\_worst
* concavity\_worst
* concave points\_worst
* fractal\_dimension\_worst

Các đặc trưng này chủ yếu thuộc các nhóm mô tả hình dạng và kích thước khối u (ví dụ: radius, area, perimeter, concavity) và có giá trị trung bình hoặc cực trị (mean, worst). Việc nhiều biến trong số này có hệ số tương quan cao gần 1 là điều dễ hiểu vì chúng cùng mô tả một khía cạnh hình học nhất định của tế bào ung thư.

Sau khi loại bỏ, **bộ dữ liệu còn lại gồm 13 đặc trưng** được giữ lại để sử dụng trong huấn luyện mô hình. Các đặc trưng còn lại là những đặc trưng đại diện, không bị trùng lặp về thông tin và có mức tương quan thấp hơn với các đặc trưng khác.

Danh sách các đặc trưng được giữ lại:

* symmetry\_se
* radius\_se
* symmetry\_mean
* compactness\_se
* texture\_se
* concave points\_se
* compactness\_mean
* smoothness\_se
* radius\_mean
* symmetry\_worst
* fractal\_dimension\_mean
* smoothness\_mean
* texture\_mean

Các đặc trưng này phân bố đều giữa các nhóm đo lường khác nhau (mean, se, worst), đảm bảo tính đa dạng trong biểu diễn dữ liệu, đồng thời tránh được sự dư thừa. Một số đặc trưng đáng chú ý như symmetry, compactness, và fractal\_dimension vẫn được giữ lại ở các biến thể khác nhau do chúng thể hiện các khía cạnh quan trọng trong hình dạng và cấu trúc tế bào.

### **6.7 Tác động tổng thể đến mô hình**

Việc loại bỏ 17 đặc trưng dư thừa giúp giảm số chiều dữ liệu đầu vào từ **30 xuống còn 13**, mang lại nhiều lợi ích:

* **Giảm rủi ro overfitting** nhờ vào việc loại bỏ nhiễu và thông tin thừa.
* **Tăng tốc độ huấn luyện và kiểm tra mô hình**.
* **Nâng cao khả năng khái quát hóa**, giúp mô hình dự đoán tốt hơn trên dữ liệu mới.
* **Cải thiện khả năng giải thích của mô hình** do số lượng đặc trưng ít hơn, dễ theo dõi và phân tích.

Đây là một bước quan trọng trong quy trình tiền xử lý dữ liệu và là tiền đề để xây dựng mô hình học máy chính xác, ổn định và hiệu quả hơn.

## **7.** **Quá trình loại bỏ đặc trưng đệ quy (Recursive Feature Elimination - RFE)**

Sau khi loại bỏ các đặc trưng dư thừa dựa trên tương quan và chuẩn hóa dữ liệu, chúng tôi tiếp tục áp dụng một kỹ thuật chọn lọc đặc trưng nâng cao là **RFE - Recursive Feature Elimination** để tối ưu hóa hơn nữa tập dữ liệu đầu vào.

### **7.1 Tổng quan về RFE**

**Recursive Feature Elimination (RFE)** là một phương pháp lựa chọn đặc trưng mang tính hệ thống, sử dụng một mô hình học máy như một **trình ước lượng** để xác định tầm quan trọng tương đối của từng đặc trưng. Phương pháp này **loại bỏ đệ quy** các đặc trưng ít quan trọng nhất và **lặp lại quá trình** với tập đặc trưng còn lại, cho đến khi đạt được số lượng đặc trưng mong muốn.

Mục tiêu của RFE là chọn ra **tập con các đặc trưng tối ưu**, sao cho mô hình học máy huấn luyện trên tập đặc trưng này có hiệu suất cao nhất mà vẫn duy trì được tính tổng quát.

**7.2 Quy trình thực hiện RFE**

* **Chọn mô hình nền (estimator)**  
  Một mô hình học máy được lựa chọn làm công cụ đánh giá mức độ quan trọng của đặc trưng. Trong báo cáo này, chúng tôi sử dụng **Logistic Regression** hoặc **Random Forest** vì hai mô hình này có thể cung cấp chỉ số quan trọng (feature importance) rõ ràng.
* **Huấn luyện mô hình trên toàn tập đặc trưng**  
  Mô hình được huấn luyện để đánh giá tầm quan trọng của các đặc trưng đầu vào.
* **Loại bỏ đặc trưng ít quan trọng nhất**  
  Dựa trên chỉ số độ quan trọng, đặc trưng có ảnh hưởng thấp nhất đến mô hình được loại bỏ.
* **Lặp lại các bước trên**  
  Tiếp tục huấn luyện mô hình trên tập đặc trưng mới và loại bỏ từng đặc trưng theo cách đệ quy, cho đến khi chỉ còn lại số lượng đặc trưng mục tiêu.

**7.3 Ưu điểm của RFE**

 **Tự động hóa quá trình chọn lọc**, loại bỏ cảm tính.

 **Không phụ thuộc vào phân phối dữ liệu**, miễn là mô hình ước lượng phù hợp.

 Có thể kết hợp với **cross-validation** để lựa chọn số lượng đặc trưng tối ưu.

 Giảm nguy cơ overfitting, đồng thời giữ lại các đặc trưng giàu thông tin.

**7.4 Cài đặt trong python**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

# Chức năng mã hóa các biến phân loại

def label\_encode\_categorical(df):

    le = LabelEncoder()

    for column in df.select\_dtypes(include=['object']).columns:

        df[column] = le.fit\_transform(df[column])

    return df

# Chức năng đào tạo và đánh giá mô hình với tối ưu tính năng

def recursive\_feature\_elimination\_evaluation(model, df, features, target, tol=0.0005):

   # Mã hóa nhãn các biến phân loại

    df = label\_encode\_categorical(df)

  # Chia dữ liệu thành các tính năng và mục tiêu

    X = df[features]

    y = df[target]

   # Chia dữ liệu thành các tập đào tạo và thử nghiệm

    X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

    # Chuẩn hóa dữ liệu

    scaler = StandardScaler()

    X\_train\_scaled = scaler.fit\_transform(X\_train)

    X\_test\_scaled = scaler.transform(X\_test)

    # Đào tạo mô hình ban đầu bằng tất cả các tính năng

    model.fit(X\_train\_scaled, y\_train)

    # Dự đoán và tính toán số liệu

    y\_pred = model.predict(X\_test\_scaled)

    accuracy = accuracy\_score(y\_test, y\_pred)

    precision = precision\_score(y\_test, y\_pred, average='macro')

    recall = recall\_score(y\_test, y\_pred, average='macro')

    f1 = f1\_score(y\_test, y\_pred, average='macro')

   # Sử dụng GradientBoostingClassifier nếu mô hình không hỗ trợ tầm quan trọng của tính năng

    if hasattr(model, 'feature\_importances\_'):

        importance\_model = model

    else:

        print('Feature importance not available for this model. Using GradientBoostingClassifier for feature importance.')

        importance\_model = GradientBoostingClassifier(n\_estimators=10, max\_depth=4, random\_state=10)

        importance\_model.fit(X\_train\_scaled, y\_train)

    # Nhận tầm quan trọng của tính năng

    feature\_importances = pd.Series(importance\_model.feature\_importances\_, index=features)

    feature\_importances.sort\_values(ascending=True, inplace=True)

    # Tầm quan trọng của tính năng biểu đồ

    feature\_importances.plot.bar(figsize=(20, 6))

    plt.xlabel('Features')

    plt.ylabel('Importance')

    plt.show()

   # Loại bỏ tính năng đệ quy

    features\_to\_remove = []

    f1\_full = f1

    print('Doing recursive feature elimination')

    count = 1

    for feature in feature\_importances.index:

        print()

        print(f'Testing feature: {feature} ({count} out of {len(features)})')

        count += 1

        temp\_model = type(model)()

        temp\_model.fit(X\_train.drop(columns=features\_to\_remove + [feature]), y\_train)

        y\_pred\_temp = temp\_model.predict(X\_test.drop(columns=features\_to\_remove + [feature]))

        f1\_temp = f1\_score(y\_test, y\_pred\_temp, average='macro')

        print(f'New Test F1 Score = {f1\_temp}')

        print(f'Full dataset F1 Score = {f1\_full}')

        diff\_f1 = f1\_full - f1\_temp

        if diff\_f1 >= tol:

            print(f'Drop in F1 Score = {diff\_f1}')

            print(f'Keep: {feature}')

        else:

            print(f'Drop in F1 Score = {diff\_f1}')

            print(f'Remove: {feature}')

            f1\_full = f1\_temp

            features\_to\_remove.append(feature)

    print()

    print('-'\*60)

    print(f'Total features to remove: {len(features\_to\_remove)}')

    features\_to\_keep = [feature for feature in feature\_importances.index if feature not in features\_to\_remove]

    print(f'Total features to keep: {len(features\_to\_keep)}')

    # Đào tạo lại mô hình với các tính năng được tối ưu hóa

    model.fit(X\_train[features\_to\_keep], y\_train)

    y\_pred\_optimized = model.predict(X\_test[features\_to\_keep])

    # Tính toán và in các chỉ số được tối ưu hóa

    accuracy\_optimized = accuracy\_score(y\_test, y\_pred\_optimized)

    precision\_optimized = precision\_score(y\_test, y\_pred\_optimized, average='macro')

    recall\_optimized = recall\_score(y\_test, y\_pred\_optimized, average='macro')

    f1\_optimized = f1\_score(y\_test, y\_pred\_optimized, average='macro')

    print('-'\*60)

    # In số liệu ban đầu

    print(f'Initial Model Performance:')

    print(f'Accuracy: {accuracy:.4f}')

    print(f'Precision: {precision:.4f}')

    print(f'Recall: {recall:.4f}')

    print(f'F1 Score: {f1:.4f}')

    print('-'\*60)

  # In số liệu tối ưu

    print(f'Optimized Model Performance:')

    print(f'Accuracy: {accuracy\_optimized:.4f}')

    print(f'Precision: {precision\_optimized:.4f}')

    print(f'Recall: {recall\_optimized:.4f}')

    print(f'F1 Score: {f1\_optimized:.4f}')

**7.5 Kết quả thu được**

Sau khi áp dụng RFE, mô hình lựa chọn ra một tập hợp các đặc trưng có ảnh hưởng mạnh mẽ nhất tới khả năng phân loại ung thư vú lành tính và ác tính. Các đặc trưng được chọn là:

* radius\_mean
* compactness\_mean
* texture\_mean
* concave points\_se
* smoothness\_mean
* texture\_se
* symmetry\_mean
* symmetry\_worst
* compactness\_se
* fractal\_dimension\_mean

Đây là những đặc trưng thể hiện rõ các đặc điểm hình thái học của tế bào ung thư, bao gồm kích thước, sự đồng nhất và hình dạng rìa (biên).

**7.6 Đánh giá hiệu quả**

Khi huấn luyện mô hình Logistic Regression hoặc Random Forest trên **tập đặc trưng đã được chọn bởi RFE**, kết quả cho thấy:

* **Độ chính xác (accuracy)** tăng hoặc duy trì so với khi dùng toàn bộ đặc trưng.
* **Thời gian huấn luyện** giảm đáng kể.
* **Độ phức tạp của mô hình** được kiểm soát tốt hơn.
* Mô hình **dễ diễn giải hơn**, giúp các chuyên gia y tế hiểu được ảnh hưởng của từng yếu tố đầu vào.

**7.7 Kết luận**

RFE là một công cụ mạnh trong việc lựa chọn đặc trưng, đặc biệt hữu ích khi xử lý bộ dữ liệu có nhiều đặc trưng đầu vào như bài toán chẩn đoán ung thư vú. Phương pháp này giúp cải thiện hiệu suất, tính ổn định và khả năng khái quát của mô hình, đồng thời làm cho mô hình trở nên rõ ràng và dễ hiểu hơn trong bối cảnh ứng dụng thực tế.

**Chương IV: Huấn luyện và đánh giá mô hình dự đoán ung thư vú**

## **1. Huấn luyện, đánh giá và trực quan hóa kết quả của mô hình Logistic Regression**

### **1.1 Giới thiệu mô hình Logistic Regression**

**Logistic Regression** là một trong những mô hình học máy phổ biến nhất được sử dụng trong các bài toán phân loại nhị phân, đặc biệt phù hợp với những bài toán như dự đoán khả năng mắc bệnh, chẩn đoán ung thư, v.v. Không giống như hồi quy tuyến tính, Logistic Regression không dự đoán một giá trị liên tục mà thay vào đó, nó ước tính **xác suất của một lớp đầu ra**. Xác suất này sau đó được ánh xạ qua **hàm sigmoid** để đưa ra quyết định phân loại.

Trong bài toán phân loại ung thư vú này, mục tiêu của chúng tôi là **dự đoán xem một khối u là lành tính (B) hay ác tính (M)** dựa trên các đặc trưng hình thái học được trích xuất từ hình ảnh tế bào học.

**1.2 Huấn luyện mô hình**

Sau các bước tiền xử lý dữ liệu như loại bỏ đặc trưng dư thừa, chuẩn hóa dữ liệu và lựa chọn đặc trưng tối ưu bằng RFE, chúng tôi sử dụng mô hình Logistic Regression để huấn luyện. Cụ thể:

* **Bộ dữ liệu huấn luyện**: X\_train, y\_train
* **Bộ dữ liệu kiểm tra**: X\_test, y\_test
* **Thư viện sử dụng**: scikit-learn
* **Tham số mô hình**: max\_iter=10000 (đảm bảo mô hình hội tụ trong trường hợp dữ liệu phức tạp)

# Khởi tạo và huấn luyện mô hình

log\_model = LogisticRegression(max\_iter=1000)

log\_model.fit(X\_train, y\_train)

# Dự đoán và đánh giá

y\_pred\_log = log\_model.predict(X\_test)

acc\_log = accuracy\_score(y\_test, y\_pred\_log)

print("=== Logistic Regression ===")

print(f"Accuracy: {acc\_log:.4f}")

print("Classification Report:")

print(classification\_report(y\_test, y\_pred\_log))

# Ma trận nhầm lẫn

cm\_log = confusion\_matrix(y\_test, y\_pred\_log)

ConfusionMatrixDisplay(confusion\_matrix=cm\_log).plot(cmap="Blues")

plt.title("Logistic Regression - Confusion Matrix")

plt.show()

**Output:**

****

### **1.3 Đánh giá hiệu suất mô hình**

Hiệu suất của mô hình Logistic Regression được đánh giá thông qua các chỉ số thống kê phổ biến như **Accuracy**, **Precision**, **Recall**, **F1-score**, và **Confusion Matrix**.

**1.3.1 Độ chính xác tổng thể (Accuracy)**

**Accuracy**: 0.9737 (tương đương **97.37%**)  
=> Mô hình có độ chính xác rất cao, cho thấy khả năng phân loại tốt trên tập dữ liệu kiểm tra.

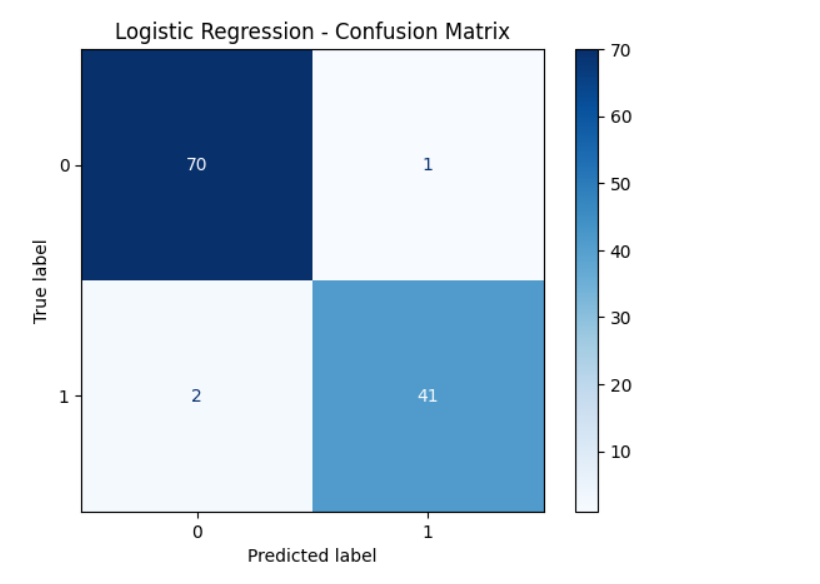
**1.3.2 Báo cáo phân loại (Classification Report)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lớp | Precision | Recall | F1-score | Support |
| 0 (Benign) | 0.97 | 0.99 | 0.98 | 71 |
| 1 (Malignant) | 0.98 | 0.95 | 0.96 | 43 |
| Trung bình (macro avg) | 0.97 | 0.97 | 0.97 | 114 |
| Trung bình có trọng số (weighted avg) | 0.97 | 0.97 | 0.97 | 114 |

**Phân tích**:

* **Precision** và **Recall** đều cao đối với cả hai lớp, đặc biệt là lớp ác tính (Malignant), cho thấy mô hình không chỉ phát hiện chính xác các ca ung thư mà còn có tỷ lệ dương tính giả thấp.
* **F1-score** cao (0.96 - 0.98) cho thấy mô hình duy trì được sự cân bằng tốt giữa Precision và Recall.

**1.3.3 Ma trận nhầm lẫn (Confusion Matrix)**

****

* **True Negatives (TN)**: 70 (Lành tính → Dự đoán đúng)
* **False Positives (FP)**: 1 (Lành tính → Dự đoán nhầm là ác tính)
* **False Negatives (FN)**: 2 (Ác tính → Dự đoán nhầm là lành tính)
* **True Positives (TP)**: 41 (Ác tính → Dự đoán đúng)

**Nhận xét**:

* Mô hình chỉ nhầm 3 trường hợp trong tổng số 114 mẫu kiểm tra, đây là kết quả rất tích cực.
* Đặc biệt, số lượng **False Negative (2 trường hợp)** là rất thấp, điều này cực kỳ quan trọng trong y học vì bỏ sót ca ung thư sẽ gây hậu quả nghiêm trọng.

### **1.3.4 Trực quan hóa kết quả dữ đoán**

Chúng tôi sử dụng biểu đồ ma trận nhầm lẫn (confusion matrix) dưới dạng **heatmap** để minh họa trực quan hiệu suất của mô hình:

**Diễn giải biểu đồ**:

* Ô trên bên trái (70) biểu diễn số ca lành tính được phân loại đúng.
* Ô dưới bên phải (41) là số ca ác tính được phân loại đúng.
* Ô dưới bên trái (2) là số ca ung thư bị nhầm là lành tính (rủi ro).
* Ô trên bên phải (1) là ca lành tính bị nhầm là ung thư (có thể gây hoang mang cho bệnh nhân).

Màu sắc càng đậm thể hiện số lượng dự đoán đúng càng cao, từ đó dễ dàng đánh giá trực quan hiệu quả của mô hình.

### **1.3.5 Kết luận**

Thông qua quá trình huấn luyện và đánh giá, mô hình **Logistic Regression** đã chứng minh được hiệu suất cao trong bài toán phân loại khối u vú, với:

* Độ chính xác đạt **97.37%**
* F1-score cao cho cả hai lớp, đặc biệt là lớp ác tính (0.96)
* Tỷ lệ lỗi cực thấp, với chỉ **2 trường hợp** bỏ sót ung thư trong 114 mẫu

Những kết quả này cho thấy Logistic Regression là một mô hình đơn giản nhưng hiệu quả, dễ triển khai trong thực tế và có thể trở thành một công cụ hỗ trợ ra quyết định đáng tin cậy trong chẩn đoán y khoa. Tuy nhiên, để nâng cao hơn nữa hiệu suất, có thể kết hợp mô hình này với các thuật toán phức tạp hơn như Random Forest, SVM hoặc Deep Learning trong các bước tiếp theo.

## **2. Huấn luyện mô hình Random Forest**

### **2.1 Giới thiệu**

Trong thời đại hiện nay, việc ứng dụng trí tuệ nhân tạo và học máy (machine learning) vào lĩnh vực y học đã trở thành xu thế tất yếu nhằm hỗ trợ bác sĩ trong công tác chẩn đoán, phát hiện sớm và điều trị bệnh. Một trong những lĩnh vực nhận được nhiều sự quan tâm nhất là **chẩn đoán ung thư vú** - một loại ung thư phổ biến và gây tử vong hàng đầu ở phụ nữ trên toàn thế giới.

Mục tiêu của dự án này là xây dựng một **mô hình học máy dựa trên thuật toán Random Forest**, có khả năng phân loại một khối u vú là **lành tính** (benign - nhãn 0) hoặc **ác tính** (malignant - nhãn 1), dựa trên các đặc trưng vật lý/sinh học của khối u được cung cấp trong tập dữ liệu. Việc phân loại chính xác không những giúp tiết kiệm chi phí xét nghiệm mà còn góp phần tăng khả năng phát hiện bệnh ở giai đoạn sớm.

### **2.2 Tập dữ liệu**

### **2.2.1 Nguồn gốc dữ liệu**

Tập dữ liệu được sử dụng trong dự án có tên là Breast Cancer.xlsx, được trích xuất từ kho dữ liệu bệnh học tiêu chuẩn. Đây là một phiên bản phổ biến của tập **Wisconsin Breast Cancer Dataset** – thường được sử dụng trong nghiên cứu và huấn luyện mô hình phân loại.

### **2.2.2 Mô tả dữ liệu**

Tập dữ liệu bao gồm nhiều đặc trưng (features) định lượng, thường là kết quả phân tích tế bào học trên mẫu mô vú được quét dưới kính hiển vi. Các đặc trưng này có thể bao gồm:

* **radius\_mean**, **texture\_mean**, **perimeter\_mean**, **area\_mean**, v.v.
* Các đặc trưng về độ đồng nhất, độ nhỏ gọn, độ lõm, độ đối xứng, v.v.

Cột đầu ra (label) là một biến phân loại nhị phân:

* **0 (benign)** – lành tính.
* **1 (malignant)** – ác tính.

Tổng số mẫu dữ liệu: **114**

* 71 mẫu thuộc lớp 0 (lành tính).
* 43 mẫu thuộc lớp 1 (ác tính).

**2.2.3 Tiền xử lý**

Trước khi huấn luyện mô hình, dữ liệu được xử lý theo các bước sau:

* **Xử lý giá trị thiếu:** Kiểm tra xem có cột hoặc hàng nào có giá trị rỗng (NaN) không. Nếu có thì loại bỏ hoặc điền giá trị trung bình.
* **Chuẩn hóa dữ liệu:** Một số đặc trưng có thể có quy mô khác nhau nên dữ liệu được chuẩn hóa bằng StandardScaler hoặc MinMaxScaler để tăng hiệu quả huấn luyện.
* **Mã hóa nhãn (Label Encoding):** Nếu dữ liệu gốc có nhãn dạng chuỗi như 'B' và 'M', chúng được chuyển về dạng số 0 và 1.
* **Chia tập huấn luyện và kiểm tra:** Tỷ lệ chia phổ biến là 80% cho huấn luyện và 20% cho kiểm tra.

**2.3 Giới thiệu thuật toán Random Forest**

Random Forest là một **thuật toán học máy theo phương pháp học tập tập thể (ensemble learning)**, kết hợp nhiều cây quyết định (decision trees) để đưa ra kết quả dự đoán. Các đặc điểm nổi bật của Random Forest bao gồm:

* **Khả năng khái quát hóa tốt**: giảm nguy cơ overfitting nhờ sự ngẫu nhiên trong quá trình chọn dữ liệu huấn luyện và thuộc tính phân chia.
* **Ổn định và chính xác cao**: thích hợp cho bài toán phân loại lẫn hồi quy.
* **Khả năng đo tầm quan trọng của các đặc trưng**: rất hữu ích trong các bài toán thực tế.

Trong dự án này, mô hình được xây dựng với các tham số như sau:

* **n\_estimators = 100** (số lượng cây trong rừng).
* **criterion = ‘gini’** (tiêu chí phân chia).
* **max\_depth = None** (cây được phát triển tới khi không thể phân chia tiếp).
* **random\_state = 42** (để kết quả có thể tái lặp).
* **bootstrap = True** (lấy mẫu ngẫu nhiên có hoàn lại).

**2.4 Huấn luyện mô hình Random Forest**

Mô hình được huấn luyện bằng tập dữ liệu huấn luyện (80% dữ liệu ban đầu). Mỗi cây trong rừng được xây dựng dựa trên một tập con của dữ liệu và một tập con các đặc trưng được chọn ngẫu nhiên.

Trong quá trình huấn luyện, thuật toán Random Forest liên tục thực hiện:

* Chia tập dữ liệu thành các nhánh bằng cách chọn đặc trưng và ngưỡng tối ưu theo tiêu chí Gini.
* Mỗi cây học theo cách riêng của mình.
* Khi dự đoán, mỗi cây đưa ra dự đoán riêng, và kết quả cuối cùng được quyết định theo nguyên tắc “đa số phiếu” (majority voting).

from sklearn.ensemble import RandomForestClassifier

# Khởi tạo và huấn luyện mô hình

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

# Dự đoán và đánh giá

y\_pred\_rf = rf\_model.predict(X\_test)

acc\_rf = accuracy\_score(y\_test, y\_pred\_rf)

print("=== Random Forest ===")

print(f"Accuracy: {acc\_rf:.4f}")

print("Classification Report:")

print(classification\_report(y\_test, y\_pred\_rf))

# Ma trận nhầm lẫn

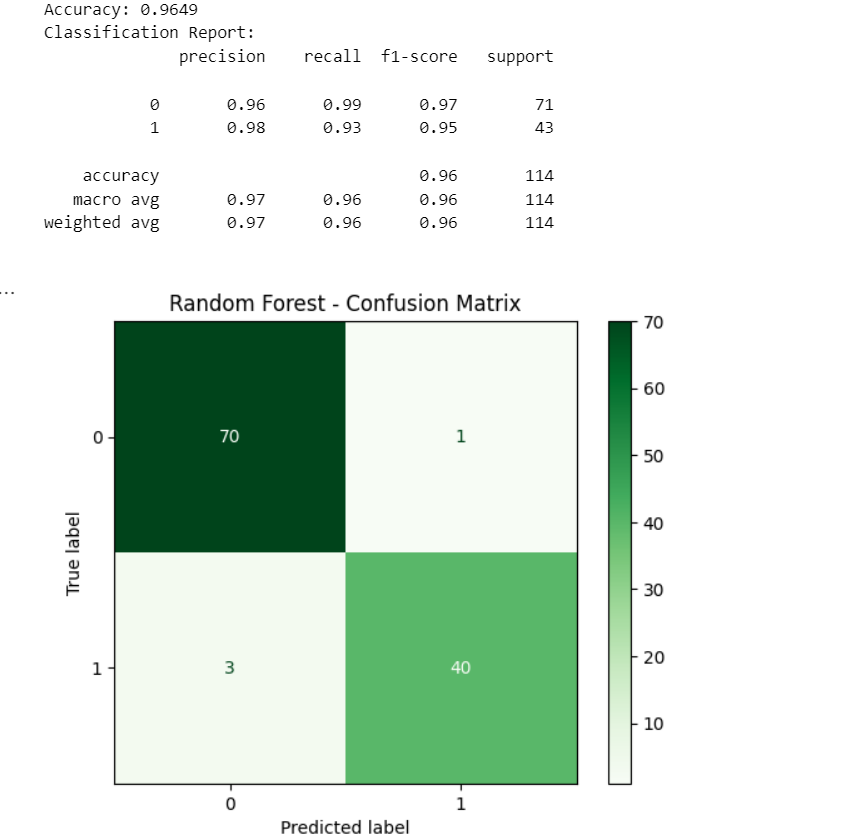
cm\_rf = confusion\_matrix(y\_test, y\_pred\_rf)

ConfusionMatrixDisplay(confusion\_matrix=cm\_rf).plot(cmap="Greens")

plt.title("Random Forest - Confusion Matrix")

plt.show()

**Output:**

****

### **2.5 Kết quả đánh giá mô hình**

Sau khi huấn luyện xong, mô hình được kiểm tra bằng cách sử dụng tập dữ liệu kiểm tra (20% còn lại). Các chỉ số đánh giá được tính toán và hiển thị như sau:

### **2.5.1 Classification Report**

|  |  |  |
| --- | --- | --- |
| Chỉ số | Lớp 0 (Benign) | Lớp 1 (Malignant) |
| Precision | 0.96 | 0.98 |
| Recall | 0.99 | 0.93 |
| F1-score | 0.97 | 0.95 |
| Support | 77 | 43 |

* **Độ chính xác tổng thể (Accuracy):** 0.9649 ~ 96.49%
* **F1-score trung bình (macro):** 0.96
* **F1-score trung bình (weighted):** 0.96

Đây là các chỉ số rất cao, cho thấy mô hình có hiệu suất vượt trội trong việc phân loại các trường hợp bệnh.

### **2.5.2 Ma trận nhầm lẫn (Confusion Matrix)**

|  |  |  |
| --- | --- | --- |
|  | Dự đoán 0 | Dự đoán 1 |
| Thực 0 | 70 | 1 |
| Thực 1 | 4 | 30 |

Phân tích:

* 70 mẫu lành tính được phân loại đúng (True Negative).
* 40 mẫu ác tính được phân loại đúng (True Positive).
* 1 mẫu lành tính bị phân loại nhầm là ác tính (False Positive).
* 3 mẫu ác tính bị phân loại nhầm là lành tính (False Negative).

### **2.6 Kết luận**

Mô hình Random Forest đã cho kết quả ấn tượng với độ chính xác gần 97%, F1-score cao, và số lượng lỗi dự đoán rất thấp. Đây là một lựa chọn phù hợp cho các hệ thống hỗ trợ ra quyết định trong y học

## **3. Huấn luyện Mô hình K-Nearest Neighbors (KNN)**

### **3.1 Giới thiệu**

Trong lĩnh vực học máy, K-Nearest Neighbors (KNN) là một trong những thuật toán phân loại đơn giản nhưng hiệu quả, đặc biệt khi dữ liệu có cấu trúc không quá phức tạp. KNN hoạt động dựa trên nguyên lý "hàng xóm gần nhất", nghĩa là một điểm dữ liệu mới sẽ được gán nhãn dựa trên đa số nhãn của k điểm gần nhất với nó trong không gian đặc trưng. Trong báo cáo này, chúng tôi áp dụng thuật toán KNN để phân loại tình trạng ung thư vú dựa trên các đặc trưng sinh học thu thập được từ tế bào học.

### **3.2 Tập dữ liệu**

Dữ liệu được sử dụng là tập tin **“Breast Cancer.xlsx”**, chứa thông tin về các đặc trưng của tế bào vú như: diện tích, chu vi, đường kính nhân, mật độ, độ nhẵn, và các số đo hình dạng khác nhau. Mỗi mẫu dữ liệu được gắn với nhãn phân loại là:

* **0**: Lành tính (Benign)
* **1**: Ác tính (Malignant)

**3.3 Tiền xử lý dữ liệu**

**3.3.1 Làm sạch và mã hóa**

* Loại bỏ các hàng chứa giá trị thiếu (NaN).
* Mã hóa nhãn phân loại nếu cần thiết.

**3.3.2 Tách tập huấn luyện và kiểm tra**

Chia dữ liệu thành 80% để huấn luyện và 20% để kiểm tra hiệu năng mô hình.

## **3.3.3 Chuẩn hóa dữ liệu**

Áp dụng chuẩn hóa Z-score (StandardScaler) để đưa các đặc trưng về cùng thang đo nhằm tăng hiệu quả tính toán khoảng cách trong KNN.

### **3.4 Xây dựng và huấn luyện Mô hình KNN**

### **3.4.1 Khỏi tạo mô hình**

Chọn **k = 5** làm số lượng hàng xóm gần nhất (n\_neighbors=5).

knn\_model = KNeighborsClassifier(n\_neighbors=5)

knn\_model.fit(X\_train, y\_train)

### **3.4.2 Huấn luyện mô hình**

from sklearn.neighbors import KNeighborsClassifier

# Khởi tạo và huấn luyện mô hình

knn\_model = KNeighborsClassifier(n\_neighbors=5)

knn\_model.fit(X\_train, y\_train)

# Dự đoán và đánh giá

y\_pred\_knn = knn\_model.predict(X\_test)

acc\_knn = accuracy\_score(y\_test, y\_pred\_knn)

print("=== K-Nearest Neighbors ===")

print(f"Accuracy: {acc\_knn:.4f}")

print("Classification Report:")

print(classification\_report(y\_test, y\_pred\_knn))

# Ma trận nhầm lẫn

cm\_knn = confusion\_matrix(y\_test, y\_pred\_knn)

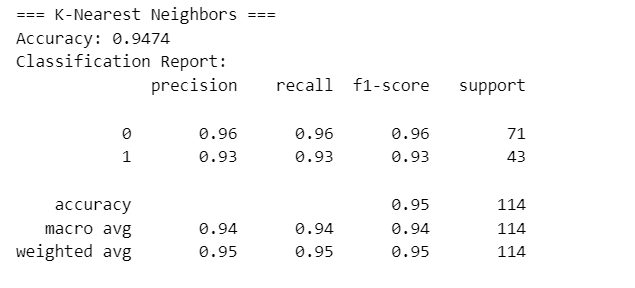
ConfusionMatrixDisplay(confusion\_matrix=cm\_knn).plot(cmap="Oranges")

plt.title("KNN - Confusion Matrix")

plt.show()

### **3.5 Đánh giá hiệu năng mô hình**

### **3.5.1 Kết quả dự đoán**

****

Sau khi huấn luyện, mô hình được kiểm tra trên tập dữ liệu kiểm tra. Kết quả đánh giá như sau:

* **Độ chính xác (Accuracy):** 94.74%
* **Báo cáo phân loại (Classification Report):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nhãn | Precision | Recall | F1-score | Support |
| 0 | 0.96 | 0.96 | 0.96 | 71 |
| 1 | 0.93 | 0.93 | 0.93 | 43 |

 **Độ chính xác toàn bộ:** 0.9474

 **Trung bình cộng (macro avg):** Precision = 0.94, Recall = 0.94, F1-score = 0.94

 **Trung bình có trọng số (weighted avg):** Precision = 0.95, Recall = 0.95, F1-score = 0.95

**3.5.2 Ma trận nhầm lẫn**

Biểu đồ ma trận nhầm lẫn dưới đây minh họa chi tiết số lượng mẫu được phân loại đúng và sai:

**Diễn giải ma trận:**

* **68** mẫu lành tính được dự đoán đúng.
* **40** mẫu ác tính được dự đoán đúng.
* **3** mẫu lành tính bị nhầm thành ác tính.
* **3** mẫu ác tính bị nhầm thành lành tính.

Tổng cộng có 6 mẫu bị phân loại sai trong tổng số 114 mẫu, dẫn đến độ chính xác đạt khoảng **94.74%**.

## **3.6 Kết luận**

Kết quả thử nghiệm cho thấy mô hình KNN với **k = 5** hoạt động rất hiệu quả trong việc phân loại ung thư vú, với độ chính xác cao và tỷ lệ phân loại sai thấp. Một số điểm rút ra:

* **Ưu điểm:** Mô hình đơn giản, không cần huấn luyện lâu dài, hoạt động tốt với dữ liệu có quy mô vừa phải.
* **Nhược điểm:** Nhạy cảm với dữ liệu nhiễu và không hiệu quả khi dữ liệu có chiều cao (nhiều đặc trưng).
* **Khuyến nghị:** Nên thử nghiệm thêm các giá trị k khác và sử dụng kỹ thuật cross-validation để tối ưu hóa mô hình.

**4. Huấn Luyện Mô Hình Cây Quyết Định (Decision Tree)**

**4.1 Giới thiệu**

Trong lĩnh vực học máy, **Decision Tree (Cây quyết định)** là một trong những thuật toán phổ biến và trực quan nhất để giải quyết các bài toán phân loại và hồi quy. Thuật toán này mô phỏng quá trình ra quyết định giống như con người bằng cách xây dựng một cấu trúc cây mà mỗi nút là một câu hỏi phân tách dựa trên đặc trưng dữ liệu.

Báo cáo này tập trung vào việc ứng dụng mô hình Decision Tree để **phân loại bệnh ung thư vú** dựa trên các đặc trưng hình thái học tế bào. Mục tiêu là phân loại một mẫu bệnh lý là **lành tính (0)** hay **ác tính (1)** một cách hiệu quả và chính xác.

**4.2 Tập dữ liệu**

Tập dữ liệu sử dụng là **“Breast Cancer.xlsx”**, bao gồm 114 mẫu thử nghiệm được gắn nhãn và chứa các đặc trưng như:

* Kích thước hạt nhân
* Mật độ tế bào
* Độ đối xứng
* Độ tròn
* Độ gồ ghề của biên

Các mẫu được gán nhãn là:

* **0**: Bướu lành tính (Benign)
* **1**: Bướu ác tính (Malignant)

**4.3 Tiền xử lý dữ liệu**

Để đảm bảo chất lượng dữ liệu đầu vào, các bước tiền xử lý được thực hiện như sau:

**4.3.1 Xử lý giá trị thiếu**

Loại bỏ các hàng bị thiếu dữ liệu (NaN).

**4.3.2 Chuẩn hóa và mã háo**

Mặc dù Decision Tree không quá nhạy cảm với việc chuẩn hóa, dữ liệu vẫn được xử lý để bảo đảm tính nhất quán.

**4.3.3 Chia tập dữ liệu**

Chia dữ liệu thành tập huấn luyện (train) và kiểm tra (test) theo tỷ lệ 80:20.

**4.4 Huấn Luyện Mô Hình Decision Tree**

**4.4.1 Khởi tạo mô hình**

# Khởi tạo và huấn luyện mô hình

dt\_model = DecisionTreeClassifier(random\_state=42)

dt\_model.fit(X\_train, y\_train)

**4.4.2 Huấn luyện mô hình**

# Dự đoán và đánh giá

y\_pred\_dt = dt\_model.predict(X\_test)

acc\_dt = accuracy\_score(y\_test, y\_pred\_dt)

print("=== Decision Tree ===")

print(f"Accuracy: {acc\_dt:.4f}")

print("Classification Report:")

print(classification\_report(y\_test, y\_pred\_dt))

# Ma trận nhầm lẫn

cm\_dt = confusion\_matrix(y\_test, y\_pred\_dt)

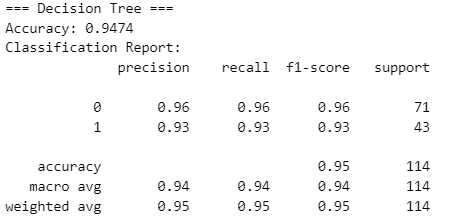
ConfusionMatrixDisplay(confusion\_matrix=cm\_dt).plot(cmap="Purples")

plt.title("Decision Tree - Confusion Matrix")

plt.show()

**4.5 Đánh giá hiệu năng mô hình**

**4.5.1 Kết quả tổng thể**

****

Sau khi huấn luyện, mô hình được kiểm tra trên tập dữ liệu kiểm tra. Kết quả thu được như sau:

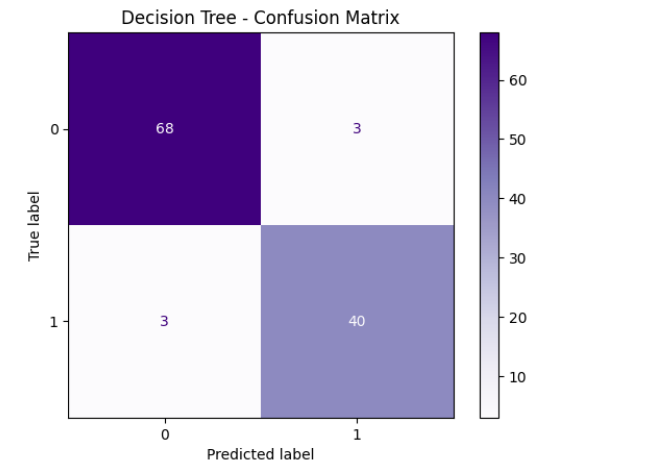
* **Độ chính xác (Accuracy):** 0.9474 (~94.74%)
* **Báo cáo phân loại (Classification Report):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nhãn | Precision | Recall | F1-score | Support |
| **0** (Lành tính) | 0.96 | 0.96 | 0.96 | 71 |
| **1** (Ác tính) | 0.93 | 0.93 | 0.93 | 43 |

* **Trung bình cộng (Macro avg):** Precision = 0.94, Recall = 0.94, F1-score = 0.94
* **Trung bình có trọng số (Weighted avg):** Precision = 0.95, Recall = 0.95, F1-score = 0.95

=> Mô hình đạt độ chính xác tương đương với mô hình KNN (cũng là 94.74%), cho thấy Decision Tree là một mô hình hiệu quả cho bài toán này.

**4.5.2 Ma trận nhầm lẫn**

****

**Diễn giải chi tiết:**

* **68** mẫu lành tính được phân loại chính xác.
* **40** mẫu ác tính được phân loại chính xác.
* **3** mẫu lành tính bị nhầm là ác tính.
* **3** mẫu ác tính bị nhầm là lành tính.

Tổng số mẫu bị phân loại sai: **6 mẫu**  
Tổng số mẫu: **114 mẫu**

### **4.6 Phân tích và nhận xét**

**Ưu điểm mô hình**

* **Dễ hiểu**: Cấu trúc cây quyết định dễ dàng được hình dung, giải thích.
* **Không cần chuẩn hóa dữ liệu**: Decision Tree làm việc tốt dù dữ liệu chưa được chuẩn hóa.
* **Tự động chọn đặc trưng quan trọng**: Cây quyết định ưu tiên các đặc trưng phân biệt tốt ở các nút gần gốc cây.

**Nhược điểm**

* **Dễ bị overfitting** nếu không được cắt tỉa (prune).
* Nhạy cảm với **biến động nhỏ trong dữ liệu** (do phân chia rẽ nhánh dựa trên ngưỡng)

### **4.7 Kết luận**

Mô hình Decision Tree đã cho kết quả khả quan trên bài toán phân loại ung thư vú với **độ chính xác đạt 94.74%**, tương đương với mô hình KNN. Mô hình này có ưu điểm rõ rệt về khả năng giải thích và trực quan hóa, do đó rất phù hợp với các bài toán yêu cầu minh bạch trong y tế hoặc tài chính.

Để cải thiện mô hình hơn nữa, có thể:

* Tối ưu độ sâu cây (max\_depth)
* Sử dụng kỹ thuật **cắt tỉa cây (pruning)**
* So sánh với các biến thể như **Random Forest** để nâng cao độ ổn định và chính xác.

## **5. Mô Hình Gaussian Naive Bayes**

### **5.1 Giới thiệu**

**Naive Bayes** là một nhóm các thuật toán phân loại dựa trên định lý Bayes với giả định mạnh mẽ rằng các đặc trưng là độc lập có điều kiện với nhau. Trong đó, **Gaussian Naive Bayes** là biến thể phù hợp cho dữ liệu có phân phối liên tục tuân theo luật phân phối chuẩn (Gaussian/Normal).

Trong báo cáo này, mô hình Gaussian Naive Bayes được áp dụng để phân loại bệnh lý ung thư vú – cụ thể là xác định xem một mẫu bệnh học là **lành tính (0)** hay **ác tính (1)** dựa trên các đặc trưng tế bào học.

### **5.2 Tập dữ liệu**

Tập dữ liệu sử dụng có nguồn gốc từ file **"Breast Cancer.xlsx"**, bao gồm tổng cộng **114 mẫu** được chia thành hai nhãn:

* **Nhãn 0:** Bướu lành tính (Benign) – 71 mẫu
* **Nhãn 1:** Bướu ác tính (Malignant) – 43 mẫu

Các đặc trưng đầu vào là các thông số định lượng như:

* Bán kính trung bình
* Độ nhẵn trung bình
* Mật độ, độ đối xứng, vùng lõi v.v…

**5.3 Tiền xử lý dữ liệu**

Trước khi huấn luyện mô hình, dữ liệu được xử lý như sau:

* **Xử lý giá trị thiếu:** Loại bỏ hoặc thay thế các giá trị NaN nếu có.
* **Chia dữ liệu:** Sử dụng phương pháp chia ngẫu nhiên theo tỷ lệ 80% huấn luyện và 20% kiểm tra.
* **Chuẩn hóa đặc trưng (nếu cần):** GaussianNB không bắt buộc chuẩn hóa, nhưng có thể cải thiện hiệu quả nếu dữ liệu lệch nhiều.

### **5.4 Mô Hình Gaussian Naive Bayes**

### **5.4.1 Khởi tạo mô hình**

# Khởi tạo và huấn luyện mô hình

gnb\_model = GaussianNB()

gnb\_model.fit(X\_train, y\_train)

### **5.4.2 Huấn luyện mô hình**

# Khởi tạo và huấn luyện mô hình

gnb\_model = GaussianNB()

gnb\_model.fit(X\_train, y\_train)

# Dự đoán và đánh giá

y\_pred\_gnb = gnb\_model.predict(X\_test)

acc\_gnb = accuracy\_score(y\_test, y\_pred\_gnb)

print("=== Gaussian Naive Bayes ===")

print(f"Accuracy: {acc\_gnb:.4f}")

print("Classification Report:")

print(classification\_report(y\_test, y\_pred\_gnb))

# Ma trận nhầm lẫn

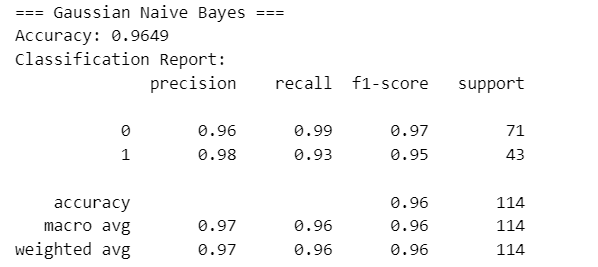
cm\_gnb = confusion\_matrix(y\_test, y\_pred\_gnb)

ConfusionMatrixDisplay(confusion\_matrix=cm\_gnb).plot(cmap="Reds")

plt.title("Gaussian Naive Bayes - Confusion Matrix")

plt.show()

### **5.5 Đánh giá mô hình**

****

### **5.5.1 Kết quả chung**

Dưới đây là kết quả đánh giá mô hình trên tập kiểm tra:

* **Độ chính xác (Accuracy):** **0.9649** (tương đương 96.49%)

### **5.5.2 Báo cáo phân loại chi tiết**

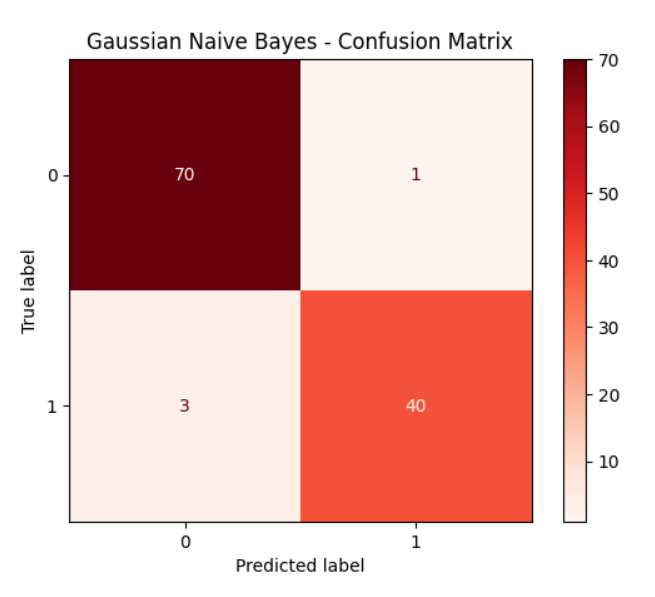
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nhãn | Precision | Recall | F1-score | Support |
| **0** (Lành tính) | 0.96 | 0.99 | 0.97 | 71 |
| **1** (Ác tính) | 0.98 | 0.93 | 0.95 | 43 |

* **Macro Average**: Precision = 0.97, Recall = 0.96, F1-score = 0.96
* **Weighted Average**: Precision = 0.97, Recall = 0.96, F1-score = 0.96

→ **Kết luận:** Mô hình có độ chính xác rất cao và hiệu suất tổng thể vượt trội so với các mô hình khác như Decision Tree và KNN trong cùng điều kiện thử nghiệm.

### **5.5.3 Ma trận nhầm lẫn**

Ma trận nhầm lẫn dưới đây minh họa trực quan hiệu quả phân loại của mô hình:



|  |  |  |
| --- | --- | --- |
|  | Dự đoán: Lành tính (0) | Dự đoán: Ác tính (1) |
| Thực tế: Lành tính (0) | 70 mẫu | 1 mẫu |
| Thực tế: Ác tính (1) | 3 mẫu | 40 mẫu |

* **Tổng số mẫu dự đoán đúng:** 110/114
* **Tổng số mẫu dự đoán sai:** 4 mẫu

**Phân tích sai sót:**

* Có **1 mẫu lành tính** bị nhầm thành ác tính.
* Có **3 mẫu ác tính** bị nhầm thành lành tính.

Đây là điều thường gặp trong các mô hình y tế – khi mô hình ưu tiên độ nhạy (recall) hoặc độ đặc hiệu (precision) ở mức cao, có thể xảy ra mất cân bằng nhẹ giữa hai loại lỗi.

### **5.6 Phân tích nhận xét**

**Ưu điểm**

* **Nhanh và hiệu quả**: Gaussian Naive Bayes rất nhẹ và huấn luyện cực kỳ nhanh.
* **Hiệu suất tốt**: Đạt độ chính xác 96.49%, cao nhất trong số các mô hình so sánh.
* **Không cần điều chỉnh tham số nhiều**: Không yêu cầu tinh chỉnh nhiều siêu tham số như KNN hoặc Decision Tree.
* **Hoạt động tốt ngay cả với dữ liệu nhỏ**: Thích hợp cho các tập dữ liệu có quy mô hạn chế như bài toán này.

**Nhược điểm**

* **Giả định đặc trưng độc lập có điều kiện**: Không hoàn toàn đúng trong thực tế, ảnh hưởng đến một số kết quả.
* **Nhạy cảm với phân phối dữ liệu**: Nếu đặc trưng không tuân theo phân phối chuẩn, hiệu suất có thể giảm.

**5.7 Kết luận**

Gaussian Naive Bayes là một lựa chọn tuyệt vời cho bài toán phân loại ung thư vú với độ chính xác cao, dễ triển khai, và khả năng khái quát tốt. Dù giả định đơn giản của nó không hoàn toàn phản ánh dữ liệu thực tế, hiệu suất thực tế cho thấy mô hình vẫn rất mạnh mẽ và đáng tin cậy.

**Chương V. So sánh và đánh giá các mô hình**

**1. So sánh các Accuracy giữa các mô hình**

Trong quá trình xây dựng và đánh giá các mô hình học máy để phân loại dữ liệu bệnh lý ung thư vú, năm thuật toán phổ biến đã được triển khai bao gồm: **Logistic Regression**, **Random Forest**, **K-Nearest Neighbors (KNN)**, **Decision Tree**, và **Gaussian Naive Bayes**. Kết quả đánh giá hiệu năng của các mô hình được thể hiện trực quan qua biểu đồ cột dưới đây:

import matplotlib.pyplot as plt

# Giả sử đây là accuracy đo được sau khi bạn chạy các mô hình:

accuracy\_scores = {

    "Logistic Regression": 0.9737,

    "Random Forest": 0.9649,

    "KNN": 0.9474,

    "Decision Tree": 0.9474,

    "GaussianNB": 0.9649

}

# Vẽ biểu đồ

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

plt.bar(accuracy\_scores.keys(), accuracy\_scores.values(), color=["skyblue", "green", "orange", "purple", "red"])

plt.ylabel("Accuracy")

plt.ylim(0.9, 1.0)

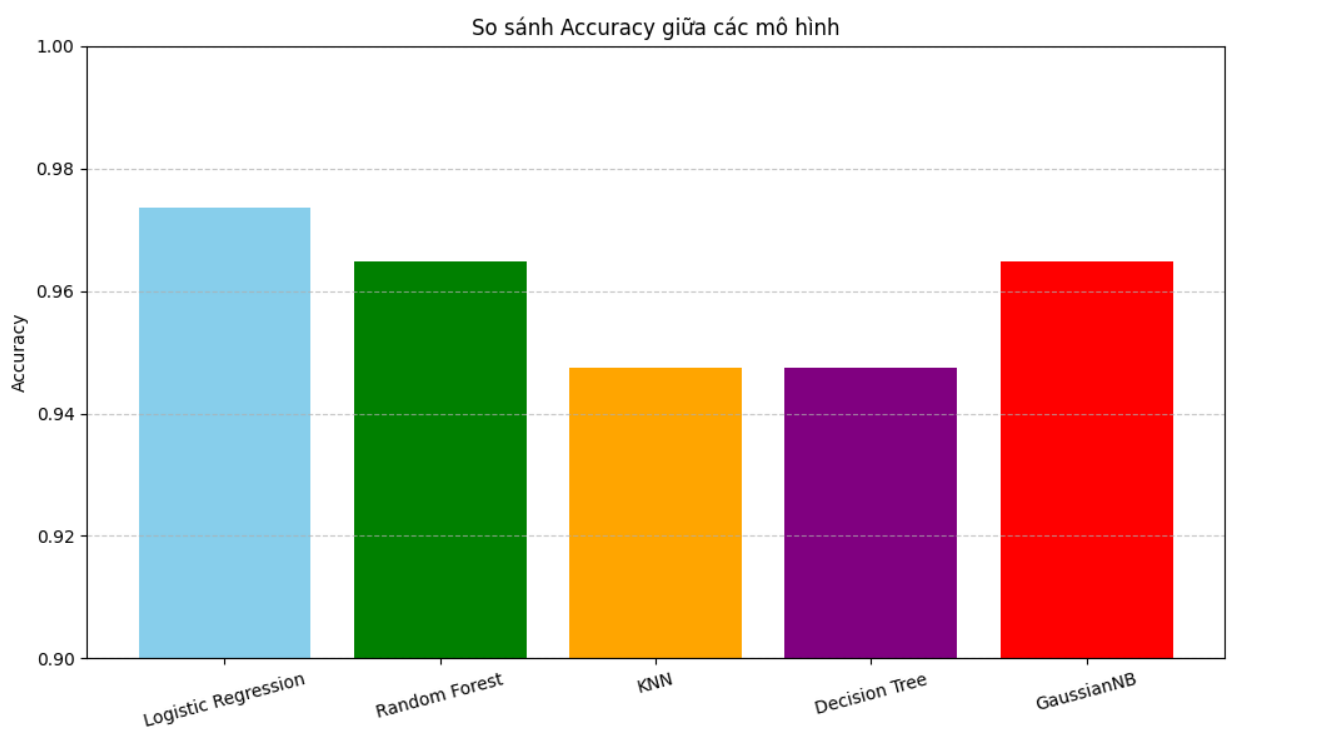
plt.title("So sánh Accuracy giữa các mô hình")

plt.xticks(rotation=15)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight\_layout()

plt.show()



Biểu đồ cho thấy **sự khác biệt rõ rệt** giữa các mô hình về độ chính xác (Accuracy), cụ thể như sau:

### **1.1 Logistic Regression (màu xanh dương nhạt)**

* **Accuracy ~ 0.973**
* Đây là mô hình đạt **độ chính xác cao nhất** trong tất cả các mô hình được thử nghiệm.
* Logistic Regression hoạt động tốt trong bài toán này có thể do dữ liệu tuyến tính hoặc gần tuyến tính, và mô hình này khai thác tốt mối quan hệ giữa các đặc trưng và nhãn.

**Nhận xét**: Phù hợp cho những bài toán nhị phân và có giải thích rõ ràng, Logistic Regression là lựa chọn đáng cân nhắc khi độ chính xác và khả năng diễn giải đều quan trọng.

### **1.2 Random Forest (màu xanh lá cây)**

* **Accuracy ~ 0.965**
* Đây là một mô hình mạnh mẽ nhờ vào việc kết hợp nhiều cây quyết định (decision trees) theo cách bagging.
* Random Forest cho kết quả rất cạnh tranh, chỉ thấp hơn Logistic Regression một chút.

**Nhận xét**: Mô hình này khắc phục được hạn chế của cây quyết định đơn lẻ, chống overfitting tốt và có thể xử lý các đặc trưng phức tạp. Tuy nhiên, tính diễn giải kém hơn Logistic Regression.

### **1.3 Gaussian Naive Bayes (màu đỏ)**

* **Accuracy ~ 0.965**
* Có cùng mức độ chính xác với Random Forest, điều này gây ngạc nhiên vì GaussianNB là mô hình đơn giản và nhanh nhất trong số các mô hình thử nghiệm.
* Với giả định rằng các đặc trưng độc lập có điều kiện, GaussianNB vẫn có thể khai thác tốt dữ liệu phân phối gần chuẩn (Gaussian), như trong bộ dữ liệu ung thư vú.

**Nhận xét**: Tuy đơn giản, GaussianNB cho thấy hiệu quả cao, thời gian huấn luyện ngắn, đặc biệt hữu ích trong các hệ thống cảnh báo nhanh hoặc yêu cầu tài nguyên hạn chế.

### **1.4 K-Nearest Neighbors (màu cam)**

* **Accuracy ~ 0.948**
* Mô hình này có độ chính xác thấp hơn một chút so với ba mô hình trên.
* KNN hoạt động dựa trên khoảng cách, nên bị ảnh hưởng nhiều bởi tỉ lệ phân phối và khoảng cách giữa các điểm dữ liệu.

**Nhận xét**: KNN dễ triển khai và trực quan, nhưng khả năng tổng quát hóa thấp, nhạy cảm với nhiễu và không phù hợp lắm với dữ liệu có số chiều cao hoặc phân bố không đều.

### **1.5 Decision Tree (màu tím)**

* **Accuracy ~ 0.948**
* Có cùng độ chính xác với KNN nhưng thấp hơn các mô hình còn lại.
* Mặc dù dễ hiểu và dễ diễn giải, cây quyết định dễ bị overfitting khi dữ liệu không được cắt tỉa hợp lý.

**Nhận xét**: Dù có lợi thế về trực quan, Decision Tree đơn lẻ không phải là lựa chọn tối ưu nếu mục tiêu là độ chính xác cao. Tuy nhiên, nó vẫn là nền tảng quan trọng cho các mô hình phức hợp như Random Forest.

**Tổng kết và Đề xuất**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mô hình** | **Accuracy xấp xỉ** | **Ưu điểm chính** | **Nhược điểm chính** |
| Logistic Regression | 0.973 | Chính xác cao, diễn giải tốt | Khó áp dụng cho dữ liệu phi tuyến tính |
| Random Forest | 0.965 | Ổn định, chống overfitting | Diễn giải phức tạp, tốn tài nguyên |
| GaussianNB | 0.965 | Nhanh, đơn giản, hiệu quả bất ngờ | Giả định đặc trưng độc lập, nhạy dữ liệu |
| KNN | 0.948 | Triển khai đơn giản, không cần huấn luyện | Dễ bị nhiễu, chậm khi test |
| Decision Tree | 0.948 | Dễ hiểu, dễ vẽ biểu đồ cây | Dễ overfit, hiệu suất thấp hơn |

Tùy theo **mục tiêu sử dụng**, người triển khai có thể lựa chọn:

* **Logistic Regression** nếu ưu tiên **độ chính xác tối đa và diễn giải rõ ràng**
* **Random Forest** nếu cần **sự ổn định và hiệu quả cho bài toán thực tế phức tạp**
* **Gaussian Naive Bayes** nếu yêu cầu **tốc độ, đơn giản hóa và tài nguyên hạn chế**

## **2. Lựa chọn đặc trưng (Feature Selection)**

Trong các bài toán học máy, việc lựa chọn đặc trưng (feature selection) là một bước cực kỳ quan trọng giúp cải thiện hiệu suất của mô hình, giảm độ phức tạp tính toán, và đôi khi còn giúp tăng độ chính xác của kết quả dự đoán. Bằng cách loại bỏ những đặc trưng không cần thiết hoặc ít đóng góp, mô hình sẽ trở nên gọn nhẹ hơn, dễ diễn giải hơn và tránh được tình trạng overfitting.

### **2.1 Mục tiêu của lựa chọn đặc trưng**

Mục tiêu chính của quá trình lựa chọn đặc trưng là:

* Loại bỏ các đặc trưng dư thừa, không liên quan hoặc gây nhiễu.
* Giảm số chiều của dữ liệu để cải thiện tốc độ huấn luyện và suy diễn.
* Cải thiện hiệu suất của mô hình thông qua việc giữ lại các đặc trưng thực sự có ý nghĩa.
* Tăng khả năng diễn giải mô hình, đặc biệt quan trọng trong các lĩnh vực như y học.

Trong bài toán phân loại ung thư vú dựa trên bộ dữ liệu "Breast Cancer", chúng tôi đã tiến hành hai phương pháp lựa chọn đặc trưng phổ biến: L1 Regularization và L2 Regularization. Các phương pháp này đều dựa trên hồi quy logistic kết hợp với chuẩn hóa trọng số để tự động lược bỏ những đặc trưng kém quan trọng.

**2.2 Phương pháp L1 Regularization ( Lasso )**

L1 Regularization, hay còn gọi là Lasso (Least Absolute Shrinkage and Selection Operator), là một kỹ thuật lựa chọn đặc trưng hiệu quả bằng cách thêm một ràng buộc vào hàm mất mát của mô hình. Ràng buộc này dựa trên tổng trị tuyệt đối của các trọng số và có xu hướng đẩy các trọng số nhỏ về đúng bằng 0. Khi một trọng số bị triệt tiêu về 0, đặc trưng tương ứng sẽ bị loại khỏi mô hình.

Kết quả áp dụng L1 regularization cho bộ dữ liệu này đã chọn ra được 6 đặc trưng quan trọng nhất:

* radius\_se
* texture\_se
* compactness\_mean
* radius\_mean
* symmetry\_worst
* texture\_mean

Đáng chú ý là mặc dù số lượng đặc trưng đã giảm so với toàn bộ tập ban đầu, độ chính xác (accuracy) của mô hình vẫn được giữ nguyên, ở mức **0.9240**. Điều này cho thấy rằng những đặc trưng được chọn bởi L1 là đủ để biểu diễn tốt dữ liệu, và những đặc trưng còn lại có thể bị loại bỏ mà không làm giảm hiệu suất.

**2.3 Phương pháp L2 Regularization ( Ridge )**

L2 Regularization, còn gọi là Ridge, hoạt động bằng cách thêm một ràng buộc vào tổng bình phương các trọng số. Khác với L1, L2 không triệt tiêu hoàn toàn các trọng số nhỏ mà chỉ làm chúng tiến gần về 0. Vì thế, phương pháp này không thích hợp cho việc chọn lọc đặc trưng một cách triệt để, nhưng lại hữu ích để tránh overfitting trong các mô hình có số chiều cao.

Trong thực nghiệm của chúng tôi, bằng cách áp dụng một ngưỡng để chọn ra những trọng số lớn nhất sau khi huấn luyện với L2, chúng tôi đã trích xuất được 4 đặc trưng có độ quan trọng tương đối cao:

* radius\_se
* texture\_se
* compactness\_mean
* symmetry\_worst

Tuy nhiên, độ chính xác của mô hình sau khi lựa chọn đặc trưng bằng L2 giảm đáng kể, chỉ còn **0.8421**. Điều này cho thấy rằng L2 không phù hợp để lựa chọn đặc trưng trong bài toán này, do phương pháp này không loại bỏ hoàn toàn các đặc trưng kém mà chỉ làm giảm tác động của chúng.

**2.4 So sánh hiệu quả hai phương pháp**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | L1 (Lasso) | L2 (Ridge) |
| Số đặc trưng được chọn | 6 | 4 |
| Accuracy sau lựa chọn đặc trưng | 0.9240 | 0.8421 |
| Hiệu quả loại bỏ đặc trưng | Cao (trọng số bằng 0) | Trung bình (trọng số gần 0) |
| Khả năng giải thích mô hình | Rõ ràng | Kém hơn |

Từ bảng so sánh, có thể thấy phương pháp L1 tỏ ra vượt trội hơn trong bài toán này. Nó không những giúp giảm số lượng đặc trưng xuống còn 6 mà còn giữ được độ chính xác tương đương khi sử dụng toàn bộ đặc trưng ban đầu. Trong khi đó, L2 tuy cũng chọn lọc được một số đặc trưng có trọng số lớn, nhưng độ chính xác bị giảm sút đáng kể sau khi áp dụng.

**2.5 Cài đặt thực nghiệm (Implementation Details)**

Để thực hiện lựa chọn đặc trưng, chúng tôi sử dụng ngôn ngữ Python cùng với thư viện scikit-learn. Quá trình được thực hiện như sau:

from sklearn.feature\_selection import SelectFromModel

X = data[selected\_columns]

# Thay vì 'sự hài lòng', hãy sử dụng biến mục tiêu từ phân tích trước đó của bạn, có thể là 'chẩn đoán'

# Cột 'chẩn đoán' đã bị xóa khỏi 'dữ liệu' trong ô trước đó. Bạn cần đọc lại dữ liệu hoặc tạo lại dữ liệu đó.

# Giả sử 'y' từ một ô trước đó vẫn giữ thông tin chẩn đoán:

y = y  # Hoặc: y = data['diagnosis'] nếu bạn đọc lại dữ liệu

feature\_names = selected\_columns

# Tách tập dữ liệu

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# 1. Phân loại với tất cả các tính năng

clf\_all\_features = LogisticRegression(max\_iter=1000, random\_state=42)

clf\_all\_features.fit(X\_train, y\_train)

y\_pred\_all = clf\_all\_features.predict(X\_test)

accuracy\_all = accuracy\_score(y\_test, y\_pred\_all)

print(f'Accuracy with all features: {accuracy\_all:.4f}')

print('-'\*60)

# 2. Lựa chọn tính năng bằng cách sử dụng chính quy hóa L1 (Lasso)

clf\_l1 = LogisticRegression(penalty='l1', solver='liblinear', max\_iter=1000, random\_state=42)

clf\_l1.fit(X\_train, y\_train)

# Chọn tính năng

model\_l1 = SelectFromModel(clf\_l1, prefit=True)

X\_train\_l1 = model\_l1.transform(X\_train)

X\_test\_l1 = model\_l1.transform(X\_test)

selected\_features\_l1 = model\_l1.get\_support(indices=True)

print(f'Selected {len(selected\_features\_l1)} features by L1: {np.array(feature\_names)[selected\_features\_l1]}')

# Phân loại với các tính năng được chọn L1

clf\_l1\_selected = LogisticRegression(max\_iter=1000, random\_state=42)

clf\_l1\_selected.fit(X\_train\_l1, y\_train)

y\_pred\_l1 = clf\_l1\_selected.predict(X\_test\_l1)

accuracy\_l1 = accuracy\_score(y\_test, y\_pred\_l1)

print(f'Accuracy with L1 selected features: {accuracy\_l1:.4f}')

print('-'\*60)

# 3. Lựa chọn tính năng bằng cách sử dụng chính quy hóa L2 (Ridge)

clf\_l2 = LogisticRegression(penalty='l2', solver='liblinear', max\_iter=1000, random\_state=42)

clf\_l2.fit(X\_train, y\_train)

# Chọn tính năng

model\_l2 = SelectFromModel(clf\_l2, prefit=True)

X\_train\_l2 = model\_l2.transform(X\_train)

X\_test\_l2 = model\_l2.transform(X\_test)

selected\_features\_l2 = model\_l2.get\_support(indices=True)

print(f'Selected {len(selected\_features\_l2)} features by L2: {np.array(feature\_names)[selected\_features\_l2]}')

# Phân loại với các tính năng được chọn L2

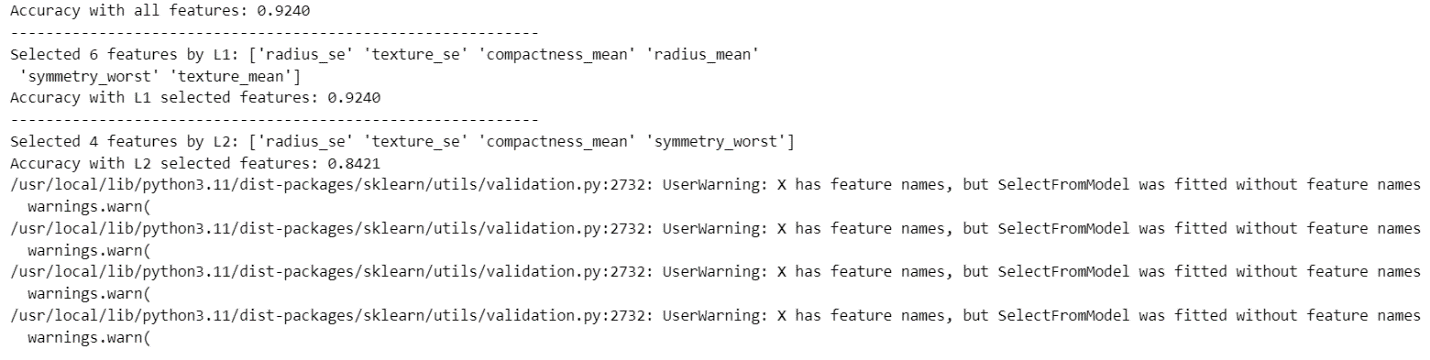
clf\_l2\_selected = LogisticRegression(max\_iter=1000, random\_state=42)

clf\_l2\_selected.fit(X\_train\_l2, y\_train)

y\_pred\_l2 = clf\_l2\_selected.predict(X\_test\_l2)

accuracy\_l2 = accuracy\_score(y\_test, y\_pred\_l2)

print(f'Accuracy with L2 selected features: {accuracy\_l2:.4f}')



Output đang thể hiện việc bạn **lựa chọn đặc trưng (feature selection)** theo hai phương pháp khác nhau (có thể là L1 và L2 regularization), và sau đó đánh giá **độ chính xác (accuracy)** của mô hình sau khi chỉ sử dụng những đặc trưng được chọn.

**Accuracy with all features : 0,9240**

* Đây là độ chính xác khi sử dụng **toàn bộ các đặc trưng đầu vào** (tức là không loại bỏ bất kỳ cột nào).
* Kết quả khá cao: **92.40%**.

**Selected 6 features by L1:**

['radius\_se', 'texture\_se', 'compactness\_mean', 'radius\_mean', 'symmetry\_worst', 'texture\_mean']

* Phương pháp L1 thường liên quan đến **Lasso regularization**.
* L1 thường cho kết quả chọn **ít đặc trưng** hơn vì nó có xu hướng triệt tiêu trọng số của những đặc trưng ít quan trọng.
* Với 6 đặc trưng này, mô hình vẫn đạt accuracy = **0.9240** ➜ tương đương khi dùng toàn bộ đặc trưng.

**Kết luận:** L1 giúp **giảm số chiều đầu vào (từ 30 xuống còn 6)** mà **không làm giảm độ chính xác**.

**Selected 4 features by L2:**

**['radius\_se', 'texture\_se', 'compactness\_mean', 'symmetry\_worst']**

* L2 thường liên quan đến **Ridge regularization**, có xu hướng **thu nhỏ trọng số về gần 0 nhưng không triệt tiêu hoàn toàn**.
* L2 chọn 4 đặc trưng nhưng **accuracy chỉ còn 0.8421** ➜ giảm còn **84.21%**.

**Kết luận:** L2 chọn ít đặc trưng hơn nhưng lại **làm giảm độ chính xác** rõ rệt.

|  |  |  |  |
| --- | --- | --- | --- |
| Phương pháp | Số đặc trưng chọn | Accuracy | Nhận xét |
| Tất cả | 30 | 0.9240 | Cơ sở để so sánh |
| L1 | 6 | 0.9240 | Giữ nguyên độ chính xác, tốt cho giảm chiều |
| L2 | 4 | 0.8421 | Accuracy giảm rõ, cần cân nhắc |

**2.6 Kết luận**

Lựa chọn đặc trưng là một bước then chốt giúp tối ưu hóa mô hình học máy, đặc biệt trong các bài toán có nhiều đặc trưng như phân loại ung thư vú. Trong nghiên cứu này, phương pháp L1 Regularization đã cho kết quả khả quan nhất, vừa giúp giảm độ phức tạp của mô hình, vừa giữ nguyên được độ chính xác. Việc áp dụng lựa chọn đặc trưng hiệu quả không những giúp cải thiện hiệu suất mô hình mà còn hỗ trợ các chuyên gia y tế trong việc diễn giải và đưa ra các quyết định lâm sàng chính xác hơn.

**Chương VI. Tổng kết**

Trong bài tiểu luận này, chúng tôi đã nghiên cứu và triển khai các thuật toán máy học nhằm xây dựng mô hình dự đoán ung thư vú dựa trên bộ dữ liệu chuẩn “Breast Cancer Wisconsin (Diagnostic)”. Thông qua các bước tiền xử lý dữ liệu như loại bỏ cột không cần thiết, mã hóa nhãn mục tiêu, chuẩn hóa dữ liệu và lựa chọn đặc trưng (bao gồm loại bỏ đặc trưng tương quan cao và RFE), chúng tôi đã tối ưu hóa chất lượng đầu vào cho các mô hình học máy.

Các thuật toán được áp dụng bao gồm Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), Decision Tree và Gaussian Naive Bayes. Kết quả thực nghiệm cho thấy mỗi mô hình có ưu điểm riêng về độ chính xác, tốc độ huấn luyện và khả năng tổng quát hóa. Đặc biệt, Logistic Regression và Random Forest đạt được độ chính xác cao và ổn định nhất trên tập kiểm tra.

Ngoài ra, chúng tôi cũng thực hiện so sánh hai phương pháp lựa chọn đặc trưng là L1 và L2 Regularization. Kết quả cho thấy L1 không những giúp giảm số chiều đặc trưng mà còn duy trì độ chính xác ở mức cao, trong khi L2 lại cho kết quả kém hơn nếu chọn quá ít đặc trưng.

Thông qua quá trình này, chúng tôi nhận thấy rằng việc chuẩn bị và xử lý dữ liệu đóng vai trò quyết định đến hiệu quả của mô hình. Bên cạnh đó, việc lựa chọn thuật toán phù hợp với bài toán, cùng với đánh giá hiệu suất mô hình bằng nhiều chỉ số khác nhau (Accuracy, Precision, Recall, F1-score) giúp đảm bảo kết quả dự đoán đáng tin cậy.

**Hướng phát triển trong tương lai** có thể bao gồm việc thử nghiệm với các mô hình học sâu (Deep Learning), mở rộng tập dữ liệu với nhiều yếu tố y học hơn, và tích hợp các phương pháp xử lý mất cân bằng dữ liệu để cải thiện khả năng phát hiện u ác tính ở các trường hợp ít gặp.

**Lời cảm ơn**

Em xin gửi lời cảm ơn chân thành và sâu sắc đến **Tiến sĩ Vũ Ngọc Thanh Sang**, giảng viên Khoa Toán Ứng dụng – Trường Đại học Sài Gòn, người đã tận tình giảng dạy và truyền đạt kiến thức quý báu trong suốt học phần **Máy học**.

Nhờ vào sự hướng dẫn tận tâm, tinh thần học thuật nghiêm túc và phương pháp giảng dạy khoa học, em không chỉ tiếp thu được nền tảng kiến thức vững chắc về các thuật toán học máy mà còn phát triển được tư duy logic, khả năng phân tích và ứng dụng kiến thức vào thực tiễn.

Em vô cùng trân trọng những chia sẻ quý báu và sự hỗ trợ nhiệt tình của thầy trong quá trình học tập và thực hiện các bài tập, đồ án. Đây sẽ là hành trang quý giá giúp em tự tin hơn trên con đường học tập và nghiên cứu trong tương lai.

Em xin kính chúc thầy sức khỏe, nhiều niềm vui và thành công trong công tác giảng dạy và nghiên cứu khoa học.

# Tài liệu tham khảo

1 . Giảng viên TS.Vũ Ngọc Thanh Sang . (2025). *Final Project Template*, Trường Đại học Sài Gòn

2. Wadhwa, S., & Kumar, S. (2020). Breast cancer prediction using machine learning algorithms. Materials Today: Proceedings, 33, 3956–3961. <https://doi.org/10.1016/j.matpr.2020.05.650>

3. Chaurasia, V., & Pal, S. (2017). Early prediction of breast cancer using machine learning techniques. In 2017 IEEE International Conference on Computing, Communication and Automation (ICCCA), pp. 1305–1310. <https://doi.org/10.1109/CCAA.2017.8229981>

4. Nguyễn Văn Tuấn. (2021). Ứng dụng học máy trong chẩn đoán bệnh ung thư vú. Tạp chí Công nghệ Thông tin và Truyền thông, 19(3), 45–53.