TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO HỌC PHẦN MÁY HỌC CƠ BẢN

ĐỀ TÀI

ỨNG DỤNG CÁC THUẬT TOÁN MÁY HỌC TRONG VIỆC DỰ DOÁN MỘT LOẠI NẤM CÓ ĐỘC HAY KHÔNG

Giảng viên hướng dẫn : TS. LÊ NGỌC HIẾU
Sinh viên thực hiện : HOÀNG GIA KIỆT
: NGUYỄN TIẾN ĐẠT
Lớp : CÔNG NGHỆ THÔNG TIN
Khoá : 62

Tp. Hồ Chí Minh, năm 2024

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO HỌC PHẦN MÁY HỌC CƠ BẢN

ĐỀ TÀI

ỨNG DỤNG CÁC THUẬT TOÁN MÁY HỌC TRONG VIỆC DỰ DOÁN MỘT LOẠI NẤM CÓ ĐỘC HAY KHÔNG

Giảng viên hướng dẫn : TS. LÊ NGỌC HIẾU
Sinh viên thực hiện : HOÀNG GIA KIỆT – 6251071049
: NGUYỄN TIẾN ĐẠT – 6251071019
Lớp : CÔNG NGHỆ THÔNG TIN
Khoá : 62

Tp. Hồ Chí Minh, năm 2024

ABSTRACT, KEYWORDS

With the advancement of Machine Learning, predicting whether a mushroom is poisonous has become significantly easier and more effective, offering immense potential to reduce cases of mushroom poisoning and enhance public knowledge about mushrooms. This study explores the implementation and comparative evaluation of various Machine Learning algorithms, including Decision Tree, Random Forests, Logistic Regression, Naive Bayes, AdaBoost, Gradient Boosting, and ANN, to build predictive models using the publicly available Binary Prediction of Poisonous Mushrooms dataset. This dataset encompasses various physical characteristics of mushrooms, providing a comprehensive foundation for predictive modeling.

The research process is divided into several critical phases, starting with data preprocessing to handle missing values, outliers, encode categorical variables, and normalize numerical features. Predictive models are then trained and fine-tuned through advanced hyperparameter optimization strategies, such as Grid Search and Random Search, to ensure optimal performance.

Extensive experiments reveal that methods such as Random Forest, Gradient Boosting, and ANN outperform simpler algorithms in terms of accuracy and generalizability. The best-performing model achieves a prediction accuracy of over 90%, demonstrating its efficacy in accurately classifying whether a mushroom is poisonous or not.

This study underscores the transformative potential of Machine Learning in addressing practical issues such as distinguishing poisonous mushrooms. By integrating robust predictive models with domain-specific insights, this research not only contributes to reducing cases of mushroom poisoning but also provides a replicable framework for future studies tackling similar classification problems.

Keywords: Poisonous Mushroom Prediction, Machine Learning, Decision Trees, Random Forest, AdaBoost, Gradient Boosting, Hyperparameter Optimization, Ensemble Learning, Neural Networks.

TÓM TẮT, TỪ KHÓA

Với sự phát triển của Machine Learning, việc dự đoán nấm có độc hay không đã trở nên dễ dàng và hiệu quả hơn đáng kể, mang lại tiềm năng to lớn để giảm các trường hợp bị ngộ độc do ăn trúng nấm độc và góp phần giúp mọi người có kiến thức hơn về nấm. Nghiên cứu này khám phá việc triển khai và đánh giá so sánh các thuật toán Machine Learning khác nhau, bao gồm Decision Tree, Random Forest, Logistic Regression, Naive Bayes, AdaBoost, Gradient Boosting và ANN, nhằm xây dựng các mô hình dự đoán sử dụng tập dữ liệu công khai **Binary Prediction of Poisonous Mushrooms**. Tập dữ liệu này bao gồm nhiều đặc điểm vật lý của nấm cung cấp khá toàn diện cho việc xây dựng mô hình dự đoán.

Quy trình nghiên cứu được chia thành nhiều giai đoạn quan trọng, bắt đầu với việc tiền xử lý dữ liệu để xử lý các giá trị bị thiếu, giá trị ngoại lai, mã hóa các biến phân loại và chuẩn hóa các đặc điểm số. Sau đó, các mô hình dự đoán được huấn luyện và tinh chỉnh thông qua các chiến lược tối ưu hóa siêu tham số tiên tiến, chẳng hạn như Grid Search và RandomSearch, nhằm đảm bảo hiệu suất tối ưu.

Các thử nghiệm chuyên sâu cho thấy các phương pháp đặc biệt là Random Forests, Gradient Boosting và ANN, vượt trội hơn so với các thuật toán đơn giản hơn về độ chính xác và khả năng tổng quát hóa. Mô hình hoạt động tốt nhất đạt được độ chính xác dự đoán trên 90%, chứng minh hiệu quả trong việc phân loại chính xác liệu nấm có độc hay là không.

Nghiên cứu này nhấn mạnh tiềm năng chuyển đổi của Machine Learning trong việc giải quyết các vấn đề thực tế như phân biệt nấm độc. Bằng cách tích hợp các mô hình dự đoán mạnh mẽ với những hiểu biết chuyên môn theo ngành, nghiên cứu này không chỉ đóng góp vào sự giảm thiểu các ca ngộ độc nấm mà còn cung cấp một khuôn khổ có thể áp dụng lại cho các nghiên cứu trong tương lai nhằm giải quyết các vấn đề phân loại tương tự.

Từ khóa: Dự đoán nấm độc hay không, Machine Learning, Cây quyết định, Random Forest, AdaBoost, Gradient Boosting, Tối ưu hóa siêu tham số, Học tập tổ hợp, Mạng nơ-ron.

LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em xin gửi lời cảm ơn sâu sắc đến thầy Lê Ngọc Hiếu đã truyền đạt kiến thức, hỗ trợ và giúp đỡ nhóm chúng em hoàn thành học phần Học máy cơ bản.

Chúng em cũng xin gửi lời cảm ơn đến Quý thầy cô Bộ môn Công nghệ thông tin Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh đã truyền đạt những kiến thức nền tảng cho chúng em và hỗ trợ chúng em khi có những khó khăn hoàn thành đề tài của học phần.

Ngoài ra, chúng em xin gửi lời cảm ơn đến các anh, chị, bạn đã đồng hành và giúp đỡ chúng em về tài liệu. Trong quá trình thực hiện đề tài, chúng em còn gặp nhiều khó khăn, thiếu kiến thức thực tiễn nên dự án phát triển ra có thể chưa được đúng với các hệ thống thực tế hiện tại. Mong thầy có thể xem xét, đánh giá cho báo cáo của nhóm chúng em được cập nhật và chỉnh sửa tốt nhất.

Chúng em xin chân thành cảm ơn thầy./.

TP. Hồ Chí Minh, ngày ... tháng ... năm 2024

TM. Nhóm sinh viên thực hiện

Hoàng Gia Kiệt

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

TP. Hồ Chí Minh, ngày ... tháng ... năm 2024

Giảng viên hướng dẫn

TS. Lê Ngọc Hiếu

MỤC LỤC

DANH MỤC CHỮ VIẾT TẮT

STT	Viết tắt	Ý nghĩa	Diễn giải
1	ML	Machine Learning	Máy học
2	AI	Artificial Intelligence	Trí tuệ nhân tạo
3	DL	Deep Learning	Học sâu
4	DNN	Deep Neural Networks	Mạng nơ ron sâu
5	AdaBoost	Adaptive Boosting	
6	ANN	Artificial Neural Networks	Mạng nơ-ron nhân tạo
7	GPU	Graphics Processing Unit	
8	IEEE	Institute of Electrical and Electronics Engineers	
9	DT	Decision Tree	
10	RF	Random Forest	
11	ROC AUC	Area Under ROC Curve	
12	IT	Information Technology	
13	PSO	Particle Swarm Optimization	
14	GA	Genetic Algorithm	

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI NGHIÊN CỨU

1.1. Giới thiệu đề tài và lý do chọn đề tài

Nấm từ lâu đã đóng một vai trò quan trọng trong đời sống con người. Không chỉ là nguồn thực phẩm dồi dào và đa dạng, nấm còn được sử dụng rộng rãi trong lĩnh vực y học và công nghiệp. Tuy nhiên, bên cạnh những lợi ích mà nấm mang lại, một số loài nấm lại tiềm ẩn những nguy hiểm đáng kể do chứa các chất độc tự nhiên. Việc ăn phải nấm độc có thể gây ra những hậu quả nghiêm trọng đối với sức khỏe, bao gồm ngộ độc cấp tính, tổn thương các cơ quan quan trọng hoặc thậm chí dẫn đến tử vong. Điều này đã đặt ra nhu cầu cấp bách trong việc phân loại và xác định nấm độc nhằm bảo vệ an toàn cho con người.

Trong tự nhiên, có hàng ngàn loài nấm khác nhau với những đặc điểm đa dạng, nhưng không phải tất cả đều an toàn cho con người. Việc xác định nấm độc từ lâu đã dựa vào kinh nghiệm hoặc các phương pháp thủ công, tuy nhiên, những cách này thường thiếu chính xác và tiềm ẩn nhiều rủi ro. Trước thực tế đó, việc áp dụng công nghệ học máy trong việc dự đoán và phân loại nấm độc là một giải pháp đầy triển vọng, giúp cung cấp các kết quả nhanh chóng và đáng tin cậy. Bên cạnh đó, với sự phát triển của các bộ dữ liệu chất lượng cao về nấm và các thuật toán học máy hiện đại, nghiên cứu này không chỉ là cơ hội để tận dụng sức mạnh của dữ liệu mà còn là cách để mở rộng phạm vi ứng dụng của trí tuệ nhân tạo vào các vấn đề thực tiễn. Hơn nữa, đề tài "**Ứng dụng thuật toán machine learning dự đoán nấm có độc hay không**" này mang ý nghĩa nhân văn sâu sắc, hướng tới việc giảm thiểu các vụ ngộ độc do ăn phải nấm độc, đồng thời bảo vệ sức khỏe cộng đồng một cách hiệu quả. Đây cũng là dịp để người thực hiện nâng cao kỹ năng chuyên môn, kết hợp giữa kiến thức học máy và thực tiễn đời sống, tạo ra những giá trị khoa học và ứng dụng hữu ích.

1.2. Mục tiêu và phạm vi nghiên cứu

1.2.1. Mục tiêu nghiên cứu

Hiểu được các thuật toán phổ biến trong học máy, cách ứng dụng trong các bài toán, bộ dữ liệu phù hợp.

Phát triển và so sánh hiệu suất của bảy thuật toán học máy trong việc dự đoán nấm có độc hay không có độc.

Phân tích hiệu quả của các thuật toán này thông qua các chỉ số như độ chính xác, độ chính xác dự đoán (precision), khả năng phát hiện (recall), và chỉ số F1-score.

Xác định mô hình có hiệu suất tốt nhất để hỗ trợ trong việc dự đoán nấm có độc hay không.

1.2.2. Phạm vi nghiên cứu

Bộ dữ liệu: Binary Prediction of Poisonous Mushrooms [1] từ Kaggle bao gồm thông tin về các đặc điểm hình thái của nấm, như màu sắc mũ, hình dạng thân, bề mặt gills (phiến dưới mũ nấm), và môi trường sinh sống. Mục tiêu của bài toán là dự đoán xem một loại nấm là độc (poisonous) hay an toàn để ăn (edible) dựa trên các đặc trưng này.

Các thuật toán sử dụng:

- Decision Tree
- Random Forest
- Logistic Regression
- Naive Bayes
- AdaBoost
- Gradient Boosting
- ANN

Phân tích: Tập trung vào bài toán phân loại nhị phân và hồi quy (dự đoán nấm có độc hay không độc).

1.3. Giới thiệu về nhóm tác giả

Nhóm chúng em gồm có 2 thành viên: Nguyễn Tiến Đạt và Hoàng Gia Kiệt, cả 2 hiện là sinh viên năm 4 Trường Đại học Giao thông Vận tải Phân hiệu TP. HCM. Chúng em đang theo chuyên ngành Công nghệ thông tin vàcó nền tảng vững chắc về các kiến thức cơ bản của ngành.

1.4. Phương pháp nghiên cứu

Nghiên cứu này tập trung vào việc ứng dụng các thuật toán học máy để dự đoán khả năng một loại nấm có độc hay không, dựa trên việc phân tích bộ dữ liệu thực tế được công khai từ các nguồn như Kaggle. Mục tiêu chính là đánh giá và so sánh hiệu suất của nhiều thuật toán học máy, bao gồm Decision Trees, Random Forest, Naive Bayes, Logistic Regression, AdaBoost, Gradient Boosting và Artificial Neural Networks. Những thuật toán này được lựa chọn dựa trên khả năng đã được chứng minh trong các bài toán phân loại, phù hợp để phân tích các đặc điểm vật lý của nấm nhằm dự đoán tính chất độc hại. Ngoài ra, nghiên cứu cũng tích hợp chiến lược tối ưu hóa như GridSearch để tinh chỉnh các siêu tham số, từ đó cải thiện hiệu suất mô hình về độ chính xác và hiệu quả của thuật toán. Kết quả của nghiên cứu không chỉ mang lại những đóng

góp khoa học mà còn mở ra các ứng dụng thực tiễn trong việc bảo vệ sức khỏe và an toàn thực phẩm.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về trí tuệ nhân tạo (AI)

2.1.1. Trí tuệ nhân tạo là gì?

Trí tuệ nhân tạo (AI) là công nghệ cho phép máy tính và máy móc mô phỏng quá trình học tập, hiểu biết, giải quyết vấn đề, ra quyết định, sáng tạo và tự chủ của con người [2].

Các tổ chức hiện đại thu thập khối lượng dữ liệu cực lớn từ nhiều nguồn khác nhau như cảm biến thông minh, nội dung do con người tạo ra, công cụ giám sát và bản ghi hệ thống. Trí tuệ nhân tạo phân tích và sử dụng dữ liệu để hỗ trợ hoạt động kinh doanh một cách hiệu quả. [3]

2.1.2. Máy học là gì?

Học máy là một nhánh của trí tuệ nhân tạo cho phép các thuật toán khám phá các mẫu ẩn trong bộ dữ liệu để từ đó đưa ra dự đoán về dữ liệu mới. Học máy truyền thống kết hợp dữ liệu với các công cụ thống kê để dự đoán kết quả đầu ra, mang lại những hiểu biết sâu sắc có thể áp dụng được. [4]

Máy học được ứnng dụng rộng rãi: nhận diện ảnh, giọng nói, đề xuất sản phẩm, phát hiện gian lận…

Ví dụ: Facebook dựa vào lịch sử thích, bình luận trên các bài viết hay video, từ đó sẽ hiểu được sở thích của người dùng và đề xuất những nội dung liên quan, cũng như quảng cáo về các nội dung đó.

2.1.3. Học sâu là gì?

Học sâu là một tập hợp con của học máy, tập trung vào việc xây dựng và huấn luyện mạng nơ-ron nhiều lớp, được gọi là mạng nơ-ron sâu (DNN – Deep neural networks) để chúng có thể tự động học, hiểu dữ liệu, mô phỏng khả năng ra quyết định phức tạp của bộ não con người.

Mô hình học sâu có thể nhận diện nhiều hình mẫu phức tạp trong hình ảnh, văn bản, âm thanh và các dữ liệu khác để tạo ra thông tin chuyên sâu và dự đoán chính xác. Bạn có thể sử dụng các phương pháp học sâu để tự động hóa các tác vụ thường đòi hỏi trí tuệ con người, chẳng hạn như phân loại hình ảnh hoặc chép lời một tập tin âm thanh. [5]

2.2. Tổng quan về ngôn ngữ Python

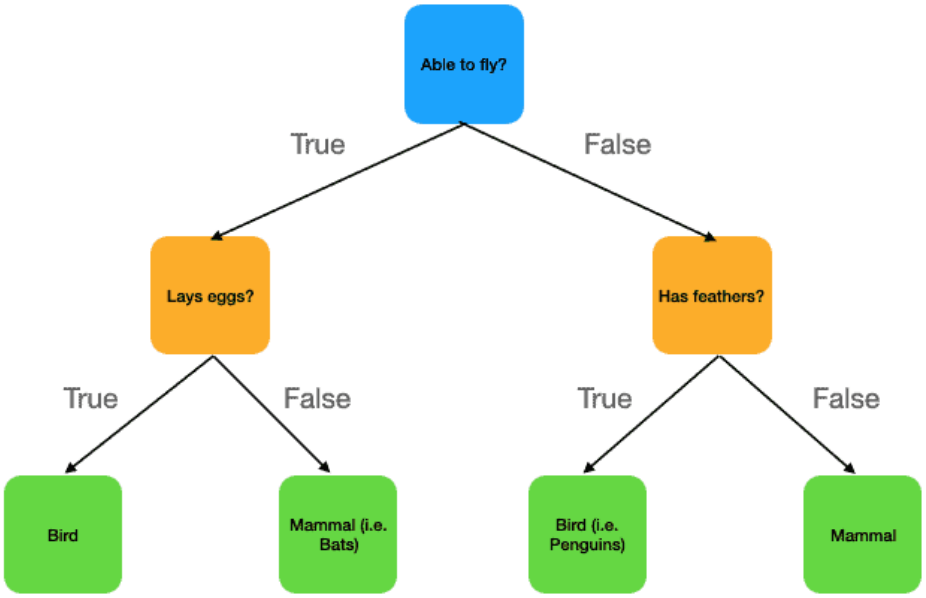
Python là một ngôn ngữ lập trình bậc cao, thông dịch, và đa mục đích, được tạo ra bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Python nổi tiếng với cú pháp đơn giản, dễ học, và dễ đọc, giúp lập trình viên phát triển phần mềm một cách nhanh chóng và hiệu quả. Python được thiết kế với triết lý "Code rõ ràng hơn là code phức tạp", khiến nó trở thành lựa chọn lý tưởng cho người mới bắt đầu cũng như các lập trình viên giàu kinh nghiệm. [6]

Hiện nay chúng ta biết đến Python là một ngôn ngữ lập trình mạnh mẽ được sử dụng phổ biến trong lĩnh vực AI, ML và Khoa học dữ liệu. Ngoài ra Python cũng được ứng dụng trong các lĩnh vực khác như: Web, Game, phần mềm… [7]

2.3. Các thuật toán trong nghiên cứu

2.3.1. Decision Tree (Cây quyết định)

Decision Tree là một thuật toán học giám sát thường được sử dụng trong học máy để dự đoán đầu ra dựa trên các dữ liệu đầu vào. Nó là một cấu trúc dạng cây trong đó mỗi nút bên trong kiểm tra thuộc tính của dữ liệu, mỗi nhánh tương ứng với giá trị của thuộc tính và mỗi nút lá đại diện cho quyết định hoặc dự đoán cuối cùng của thuật toán. Có thể được sử dụng để giải quyết các bài toán hồi quy và phân loại. [8]



Hình 2.1. Ví dụ thuật toán Decision Tree

Ưu điểm:

- Trực quan, dễ hiểu.

- Cần ít dữ liệu để huấn luyện, không yêu cầu chuẩn hoá dữ liệu.

- Có thể xử lý tốt với dữ liệu dạng số (rời rạc và liên tục) và dữ liệu hạng mục.

- Xây dựng nhanh.

Nhược điểm:

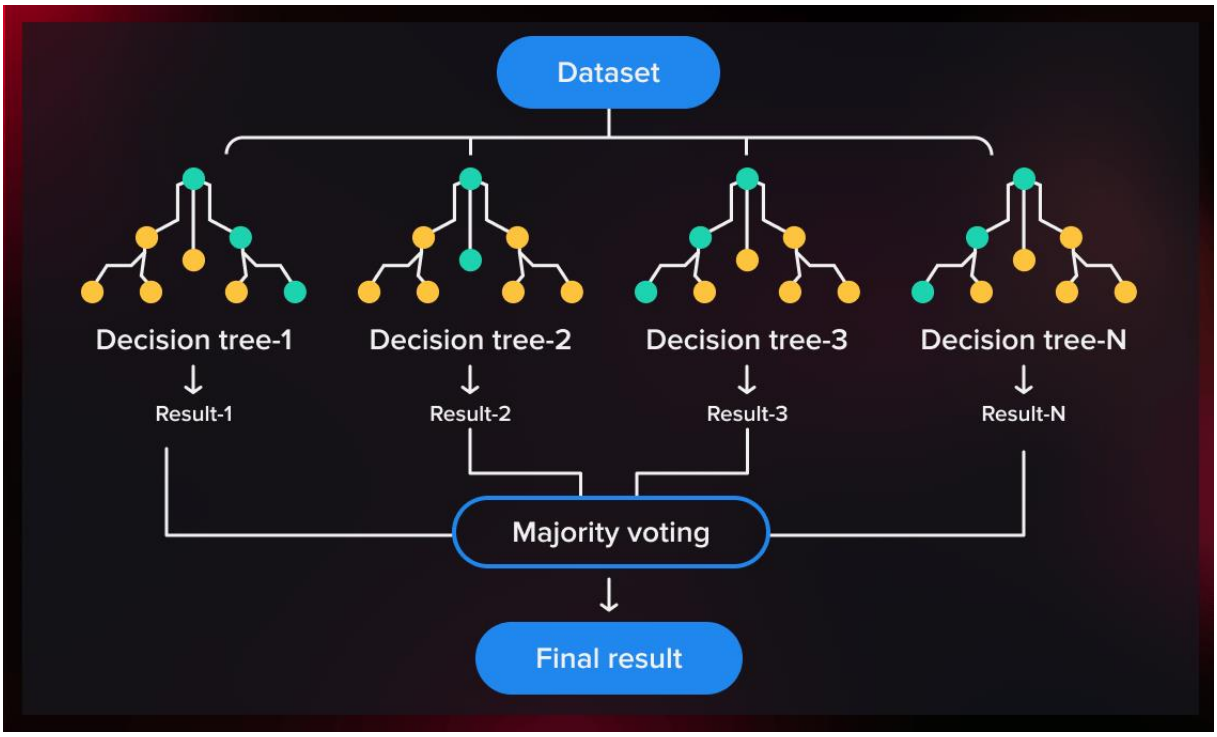
- Không đảm bảo xây dựng được cây tối ưu.

- Dễ bị overfitting nếu không được giới hạn độ sâu của cây.

2.3.2. Random Forest (Rừng ngẫu nhiên)

Random Forest là một thuật toán học máy mạnh mẽ thuộc nhóm thuật toán học giám sát, được sử dụng để giải quyết các bài toán phân loại và hồi quy. Nó là một sự mở rộng của cây quyết định, nhưng có một số cải tiến giúp tăng độ chính xác và giảm hiện tượng overfitting.

Những cái cây trong rừng ngẫu nhiên là những cây quyết định nhưng nó sẽ khác nhau vì các cây được tạo ra bằng yếu tố random. Kết quả dự đoán cuối cùng của rừng ngẫu nhiên sẽ là tổng hợp kết quả dự đoán từ các cây trong rừng theo 2 cách là: bỏ phiếu hoặc là trung bình. [9]



Hình 2.2. Ví dụ thuật toán Random Forest

Ưu điểm:

- Giảm overfitting so với Decision Tree.

- Hiệu năng tốt với nhiều loại dữ liệu.

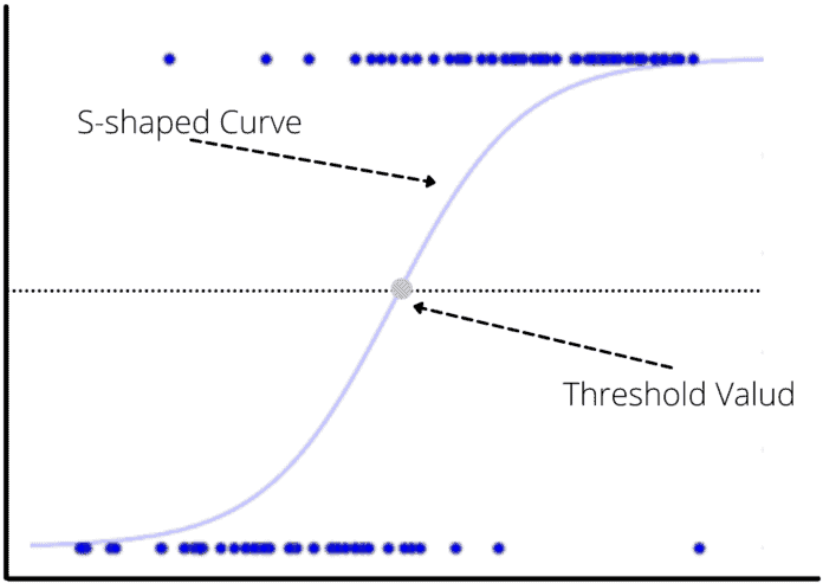
Nhược điểm:

- Cần nhiều tài nguyên để tính toán

2.3.3. Logistic Regression (Hồi quy Logistic)

Logistic Regression là một mô hình thống kê được sử dụng để phân loại nhị phân, tức dự đoán một đối tượng thuộc vào một trong hai nhóm. Hồi quy Logistic làm việc dựa trên nguyên tắc của hàm sigmoid – một hàm để chuyển đầu vào của nó thành đầu ra có giá trị xác xuất trong khoảng 0 đến 1. [10]

Tuỳ vào tập dữ liệu của bài toán mà chúng ta sẽ chọn ngưỡng để thuật toán có thể dự đoán được lớp đầu ra. Ví dụ: Chúng ta chọn ngưỡng là 0.6, nếu giá trị thuật toán cho ra lớn hơn hoặc bằng 0.6 thì đầu vào thuộc lớp 1 và ngược lại là lớp 0.



Hình 2.3. Ví dụ thuật toán Logistic Regression

Ưu điểm:

- Hiệu quả với dữ liệu tuyến tính.

- Chi phí tính toán thấp.

- Không yêu cầu quá nhiều siêu tham số.

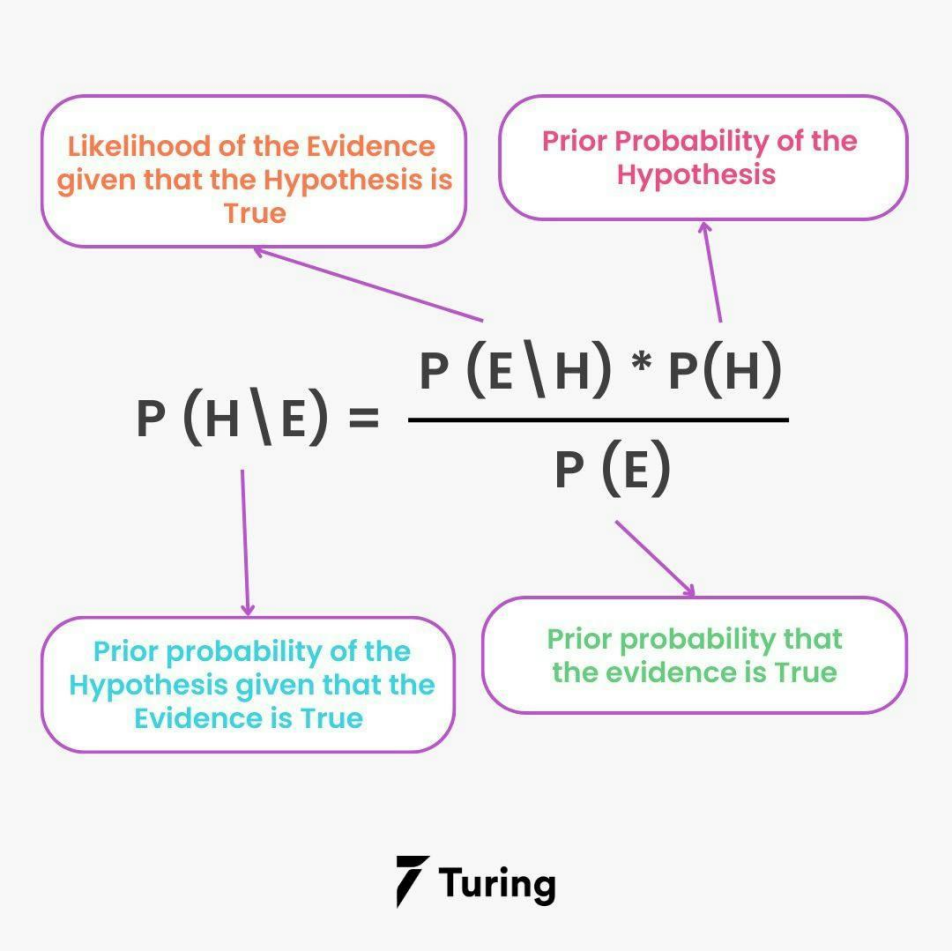
Nhược điểm:

- Hiệu suất kém với dữ liệu phi tuyến tính.

- Không phù hợp với dữ liệu nhỏ hoặc không đầy đủ.

2.3.4. Naive Bayes

Naive Bayes là một thuật toán học máy dựa trên lý thuyết xác suất Bayes và giả định "ngây thơ" rằng các đặc trưng của dữ liệu là độc lập với nhau. Thuật toán này thường được dùng trong các bài toán phân loại, như phân loại văn bản (spam email, cảm xúc), và hoạt động tốt với dữ liệu có tính phân loại cao. [11]



Hình 2.4. Ví dụ thuật toán Naive Bayes

Ưu điểm:

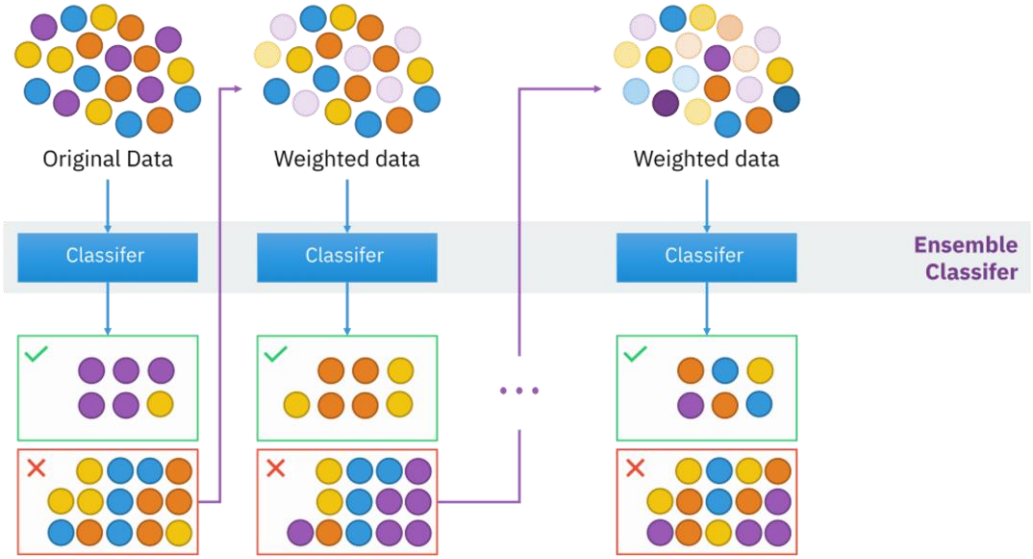
- Dễ triển khai và tính toán nhanh.
- Hoạt động tốt trên tập dữ liệu nhỏ và cả khi có nhiều đặc trưng.
- Ít bị ảnh hưởng bởi hiện tượng overfitting khi có nhiều dữ liệu.

Nhược điểm:

- Giả định độc lập giữa các đặc trưng không phải lúc nào cũng đúng.
- Không phù hợp với dữ liệu phức tạp, có mối quan hệ phi tuyến tính.

2.3.5. AdaBoost

AdaBoost là một thuật toán ensemble, kết hợp nhiều mô hình học yếu, thường là cây quyết định đơn giản, để tạo ra một mô hình mạnh hơn. Ý tưởng chính của AdaBoost là đặt trọng số cao hơn vào các điểm dữ liệu khó phân loại và cải thiện mô hình qua từng vòng lặp. AdaBoost thường được dùng trong bài toán phân loại nhị phân, đa lớp, và cả hồi quy. [12]



Hình 2.5. Ví dụ thuật toán Adaboost

Ưu điểm:

- Hiệu quả với dữ liệu không nhiễu.

- Không cần điều chỉnh nhiều siêu tham số.

- Có thể đạt độ chính xác cao trên nhiều bài toán thực tế.

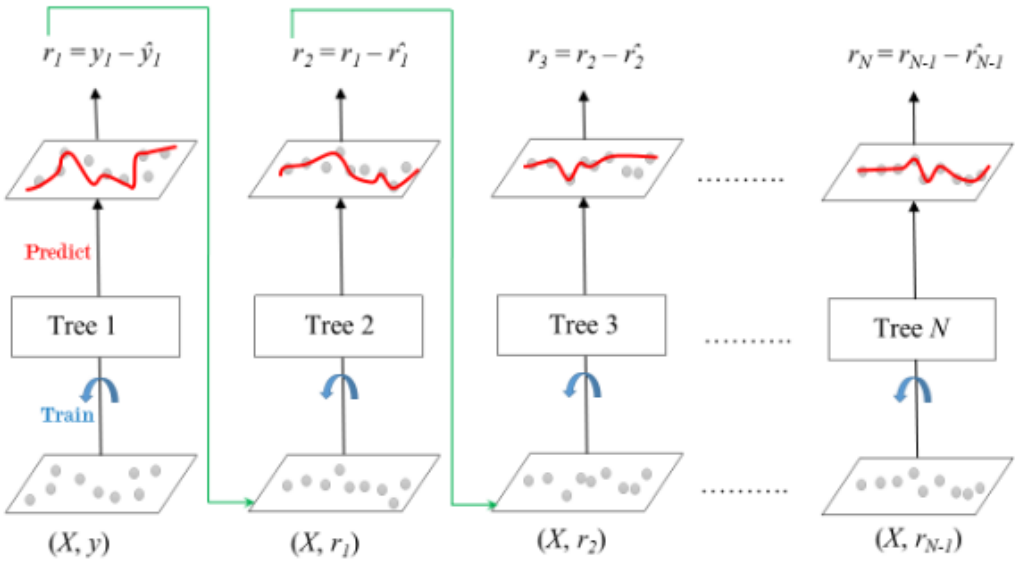
Nhược điểm:

- Nhạy cảm với nhiễu trong dữ liệu.

- Cần nhiều vòng lặp, tốn thời gian với dữ liệu lớn.

2.3.6. Gradient Boosting

Gradient Boosting là một thuật toán mạnh mẽ dựa trên ý tưởng tối ưu gradient. Nó xây dựng một mô hình dự đoán bằng cách kết hợp nhiều mô hình yếu (thường là cây quyết định) qua từng bước. Ở mỗi bước, thuật toán cố gắng giảm thiểu lỗi dự đoán bằng cách tối ưu hóa một hàm mất mát. Gradient Boosting phổ biến trong nhiều lĩnh vực như tài chính, y tế, và phân tích dữ liệu lớn. [13]



Hình 2.6. Ví dụ thuật toán Gradient Boosting

Ưu điểm:

- Hiệu suất cao trên cả dữ liệu tuyến tính và phi tuyến tính.

- Có thể xử lý dữ liệu mất mát tốt bằng cách tùy chỉnh hàm mất mát.

- Hỗ trợ tối ưu hóa linh hoạt với nhiều dạng bài toán.

Nhược điểm:

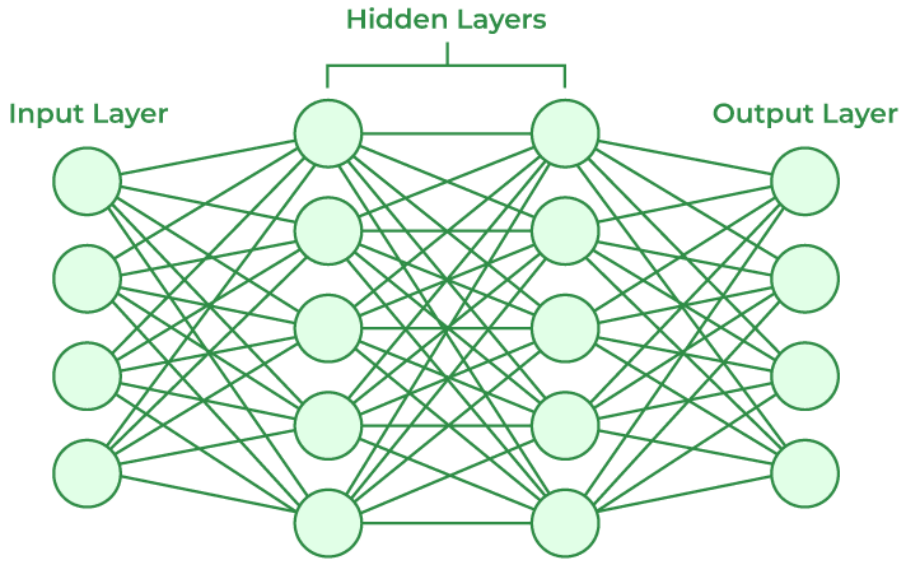
- Tốn tài nguyên tính toán và thời gian huấn luyện.

- Dễ bị overfitting nếu không tinh chỉnh tốt.

2.3.7. ANN

ANN là một thuật toán lấy cảm hứng từ cách hoạt động của não bộ, gồm các lớp nơ-ron nhân tạo kết nối với nhau. ANN học bằng cách điều chỉnh trọng số thông qua quá trình lan truyền ngược (backpropagation) dựa trên hàm mất mát. ANN rất phổ

biến trong các bài toán phức tạp như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên…



[14]

Hình 2.7. Ví dụ thuật toán ANN

Ưu điểm:

- Khả năng mô hình hóa các quan hệ phi tuyến tính phức tạp.
- Linh hoạt, có thể áp dụng cho nhiều dạng dữ liệu.
- Cải thiện hiệu suất nhờ sức mạnh tính toán của GPU.

Nhược điểm:

- Yêu cầu nhiều dữ liệu và tài nguyên tính toán.
- Dễ bị overfitting nếu không áp dụng các kỹ thuật giảm thiểu (như dropout).
- Khó giải thích mô hình.

2.4. Các nghiên cứu trước đây liên quan đến đề tài

Dự đoán tính độc hại của nấm là một lĩnh vực nghiên cứu quan trọng, với nhiều ứng dụng thực tiễn trong việc bảo vệ sức khỏe con người và hỗ trợ phân loại nấm trong nông nghiệp cũng như nghiên cứu sinh học. Các nghiên cứu trước đây đã áp dụng nhiều thuật toán học máy khác nhau để xây dựng mô hình phân loại, đồng thời triển khai các ứng dụng thực tế nhằm giải quyết bài toán này. Một số nghiên cứu trước đây về đề tài:

Mushroom Toxicity Recognition Based on Multigrained Cascade Forests [15]: Nghiên cứu này áp dụng phương pháp rừng ngẫu nhiên đa tầng để nhận diện tính độc của nấm dựa trên các đặc điểm bề ngoài.

Product Prediction of Mushroom Agricultural Plants Using Machine Learning Techniques [16]: Bài báo này trình bày việc sử dụng các phương pháp học máy để xác định nấm độc và không độc, đồng thời đánh giá độ chính xác của các mô hình khác nhau.

Predicting Poisonous Mushrooms by Using Confusion Matrix Method [17]: Nghiên cứu này tập trung vào việc dự đoán tính độc của nấm và phân tích các thuộc tính độc học liên quan, sử dụng ma trận nhầm lẫn để đánh giá hiệu suất mô hình.

Mushroom Poisonous Prediction Based on the Logistic Regression Model [18]: Bài báo đề xuất mô hình hồi quy logistic để dự đoán tính độc của nấm, đạt độ chính xác cao và giảm thiểu rủi ro liên quan đến ngộ độc nấm.

A New Deep Learning Model for the Classification of Poisonous and Edible Mushrooms [19]: Nghiên cứu này giới thiệu mô hình học sâu mới để phân loại nấm độc và nấm ăn được, với kết quả cho thấy mô hình có độ chính xác cao và thời gian huấn luyện, kiểm tra ngắn.

Using Deep Convolutional Neural Networks to Classify Poisonous and Edible Mushrooms Found in China [20]: Nghiên cứu áp dụng mạng nơ-ron tích chập sâu để phân loại nấm độc và nấm ăn được tại Trung Quốc, nhằm giảm thiểu các trường hợp ngộ độc do ăn nhầm nấm độc.

Mặc dù các nghiên cứu về dự đoán tính độc của nấm bằng Machine Learning đã đạt được độ chính xác cao, chúng vẫn tồn tại một số hạn chế:

Đầu tiên, dữ liệu thường hạn chế và không đa dạng, chủ yếu tập trung vào một số loài nấm tại các khu vực cụ thể, dẫn đến khả năng tổng quát hóa kém.

Thứ hai, các mô hình phức tạp như Gradient Boosting hay Deep Neural Networks thiếu tính minh bạch, gây khó khăn trong việc giải thích và ứng dụng thực tiễn. Ngoài ra, các bộ dữ liệu thường chỉ chứa thông tin bề mặt, các đặc điểm vật lý như màu sắc và hình dạng, kích thước mà bỏ qua các yếu tố sinh học và hóa học quan trọng. Cuối cùng, việc chuẩn hóa dữ liệu đầu vào trong môi trường thực tế có thể gây ra lỗi, làm giảm hiệu quả của mô hình. Các nghiên cứu trong tương lai cần tập trung vào nâng cao tính đa dạng của dữ liệu, tích hợp thông tin sinh học và phát triển các mô hình dễ áp dụng thực tiễn.

CHƯƠNG 3: GIỚI THIỆU BỘ DỮ LIỆU

3.1. Tổng quan

Tên bộ dữ liệu: Binary Prediction of Poisonous Mushrooms

Nguồn: Kaggle

Các đặc trưng của tập dữ liệu:

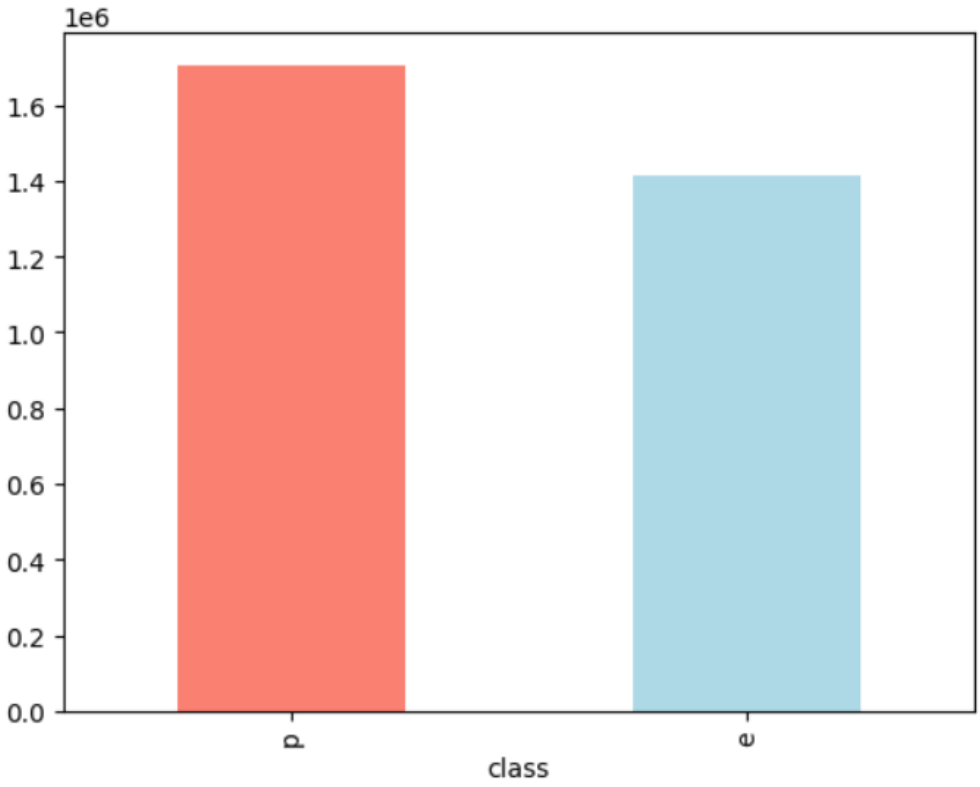
Tên đặc trưng	Kiểu dữ liệu	Mô tả
id	int64	Mã định dạng
class	object	Biến mục tiêu cần dự đoán e (edible) và p (poisonous)
cap-diameter	float64	Đường kính mũ nấm
cap-shape	object	Hình dạng mũ nấm bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
cap-surface	object	Bề mặt mũ nấm fibrous=f, grooves=g, scaly=y, smooth=s
cap-color	object	Màu sắc của mũ nấm brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
does-bruise-or-bleed	object	Có vết dập hay không bruises=t, no=f
gill-attachment	object	Cách phiến nấm gắn vào thân nấm attached=a, descending=d, free=f, notched=n
gill-spacing	object	Khoảng cách giữa các phiến nấm close=c, crowded=w, distant=d
gill-color	object	Màu sắc của phiến nấm black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
stem-height	float64	Chiều cao thân nấm
stem-width	float64	Chiều rộng thân nấm

Tên đặc trưng	Kiểu dữ liệu	Mô tả
stem-root	object	Dạng rễ của thân nấm
stem-surface	object	Bề mặt thân nấm
stem-color	object	Màu sắc thân nấm
veil-type	object	Loại màng che partial=p, universal=u
veil-color	object	Màu sắc của màng che brown=n, orange=o, white=w, yellow=y
has-ring	object	Có vòng tròn không
ring-type	object	Loại vòng tròn cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
spore-print-color	object	Màu sắc của bào tử black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
habitat	object	Môi trường sống của nấm grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d
season	object	Mùa phát triển

Bảng 3.1. Các đặc trưng của tập dữ liệu

3.2. Đặc điểm của bộ dữ liệu

Phân phối biến mục tiêu (class)



Hình 3.1. Biểu đồ phân phối biến mục tiêu "class"

3.2.1. Mô tả thuộc tính "cap-diameter"

	cap-diameter
count	3.116941e+06
mean	6.309848e+00
std	4.657931e+00
min	3.000000e-02
25%	3.320000e+00
50%	5.750000e+00
75%	8.240000e+00
max	8.067000e+01

Hình 3.2. Mô tả thuộc tính "cap-diameter"

- Số lượng dữ liệu (count): Có 3,116,941 mẫu dữ liệu trong thuộc tính.

- Giá trị trung bình (mean): Giá trị trung bình của đường kính mũ nấm là 6.309.

- Độ lệch chuẩn (std): Độ lệch chuẩn 4.657, cho thấy có sự biến động lớn trong giá trị cap-diameter (đường kính mũ nấm).

- Giá trị nhỏ nhất (min): Đường kính mũ nấm nhỏ nhất là 0.03.

- Tứ phân vị thứ nhất (25%): 25% giá trị của đường kính mũ nấm nhỏ hơn hoặc bằng 3.32.

- Trung vị (50%): 50% giá trị của đường kính mũ nấm nhỏ hơn hoặc bằng 5.75. Đây là giá trị giữa khi sắp xếp dữ liệu theo thứ tự tăng dần.

- Tứ phân vị thứ ba (75%): 75% giá trị của đường kính mũ nấm nhỏ hơn hoặc bằng 8.24.

- Giá trị lớn nhất (max): Giá trị lớn nhất của đường kính mũ nấm là 80.67.

3.2.2. Mô tả thuộc tính "stem-height"

	stem-height
count	3.116945e+06
mean	6.348333e+00
std	2.699755e+00
min	0.000000e+00
25%	4.670000e+00
50%	5.880000e+00
75%	7.410000e+00
max	8.872000e+01

Hình 3.3. Mô tả thuộc tính "stem-height"

- Số lượng dữ liệu (count): Có 3,116,945 mẫu dữ liệu trong thuộc tính.

- Giá trị trung bình (mean): Giá trị trung bình của chiều cao thân nấm là 6.348.

- Độ lệch chuẩn (std): Độ lệch chuẩn 2.699, cho thấy có sự biến động tương đối lớn trong giá trị stem-height (chiều cao thân nấm).

- Giá trị nhỏ nhất (min): Chiều cao thân nấm nhỏ nhất là 0

- Tứ phân vị thứ nhất (25%): 25% giá trị của chiều cao thân nấm nhỏ hơn hoặc bằng 4.67.

- Trung vị (50%): 50% giá trị của chiều cao thân nấm nhỏ hơn hoặc bằng 5.88. Đây là giá trị giữa khi sắp xếp dữ liệu theo thứ tự tăng dần.

- Tứ phân vị thứ ba (75%): 75% giá trị của chiều cao thân nấm nhỏ hơn hoặc bằng 7.41.

- Giá trị lớn nhất (max): Giá trị lớn nhất của chiều cao thân nấm là 88.72.

3.2.3. Mô tả thuộc tính "stem-width"

	stem-width
count	3.116945e+06
mean	1.115379e+01
std	8.095477e+00
min	0.000000e+00
25%	4.970000e+00
50%	9.650000e+00
75%	1.563000e+01
max	1.029000e+02

Hình 3.4. Mô tả thuộc tính "stem-width"

- Số lượng dữ liệu (count): Có 3,116,945 mẫu dữ liệu trong thuộc tính.

- Giá trị trung bình (mean): Giá trị trung bình của chiều rộng thân nấm là 11.153

- Độ lệch chuẩn (std): Độ lệch chuẩn 8.095, cho thấy có sự biến động lớn trong giá trị stem-width (chiều rộng thân nấm).

- Giá trị nhỏ nhất (min): Chiều rộng thân nấm nhỏ nhất là 0.

- Tứ phân vị thứ nhất (25%): 25% giá trị của chiều rộng thân nấm nhỏ hơn hoặc bằng 4.97.

- Trung vị (50%): 50% giá trị của chiều rộng thân nấm nhỏ hơn hoặc bằng 9.65. Đây là giá trị giữa khi sắp xếp dữ liệu theo thứ tự tăng dần.

- Tứ phân vị thứ ba (75%): 75% giá trị của chiều rộng thân nấm nhỏ hơn hoặc bằng 15.63.

- Giá trị lớn nhất (max): Giá trị lớn nhất của chiều rộng thân nấm là 102.9.

3.2.4. Mô tả thuộc tính "class"

	class
count	3116945
unique	2
top	p
freq	1705396

Hình 3.5. Mô tả thuộc tính "class"

- Tổng số mẫu dữ liệu có trong thuộc tính: 3,116,945 mẫu.

- Số lượng giá trị duy nhất của thuộc tính: 2.

- Giá trị xuất hiện phổ biến nhất trong thuộc tính: p.

- Số lần giá trị phổ biến nhất xuất hiện: 1,705,396 lần.

3.2.5. Mô tả thuộc tính "cap-shape"

	cap-shape
count	3116905
unique	74
top	x
freq	1436026

Hình 3.6. Mô tả thuộc tính "cap-shape"

- Tổng số mẫu dữ liệu có trong thuộc tính: 3,116,905 mẫu.

- Số lượng giá trị duy nhất của thuộc tính: 74.
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: x (convex).
- Số lần giá trị phổ biến nhất xuất hiện: 1,436,026 lần.

3.2.6. Mô tả thuộc tính "does-bruise-or-bleed"

	cap-surface
count	2445922
unique	83
top	t
freq	460777

Hình 3.7. Mô tả thuộc tính "does-bruise-or-bleed"

- Tổng số mẫu dữ liệu có trong thuộc tính: 3,116,937 mẫu.
- Số lượng giá trị duy nhất của thuộc tính: 26.
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: f.
- Số lần giá trị phổ biến nhất xuất hiện: 2,569,743 lần.

3.2.7. Mô tả thuộc tính "gill-color"

	gill-color
count	3116888
unique	63
top	w
freq	931538

Hình 3.8. Mô tả thuộc tính "gill-color"

- Tổng số mẫu dữ liệu có trong thuộc tính: 3,116,888 mẫu.
- Số lượng giá trị duy nhất của thuộc tính: 63.

- Giá trị xuất hiện phổ biến nhất trong thuộc tính: w (white).
- Số lần giá trị phổ biến nhất xuất hiện: 931,538 lần.

3.2.8. Mô tả thuộc tính "stem-color"

	stem-color
count	3116907
unique	59
top	w
freq	1196637

Hình 3.9. Mô tả thuộc tính "stem-color"

- Tổng số mẫu dữ liệu có trong thuộc tính: 3,116,907 mẫu.
- Số lượng giá trị duy nhất của thuộc tính: 59.
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: w.
- Số lần giá trị phổ biến nhất xuất hiện: 1,196,637 lần.

3.2.9. Mô tả thuộc tính "has-ring"

	has-ring
count	3116921
unique	23
top	f
freq	2368820

Hình 3.10. Mô tả thuộc tính "has-ring"

- Tổng số mẫu dữ liệu có trong thuộc tính: 3,116,921 mẫu.
- Số lượng giá trị duy nhất của thuộc tính: 23.
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: f.

- Số lần giá trị phổ biến nhất xuất hiện: 2,368,820 lần.

3.2.10. Mô tả thuộc tính "habitat"

	habitat
count	3116900
unique	52
top	d
freq	2177573

Hình 3.11. Mô tả thuộc tính "habitat"

- Tổng số mẫu dữ liệu có trong thuộc tính: 3,116,900 mẫu.
- Số lượng giá trị duy nhất của thuộc tính: 52.
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: d (woods).
- Số lần giá trị phổ biến nhất xuất hiện: 2,177,573 lần.

3.2.11. Mô tả thuộc tính "season"

	season
count	3116945
unique	4
top	a
freq	1543321

Hình 3.12. Mô tả thuộc tính "season"

- Tổng số mẫu dữ liệu có trong thuộc tính: 3,116,945 mẫu.
- Số lượng giá trị duy nhất của thuộc tính: 4.
- Giá trị xuất hiện phổ biến nhất trong thuộc tính: a .
- Số lần giá trị phổ biến nhất xuất hiện: 1,543,321 lần.

CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ THU ĐƯỢC

4.1. Cài đặt môi trường thực nghiệm

Quá trình huấn luyện sử dụng kết hợp các công cụ phần cứng và phần mềm để tăng tốc tính toán và tối ưu hóa hiệu suất của mô hình. Dưới đây là chi tiết về các tài nguyên được sử dụng:

Phần cứng:

Google Colaboratory (Colab): Môi trường dựa trên đám mây này cung cấp quyền truy cập vào GPU Tesla T4, hỗ trợ tăng tốc huấn luyện mô hình.

Phần mềm:

- Scikit-learn: Thư viện máy học cung cấp các thuật toán

- TensorFlow: Khung deep learning đa năng hỗ trợ huấn luyện mô hình Mạng Neural Nhân Tạo (Artificial Neural Network).

4.2. Kết quả đạt được

Phần này trình bày kết quả đạt được khi áp dụng các mô hình học máy khác nhau lên tập dữ liệu **Binary Prediction of Poisonous Mushrooms** đã qua tiền xử lý. Như đã đề cập trong các phần trước, tập dữ liệu đã trải qua nhiều bước tiền xử lý, bao gồm: loại bỏ các bản ghi trùng lặp, mã hóa các đặc trưng phân loại bằng phương pháp Label Encoding, Ordinal Encoding, loại bỏ các giá trị ngoại lai cho các đặc trưng. Sau khi loại bỏ ngoại lai, tập dữ liệu có kích thước **(3042143, 21)** và được chia thành hai tập: **huấn luyện (80%)** và **kiểm tra (20%)**.

Bảy mô hình học máy khác nhau đã được huấn luyện và đánh giá trên tập dữ liệu đã chuẩn bị. Các mô hình này được lựa chọn dựa trên khả năng xử lý bài toán phân loại nhị phân và mức độ phức tạp khác nhau của chúng. Tiêu chí đánh giá chính được sử dụng là **độ chính xác (accuracy)**, đo lường mức độ chính xác tổng thể của dự đoán của mô hình. Độ chính xác đạt được trên tập kiểm tra đối với từng mô hình được trình bày dưới đây:

Mô hình	Độ chính xác
Decision Tree	98.33%
Random Forest	99.21%
Naive Bayes	78.63%

Mô hình	Độ chính xác
Logistic Regression	67.24%
Gradient Boosting	93.53%
AdaBoost	96%
ANN	98.62%

Bảng 4.1. Độ chính xác của các mô hình sau khi huấn luyện

4.3. Đánh giá các mô hình

4.3.1. Decision Tree

```
Confusion Matrix:
[[135172   2607]
 [  2456 163980]]
```

Hình 4.1. Ma trận nhầm lẫn Decision Tree

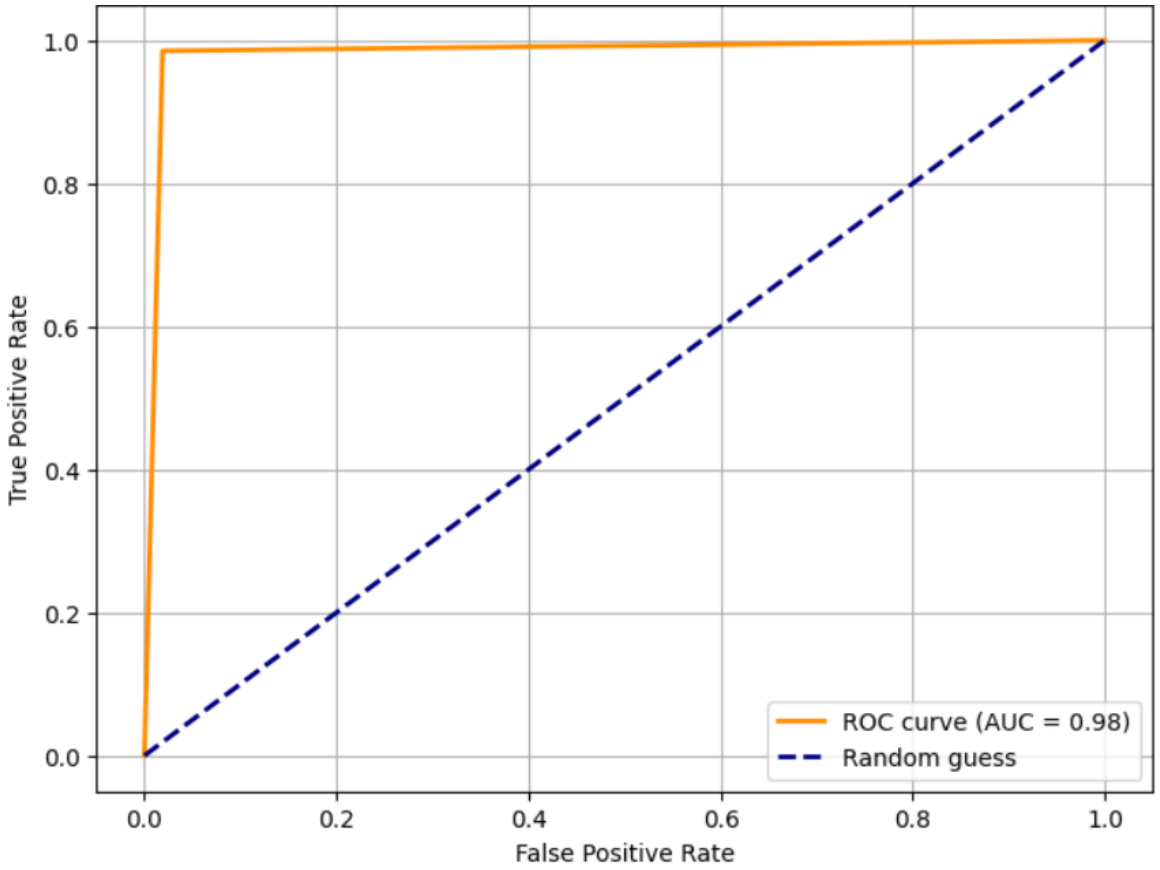
- Dòng 1 (0):
  - Có 135,172 mẫu được dự đoán đúng là 0.
  - Có 2,607 mẫu bị dự đoán nhầm là 1.
- Dòng 2 (1):
  - Có 2,456 mẫu bị dự đoán nhầm là 0.
  - Có 163,980 mẫu dự đoán đúng là 1.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98    137779
           1       0.98      0.99      0.98    166436

    accuracy                           0.98    304215
   macro avg       0.98      0.98      0.98    304215
weighted avg       0.98      0.98      0.98    304215
```

Hình 4.2. Classification Report Decision Tree

- Độ chính xác tổng thể (Accuracy): 98.33%
  - Mô hình phân loại rất tốt với độ chính xác cao.
- Hiệu suất trên từng lớp:

- Lớp 0 (e): Precision = 0.98, Recall = 0.98. Mô hình dự đoán tốt trên lớp này.
- Lớp 1 (p): Precision = 0.98, Recall = 0.99. Mô hình dự đoán tốt trên lớp này.



Hình 4.3. Đường cong ROC của Decision Tree

- Đường cong ROC:
  - Đường màu cam cho thấy Đường cong ROC của mô hình. Đường cong này tiến gần đến góc trên bên trái, cho biết mô hình là một bộ phân loại tốt (Tỷ lệ dương tính thật cao với Tỷ lệ dương tính giả thấp).
- AUC (Diện tích dưới đường cong):
  - Giá trị AUC = 0,98 rất cao, cho biết mô hình có thể phân biệt tốt giữa hai lớp (nấm độc và không độc).
- So với Random Guess:
  - màu xanh chấm bi biểu thị Random Guess với AUC = 0,5. Rõ ràng, mô hình Decision Tree hoạt động tốt hơn Random Guess.
- Ý nghĩa thực tế:

- Với AUC gần bằng 1, mô hình này có bộ phân loại gần như hoàn hảo. Điều này đặc biệt hữu ích khi bạn cần phát hiện chính xác nấm độc để tránh nguy hiểm.

4.3.2. Random Forest

```
Confusion Matrix:
 [[273513   2060]
 [  2593 330263]]
```

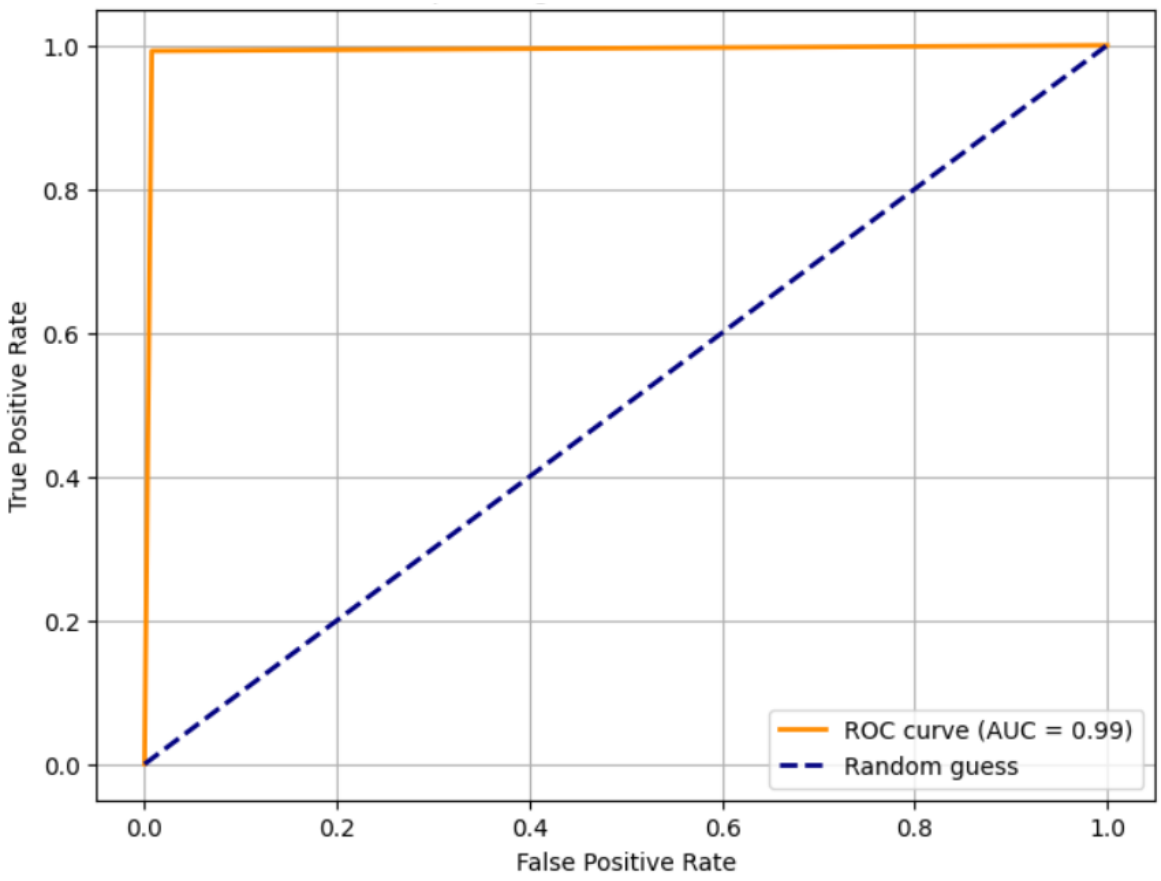
Hình 4.4. Ma trận nhầm lẫn Random Forest

- Dòng 1 (0):
  - Có 273,513 mẫu được dự đoán đúng là 0.
  - Có 2,060 mẫu bị dự đoán nhầm là 1.
- Dòng 2 (1):
  - Có 2,593 mẫu bị dự đoán nhầm là 0.
  - Có 330,263 mẫu dự đoán đúng là 1.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99    275573
           1       0.99      0.99      0.99    332856

    accuracy                           0.99    608429
   macro avg       0.99      0.99      0.99    608429
weighted avg       0.99      0.99      0.99    608429
```

Hình 4.5. Classification Report Random Forest

- Độ chính xác tổng thể: 99.21%
  - Mô hình phân loại rất tốt với độ chính xác cao.
- Hiệu suất trên từng lớp:
  - Lớp 0 (e): Precision = 0.99, Recall = 0.99. Mô hình dự đoán tốt trên lớp này.
  - Lớp 1 (p): Precision = 0.99, Recall = 0.99. Mô hình dự đoán tốt trên lớp này.



Hình 4.6. Đường cong ROC của Random Forest

- Đường cong ROC:
  - Đường màu cam gần như ôm trọn góc trên cùng và góc trái của biểu đồ, cho biết mô hình có Tỷ lệ dương tính thật (TPR) cao ngay cả khi Tỷ lệ dương tính giả (FPR) thấp. Đây là dấu hiệu của một mô hình phân loại mạnh.
- AUC = 0,99:
  - Với AUC cao hơn 0,98 trong hình trước, mô hình có khả năng phân loại thậm chí còn tốt hơn. Giá trị này cho biết mô hình có 99% khả năng phân loại chính xác mẫu dương tính (nấm độc) và mẫu âm tính (nấm không độc).
- So với Random Guess:
  - Đường chấm màu xanh lam (Random Guess) nằm trên đường chéo, tương ứng với AUC = 0,5. Kết quả này nhấn mạnh rằng mô hình Random Forest vượt trội hơn dự đoán ngẫu nhiên.

4.3.3. Naive Bayes

```
Confusion Matrix:
 [[203999  71574]
 [ 58397 274459]]
```

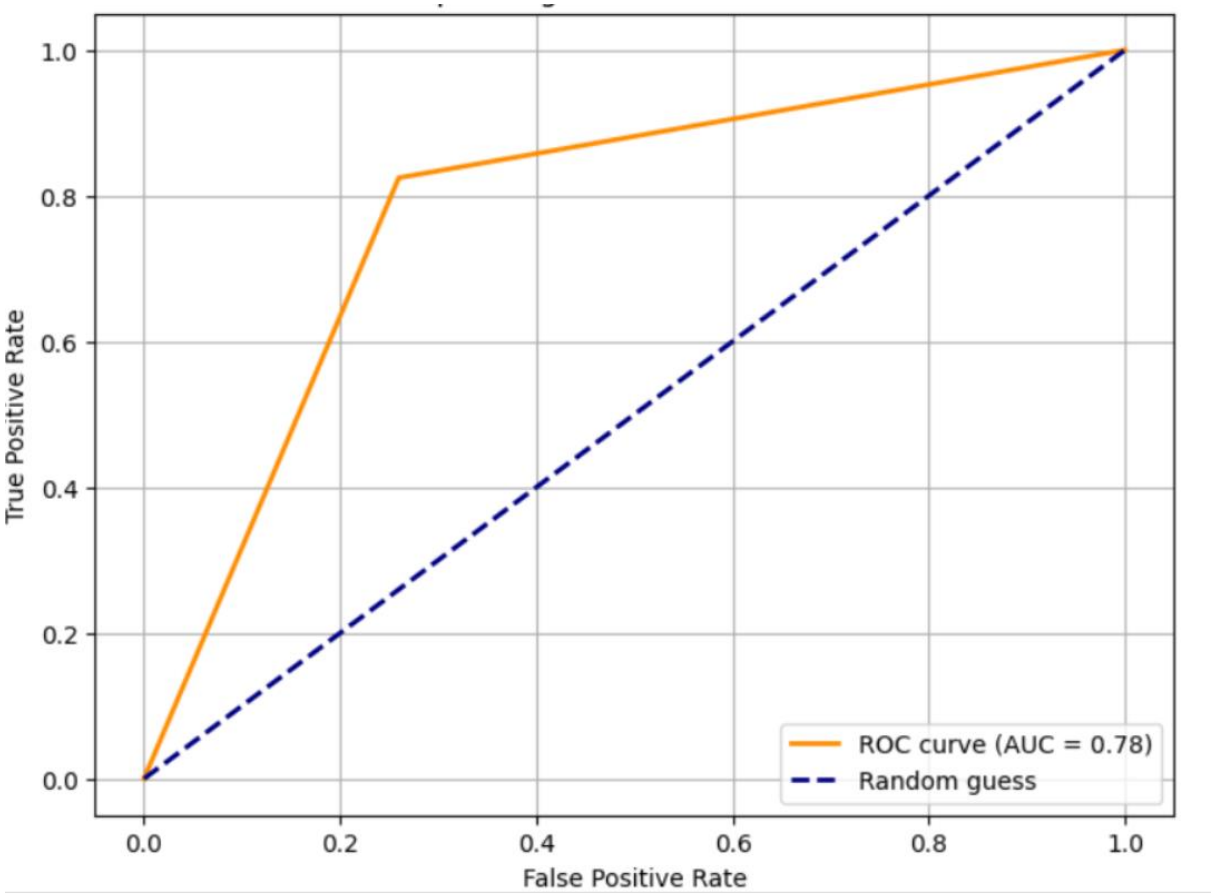
Hình 4.7. Ma trận nhầm lẫn Naive Bayes

- Dòng 1 (0):
  - Có 203,999 mẫu được dự đoán đúng là 0.
  - Có 71,574 mẫu bị dự đoán nhầm là 1.
- Dòng 2 (1):
  - Có 58,397 mẫu bị dự đoán nhầm là 0.
  - Có 274,459 mẫu dự đoán đúng là 1.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.74      0.76    275573
           1       0.79      0.82      0.81    332856

    accuracy                           0.79    608429
   macro avg       0.79      0.78      0.78    608429
weighted avg       0.79      0.79      0.79    608429
```

Hình 4.8. Classification Report Naive Bayes

- Độ chính xác tổng thể: 78.63%
  - Mô hình phân loại chưa được tốt
- Hiệu suất trên từng lớp:
  - Lớp 0 (e): Precision = 0.78, Recall = 0.74. Mô hình dự đoán chưa được tốt trên lớp này. Chưa dự toán đúng nhiều các mẫu lớp 0 và các mẫu thật sự lớp 0 mô hình xác định còn khá kém.
  - Lớp 1 (p): Precision = 0.79, Recall = 0.82. Mô hình dự đoán chưa tốt trên lớp này. Mô hình dự đoán các mẫu thực sự thuộc lớp 1 còn kém, tuy nhiên khả năng xác định các mẫu thực sự là lớp 1 khá ổn.



Hình 4.9. Đường cong ROC của Naive Bayes

- Hiệu suất mô hình: Với AUC = 0,78, mô hình Naive Bayes cho thấy khả năng phân loại tốt nhưng không nổi bật. Điều này ngụ ý rằng mô hình chỉ có 78% khả năng phân loại chính xác một cặp ngẫu nhiên (nấm độc và không độc).
- Đường cong ROC:
    - Đường cong không tiếp cận góc trên bên trái, cho thấy mô hình gặp khó khăn trong việc tối ưu hóa tỷ lệ phát hiện dương tính thực (TPR) khi tỷ lệ dương tính giả (FPR) thấp.
    - Kết quả này có thể chỉ ra rằng mô hình bị ảnh hưởng bởi các tính năng dữ liệu phân tách kém hoặc thiếu tối ưu hóa trong quá trình đào tạo.
- So sánh với Đoán ngẫu nhiên:
    - Đường chấm màu xanh biểu thị đoán ngẫu nhiên với AUC = 0,5. Mô hình này có cải tiến rõ ràng so với đoán ngẫu nhiên, nhưng không lý tưởng (AUC > 0,9).

4.3.4. Logistic Regression

```
Confusion Matrix:
 [[158596 116977]
 [ 82332 250524]]
```

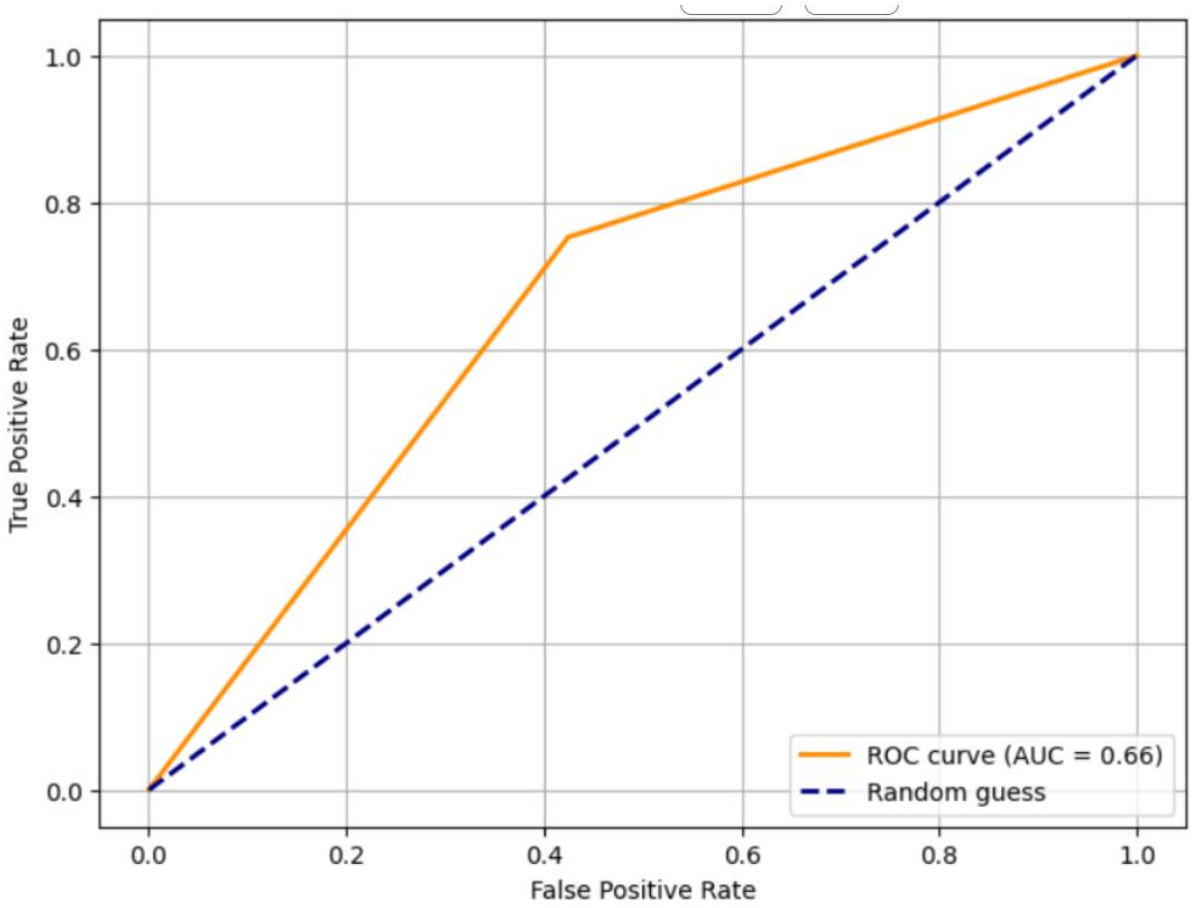
Hình 4.10. Ma trận nhầm lẫn Logistic Regression

- Dòng 1 (0):
  - Có 158,596 mẫu được dự đoán đúng là 0.
  - Có 116,977 mẫu bị dự đoán nhầm là 1.
- Dòng 2 (1):
  - Có 82,332 mẫu bị dự đoán nhầm là 0.
  - Có 250,524 mẫu dự đoán đúng là 1.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.58      0.61    275573
           1       0.68      0.75      0.72    332856

    accuracy                           0.67    608429
   macro avg       0.67      0.66      0.66    608429
weighted avg       0.67      0.67      0.67    608429
```

Hình 4.11. Classification Report Logistic Regression

- Độ chính xác tổng thể: 67.24%
  - Mô hình phân loại chưa được tốt
- Hiệu suất trên từng lớp:
  - Lớp 0 (e): Precision = 0.66, Recall = 0.58. Mô hình dự đoán chưa được tốt trên lớp này. Chưa dự toán đúng nhiều các mẫu lớp 0 và các mẫu thật sự lớp 0 mô hình xác định còn khá kém.
  - Lớp 1 (p): Precision = 0.68, Recall = 0.75. Mô hình dự đoán chưa tốt trên lớp này. Mô hình dự đoán các mẫu thực sự thuộc lớp 1 còn kém, tuy nhiên khả năng xác định các mẫu thực sự là lớp 1 tạm ổn.



Hình 4.12. Đường cong ROC của mô hình Logistic Regression

- Điểm AUC:
  - Diện tích dưới đường cong (AUC) là 0,66, cao hơn dự đoán ngẫu nhiên (AUC = 0,5) nhưng cho thấy mô hình chỉ có khả năng phân biệt vừa phải. Vẫn còn chỗ để cải thiện.
- Hình dạng đường cong:
  - Đường cong ROC cho thấy mô hình có một số khả năng dự đoán nhưng có thể được tối ưu hóa hơn nữa. Đường cong lồi hơn sẽ cho thấy hiệu suất tốt hơn.

4.3.5. Gradient Boosting

```
Confusion Matrix:
[[255008  20565]
 [ 18768 314088]]
```

Hình 4.13. Ma trận nhầm lẫn Gradient Boosting

- Dòng 1 (0):
  - Có 255,008 mẫu được dự đoán đúng là 0.

- Có 20, 565 mẫu bị dự đoán nhầm là 1.

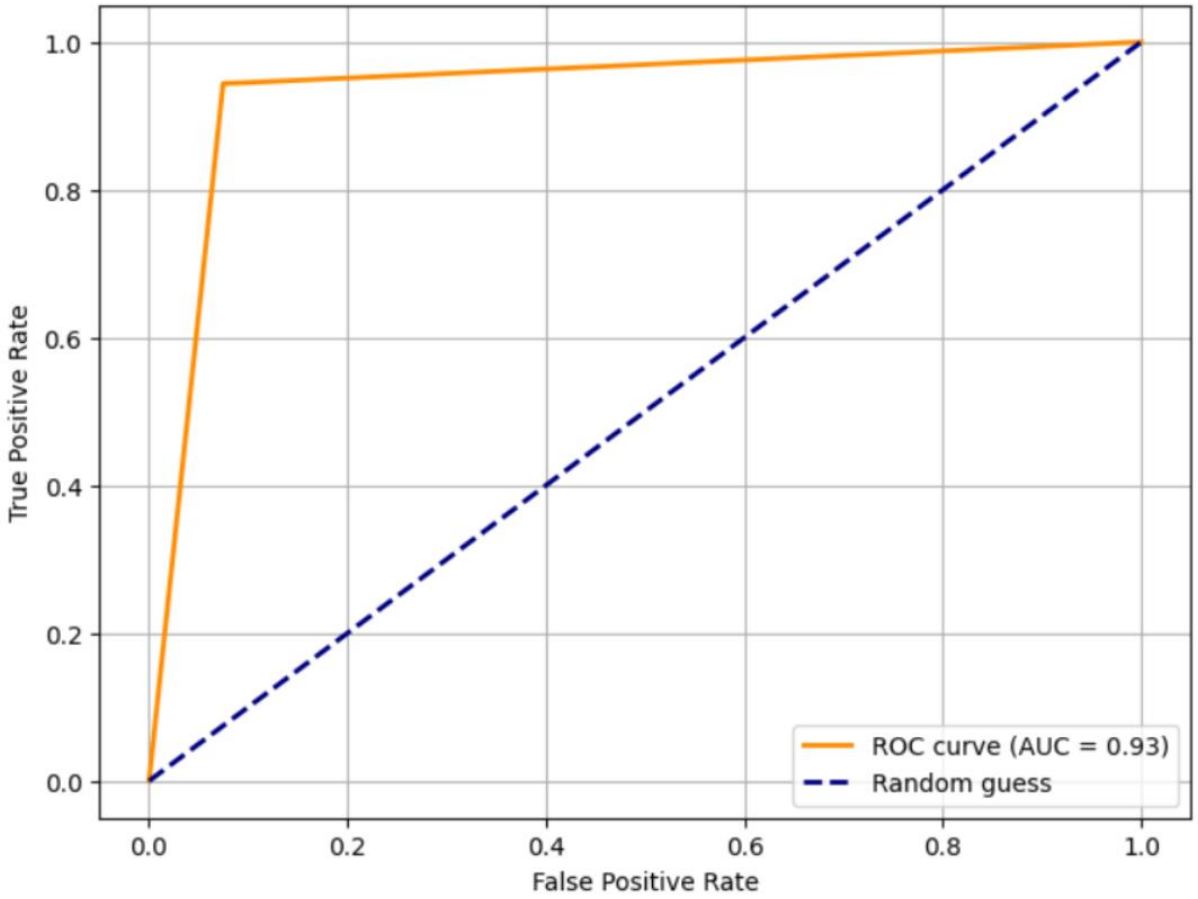
- Dòng 2 (1):
  - Có 18,768 mẫu bị dự đoán nhầm là 0.
  - Có 314,088 mẫu dự đoán đúng là 1.

Classification Report:

	precision	recall	f1-score	support
0	0.93	0.93	0.93	275573
1	0.94	0.94	0.94	332856
accuracy			0.94	608429
macro avg	0.93	0.93	0.93	608429
weighted avg	0.94	0.94	0.94	608429

Hình 4.14. Classification Report Gradient Boosting

- Độ chính xác tổng thể (Accuracy): 93.53%
  - Mô hình phân loại khá tốt
- Hiệu suất trên từng lớp:
  - Lớp 0 (e): Precision = 0.93, Recall = 0.93. Mô hình dự đoán khá tốt trên lớp này.
  - Lớp 1 (p): Precision = 0.94, Recall = 0.94. Mô hình dự đoán khá tốt trên lớp này.



Hình 4.15. Đường cong ROC của Gradient Boosting

- Điểm AUC:
  - AUC đã tăng lên 0,93 cho thấy khả năng phân biệt mạnh. Mô hình hiện rất hiệu quả trong việc phân biệt giữa các lớp dương và âm.
- Hình dạng đường cong:
  - Đường cong lồi hơn và gần góc trên bên trái, biểu thị rằng mô hình đạt được Tỷ lệ dương tính thật cao trong khi vẫn duy trì Tỷ lệ dương tính giả thấp.

4.3.6. AdaBoost

```
Confusion Matrix:
[[264600  10973]
 [ 13318 319538]]
```

Hình 4.16. Ma trận nhầm lẫn AdaBoost

- Dòng 1 (0):
  - Có 264,600 mẫu được dự đoán đúng là 0.
  - Có 10,973 mẫu bị dự đoán nhầm là 1.

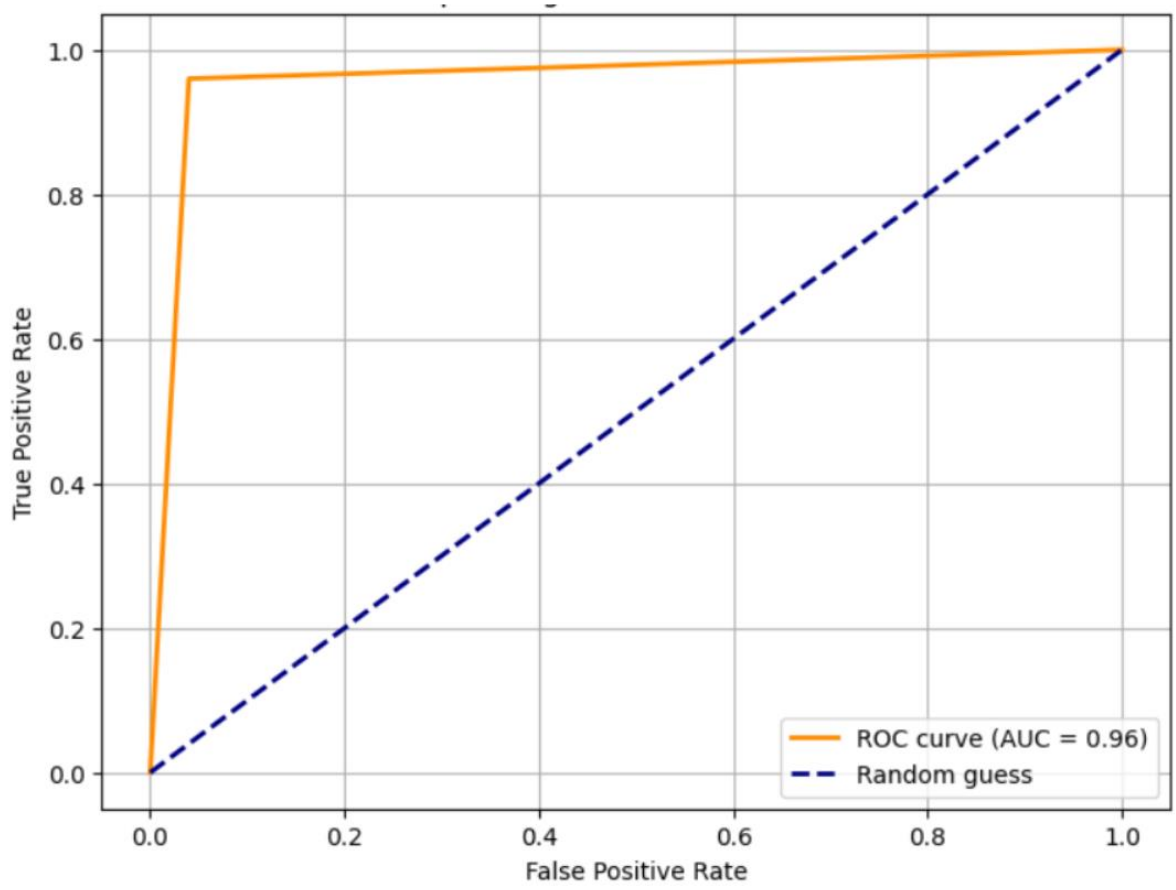
- Dòng 2 (1):
  - Có 13,318 mẫu bị dự đoán nhầm là 0.
  - Có 319,538 mẫu dự đoán đúng là 1.

Classification Report:

	precision	recall	f1-score	support
0	0.95	0.96	0.96	275573
1	0.97	0.96	0.96	332856
accuracy			0.96	608429
macro avg	0.96	0.96	0.96	608429
weighted avg	0.96	0.96	0.96	608429

Hình 4.17. Classification Report Adaboost

- Độ chính xác tổng thể: 96%
  - Mô hình phân loại khá tốt
- Hiệu suất trên từng lớp:
  - Lớp 0 (e): Precision = 0.95, Recall = 0.96. Mô hình dự đoán khá tốt trên lớp này.
  - Lớp 1 (p): Precision = 0.97, Recall = 0.96. Mô hình dự đoán tốt trên lớp này.



Hình 4.18. Đường cong ROC của AdaBoost

- Điểm AUC:
  - AUC đã tăng lên 0,96 cho thấy khả năng phân biệt mạnh. Mô hình hiện rất hiệu quả trong việc phân biệt giữa các lớp dương và âm.
- Hình dạng đường cong:
  - cong lồi hơn và gần góc trên bên trái, biểu thị rằng mô hình đạt được Tỷ lệ dương tính thật cao trong khi vẫn duy trì Tỷ lệ dương tính giả thấp.

4.3.7. ANN

```
Confusion Matrix:
[[271959   3614]
 [  4247 328609]]
```

Hình 4.19. Ma trận nhầm lẫn ANN

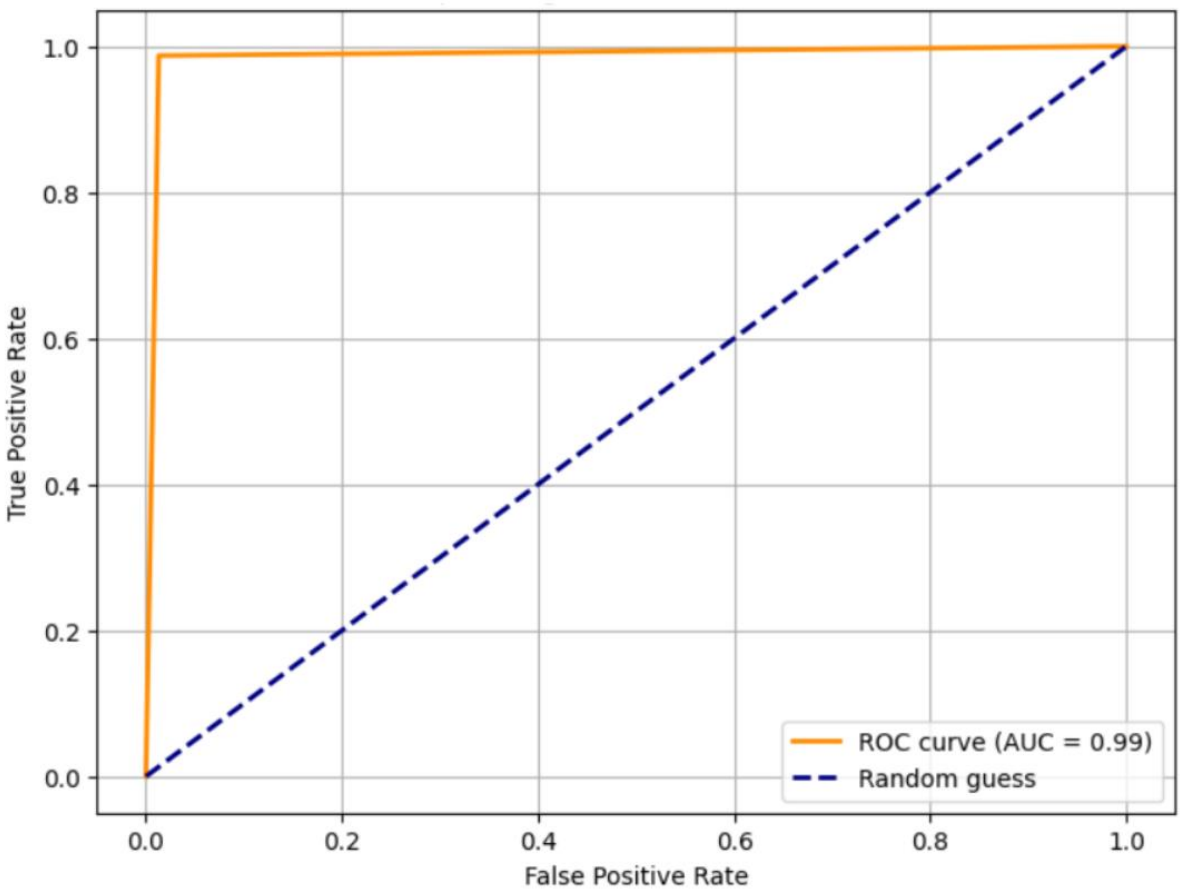
- Dòng 1 (0):
  - Có 271,959 mẫu được dự đoán đúng là 0.
  - Có 3,614 mẫu bị dự đoán nhầm là 1.

- Dòng 2 (1):
  - Có 4,247 mẫu bị dự đoán nhầm là 0.
  - Có 328,609 mẫu dự đoán đúng là 1.

	precision	recall	f1-score	support
0	0.98	0.99	0.99	275573
1	0.99	0.99	0.99	332856
accuracy			0.99	608429
macro avg	0.99	0.99	0.99	608429
weighted avg	0.99	0.99	0.99	608429

Hình 4.20. Classification Report ANN

- Độ chính xác tổng thể (Accuracy): 98.62%
  - Mô hình phân loại tốt
- Hiệu suất trên từng lớp:
  - Lớp 0 (e): Precision = 0.98, Recall = 0.99. Mô hình dự đoán rất tốt trên lớp này.
  - Lớp 1 (p): Precision = 0.99, Recall = 0.99. Mô hình dự đoán tốt trên lớp này.



Hình 4.21. Đường cong ROC của AdaBoost

- Điểm AUC:
  - AUC là 0,99 cho thấy mô hình gần như hoàn hảo trong việc phân biệt giữa các lớp dương và âm. Đây là một kết quả nổi bật.
- Hình dạng đường cong:
  - Đường cong gần như là một góc vuông hoàn hảo ở góc trên cùng bên trái, cho thấy mô hình đạt được Tỷ lệ dương tính thật cực kỳ cao với Tỷ lệ dương tính giả tối thiểu.

CHƯƠNG 5: GIẢI THÍCH VÀ BÀN LUẬN

5.1. Phân tích kết quả mô hình

Các mô hình Cây Quyết định (Decision Tree), Rừng Ngẫu nhiên (Random Forest), Gradient Boosting, AdaBoost hoạt động tương đối tốt với độ chính xác từ trên 90%. Những thuật toán này nắm bắt hiệu quả các mẫu phi tuyến và xử lý dữ liệu có chiều cao một cách hiệu quả. Ma trận nhầm lẫn và báo cáo phân loại cho thấy sự cải thiện tương đối về hiệu suất so với các mô hình trước khác, với điểm precision và recall cao, đồng thời giảm số lượng phân loại sai.

Mô hình Naive Bayes và mô hình Logistic Regression là hai mô hình hoạt động kém nhất trong số các mô hình được đánh giá. Điều này cho thấy mối quan hệ giữa các đặc trưng và biến mục tiêu có thể không thể tách rời tuyến tính, vốn là một giả định quan trọng của mô hình. Ma trận nhầm lẫn cho thấy một số lượng lớn các phân loại sai. Kết quả này được giải thích bởi bản chất tuyến tính của hồi quy Logistic và Naive Bayes, vốn khó nắm bắt các mối quan hệ phức tạp trong dữ liệu. Báo cáo phân loại cho thấy các giá trị precision và recall thấp, dẫn đến điểm F1 giảm.

Mạng nơ-ron nhân tạo (ANN) hoạt động có thể xem là tốt khi so với các mô hình khác. Nguyên nhân có thể do cấu trúc mạng, tham số huấn luyện, hoặc kích thước hạn chế của tập dữ liệu. ANN khá phù hợp dữ liệu phức tạp, phi tuyến tính hay mẫu lớn. Ma trận nhầm lẫn chỉ ra sự giảm số lượng phân loại sai so với hồi quy Logistic và Naive Bayes. Báo cáo phân loại cho thấy có sự cải thiện rõ rệt về precision và recall, kéo theo sự gia tăng của điểm F1. Tuy nhiên, tính phức tạp tính toán cao và độ nhạy với việc tinh chỉnh tham số của ANN có thể đã hạn chế hiệu quả của mô hình. Hiệu suất của ANN cho thấy kiến trúc và tham số huấn luyện được chọn chưa đủ để khai thác tối đa các mẫu phi tuyến trong tập dữ liệu.

Đánh giá trên cho thấy một xu hướng rõ ràng: các phương pháp ensemble như: Rừng Ngẫu nhiên, AdaBoost và Gradient Boosting cho kết quả vượt trội hơn so với các mô hình đơn giản như hồi quy Logistic và Naive Bayes.Mô hình Random Forest là mô hình đạt hiệu suất hợp lý với tập dữ liệu. AdaBoost cũng là mô hình có hiệu suất khá cao, thể hiện khả năng dự đoán chính xác việc nấm độc hay không. Hiệu suất vượt trội này có thể được lý giải bởi khả năng của AdaBoost trong việc kết hợp nhiều bộ phân loại yếu thành một bộ phân loại mạnh, nắm bắt hiệu quả các mối quan hệ phức tạp trong dữ liệu và giảm thiểu cả lỗi dương tính giả và âm tính giả. Hiệu suất tương tự của Gradient Boosting cho thấy các phương pháp ensemble rất phù hợp cho loại bài toán dự đoán này. Tuy nhiên, với tập dữ liệu này, mô hình ANN cũng cho hiệu suất cao.

5.2. Ảnh hưởng của việc tiền xử lý dữ liệu

- Xử lý mất cân bằng dữ liệu
    - Các mô hình phân loại dễ bị ảnh hưởng với các dữ liệu có sự mất cân bằng dữ liệu.
- Mã hóa biến mục tiêu và các biến category
    - Việc mã hóa cho các biến này giúp các mô hình học tốt hơn
- Chuẩn hóa dữ liệu
    - Đối với các mô hình như Logistic Regression, ANN, việc chuẩn hóa dữ liệu giúp mô hình học, đánh giá chính xác hơn và cải thiện hiệu suất mô hình.

Phân tích cho thấy giai đoạn tiền xử lý dữ liệu có ảnh hưởng khác nhau đối với các mô hình khác nhau. Trong khi Random Forest thể hiện hiệu suất mạnh mẽ bất kể có thực hiện giai đoạn tiền xử lý dữ liệu hay không, các mô hình như Gradient Boosting, AdaBoost, Decision Tree, Logistic Regression, Naive Bayes và ANN đã được cải thiện đáng kể, đặc biệt là trong việc cải thiện khả năng nhận diện các trường hợp không độc (class 0). Hiệu suất của Logistic Regression vẫn ổn định sau khi tiền xử lý. Điều này cho thấy việc lựa chọn các kỹ thuật tiền xử lý cần được cân nhắc kỹ lưỡng dựa trên các đặc điểm cụ thể của mô hình đã chọn. Đối với các mô hình tuyến tính như Logistic Regression, việc chọn các kỹ thuật đặc trưng hoặc các phương pháp tiền xử lý dữ liệu chuyên biệt có thể cần thiết để cải thiện hiệu suất dự đoán. Đối với các phương pháp ensemble dựa trên cây như Random Forest và Gradient Boosting, sự mạnh mẽ vốn có của chúng thường giảm thiểu nhu cầu về tiền xử lý phức tạp. Ngược lại, mô hình ANN lại rất nhạy cảm với việc chuẩn hóa dữ liệu và hiệu suất đã được cải thiện đáng kể nhờ tiền xử lý. Điều này này nhấn mạnh tầm quan trọng của giai đoạn tiền xử lý dữ liệu đối với từng mô hình một cách riêng biệt để đạt được hiệu suất tối ưu trong việc dự đoán.

5.3. Ý nghĩa của kết quả đối với nghiên cứu

Hiệu suất cao ổn định đạt được bởi một số mô hình Random Forest, AdaBoost, ANN, Decision Tree, Gradient Boosting đạt độ chính xác 90% hoặc cao hơn sau tối ưu hóa và tiền xử lý, chứng tỏ tính khả thi và tiềm năng của học máy trong việc dự đoán chính xác loại nấm có độc hay không bằng các đặc trưng vật lý. Điều này có những tác động đến thực tiễn như giảm rủi ro ngộ độc, giúp các cơ quan bảo vệ môi trường đưa ra các biện pháp bảo vệ sức khỏe cộng đồng và bảo tồn các loài nấm có giá trị.

Nghiên cứu làm nổi bật độ nhạy khác nhau của các thuật toán đối với tiền xử lý dữ liệu. Bên cạnh đó còn nhấn mạnh tầm quan trọng của việc xem xét các đặc điểm cụ thể của mỗi thuật toán khi thiết kế một pipeline học máy.

Nghiên cứu chứng minh vai trò quan trọng của tối ưu hóa siêu tham số trong việc đạt được hiệu suất tối đa. Các mô hình như SVM và AdaBoost đã có sự cải thiện đáng kể về độ chính xác sau khi điều chỉnh siêu tham số. Điều này làm nổi bật những hạn chế của việc chỉ dựa vào các thiết lập tham số mặc định và nhấn mạnh sự cần thiết của tối ưu hóa có hệ thống để khai thác tối đa tiềm năng của các thuật toán này. Mặc dù một số mô hình, như Logistic Regression và Decision Tree, chỉ có sự thay đổi tối thiểu sau tối ưu hóa, nhưng những cải thiện đáng kể ở các mô hình khác chứng minh tầm quan trọng của bước này trong quá trình phát triển mô hình.

5.4. Những hạn chế của nghiên cứu

Tập dữ liệu nghiên cứu còn nhiều vấn đề như: thiếu dữ liệu, mất cân bằng dữ liệu.

Đặc trưng vật lý không đủ để phân biệt. Ví dụ: Một số loài nấm độc có thể không có đặc trưng vật lý rõ ràng hoặc chúng có thể có sự biến đổi hình thái theo mùa hoặc độ tuổi, gây khó khăn cho việc phân loại dựa trên đặc trưng vật lý duy nhất.

Mô hình học máy có thể bị ảnh hưởng bởi các yếu tố không được tính đến.

5.5. Khả năng mở rộng và cải thiện

Khả năng mở rộng và phát triển của việc sử dụng machine learning (học máy) trong việc kiểm tra nấm có độc hay không có rất nhiều tiềm năng:

- Tích hợp, thu thập nhiều dữ liệu đa dạng phong phú.
- Tăng cường độ chính xác và đầy đủ của dữ liệu.
- Cải thiện và tối ưu hóa các mô hình học máy.

CHƯƠNG 6: KẾT LUẬN

6.1. Tóm tắt kết quả

Nghiên cứu này khám phá việc ứng dụng các thuật toán học máy để dự đoán nấm có độc hay không, tập trung vào ảnh hưởng của tiền xử lý dữ liệu và tối ưu hóa siêu tham số đến hiệu suất mô hình. Bắt đầu với một tập dữ liệu thô có kích thước (3116945, 22), đại diện cho 3116945 trường hợp nấm với 22 đặc trưng, chúng em đã tiến hành tiền xử lý dữ liệu để cải thiện hiệu suất mô hình. Thông qua quá trình phân tích đặc trưng và xử lý dữ liệu, một tập dữ liệu với 3042143 bản ghi và 21 đặc trưng đã được tạo ra. Nghiên cứu này là phân tích và so sánh hiệu suất mô hình qua hai kịch bản khác nhau: sử dụng dữ liệu đã qua xử lý mà không tối ưu hóa siêu tham số, và sử dụng dữ liệu đã qua xử lý với tối ưu hóa siêu tham số.

Kết quả cho thấy ảnh hưởng đáng kể của cả tiền xử lý dữ liệu và tối ưu hóa siêu tham số đến độ chính xác dự đoán. Tiền xử lý dữ liệu, bao gồm việc chuẩn hóa đặc trưng, mã hóa và xử lý giá trị thiếu, giá trị nhiễu đã cải thiện đáng kể hiệu suất của nhiều mô hình, đặc biệt là SVM, AdaBoost, Decision Tree và ANN.

Tối ưu hóa siêu tham số cũng đóng vai trò quan trọng trong việc cải thiện hiệu suất mô hình. Điều này nhấn mạnh tầm quan trọng của việc lựa chọn tham số cẩn thận để khai thác tối đa tiềm năng của các thuật toán này.

Nghiên cứu này nhấn mạnh sức mạnh dự đoán của các mô hình học máy trong việc dự đoán nấm độc hay không. Các phát hiện cung cấp những hiểu biết quý giá cho các chuyên gia trong vấn đề giảm thiểu ngộ độc, nhấn mạnh tầm quan trọng của một quy trình học máy rõ ràng, bao gồm cả việc chuẩn bị dữ liệu và tinh chỉnh mô hình để dự báo hủy bỏ chính xác và đáng tin cậy.

6.2. Hạn chế của nghiên cứu

Mặc dù nghiên cứu này cung cấp những hiểu biết về việc ứng dụng học máy trong dự đoán nấm độc hay không, nhưng cần phải thừa nhận những hạn chế của nó. Những hạn chế này mở ra những hướng nghiên cứu trong tương lai và góp phần tạo ra sự hiểu biết sâu sắc hơn về phạm vi và khả năng khái quát của nghiên cứu.

Kết quả của nghiên cứu dựa trên một tập dữ liệu duy nhất có kích thước (3116945, 22). Mặc dù tập dữ liệu này cung cấp đủ dữ liệu cho việc huấn luyện và đánh giá, nhưng nó chỉ đại diện cho một ngữ cảnh cụ thể và có thể không phản ánh đầy đủ sự đa dạng của các mẫu nấm trong thực tế. Cần tập trung tìm các đặc trưng tiêu biểu thể hiện được mục tiêu, cũng như thu thập thêm các mẫu dữ liệu khác.

Quá trình tối ưu hóa siêu tham số được thực hiện trong một không gian tìm kiếm xác định. Mặc dù đã có nỗ lực để khám phá một phạm vi giá trị hợp lý cho mỗi siêu tham số, nhưng có thể sự kết hợp tối ưu của các tham số nằm ngoài không gian đã định. Hơn nữa, quá trình tối ưu hóa này chỉ giới hạn ở hai kỹ thuật cụ thể là GridSearch và RandomSearch. Tuy nhiên ta có các phương pháp tối ưu hóa tiên tiến hơn, chẳng hạn như tối ưu hóa bầy đàn (PSO) hoặc thuật toán di truyền (GA), có thể mang lại những cải tiến vượt trội về hiệu suất khi áp dụng.

6.3. Khuyến nghị và hướng phát triển trong tương lai

Dựa trên kết quả và những hạn chế trong quá trình nghiên cứu, ta có thể đưa ra một số khuyến nghị và hướng phát triển trong tương lai nhằm để thúc đẩy việc ứng dụng học máy trong việc phân biệt nấm có độc hay không.

Để đảm bảo hiệu suất tối ưu của mô hình, các nghiên cứu trong tương lai nên khám phá các không gian tìm kiếm siêu tham số rộng hơn và các chiến lược tối ưu hóa khác nhau. Như áp dụng các kỹ thuật tối ưu hóa tiên tiến hơn, chẳng hạn: tối ưu hóa bầy đàn (PSO) hoặc thuật toán di truyền (GA), chúng có thể giúp khám phá những kết hợp tham số tốt hơn và cải thiện độ chính xác dự đoán mô hình. Bên cạnh đó, việc khám phá các công cụ học máy tự động (AutoML) có thể giúp đơn giản hóa quy trình tối ưu hóa và có thể khám phá ra các kiến trúc mô hình mới.

Triển khai một quy trình tiền xử lý dữ liệu mạnh mẽ trước khi áp dụng các mô hình vào dữ liệu mới. Quy trình này cần bao gồm việc kiểm tra các vấn đề chất lượng dữ liệu như giá trị thiếu, sự không nhất quán và ngoại lệ. Điều này sẽ đảm bảo tính ổn định của hệ thống và ngăn ngừa lỗi khi xử lý các tập dữ liệu mới. Hơn nữa, việc phân tích tầm quan trọng của các đặc trưng có thể cung cấp những hiểu biết chuyên sâu về các yếu tố quyết định nấm độc hay không.

Để đánh giá tính ổn định và khả năng thích ứng của các mô hình đã phát triển, các nghiên cứu trong tương lai nên khám phá hiệu suất của chúng trên các tập dữ liệu từ các nguồn khác nhau. Việc đánh giá hiệu suất mô hình trên nhiều tập dữ liệu sẽ giúp hiểu rõ hơn về khả năng khái quát của chúng và xác định những sai lệch hoặc hạn chế có thể có trong các ngữ cảnh cụ thể.

Để đảm bảo tính khả dụng lâu dài và hiệu quả của hệ thống, điều quan trọng là xây dựng một kiến trúc có khả năng mở rộng. Điều này bao gồm việc thiết kế hệ thống sao cho có thể tiếp nhận các đặc trưng và dữ liệu mới khi có sẵn. Một thiết kế linh hoạt và có khả năng mở rộng sẽ tạo điều kiện thuận lợi cho việc phát triển trong tương lai và tích hợp với các hệ thống.

TÀI LIỆU THAM KHẢO

[1] [Online]. Available: https://www.kaggle.com/competitions/playground-series-s4e8.

[2] Cole Stryker, Eda Kavlakoglu, "IBM," [Online]. Available: https://www.ibm.com/think/topics/artificial-intelligence.

[3] [Online]. Available: https://aws.amazon.com/vi/what-is/artificial-intelligence/.

[4] [Online]. Available: https://www.geeksforgeeks.org/ml-machine-learning/.

[5] [Online]. Available: https://www.sap.com/resources/what-is-deep-learning.

[6] [Online]. Available: https://www.geeksforgeeks.org/what-is-python/.

[7] [Online]. Available: https://www.geeksforgeeks.org/python-applications-in-real-world.

[8] [Online]. Available: https://www.geeksforgeeks.org/decision-tree.

[9] [Online]. Available: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning.

[10] [Online]. Available: https://www.geeksforgeeks.org/understanding-logistic-regression.

[11] [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers.

[12] [Online]. Available: https://www.almabetter.com/bytes/tutorials/data-science/adaboost-algorithm.

[13] [Online]. Available: https://www.geeksforgeeks.org/ml-gradient-boosting.

[14] [Online]. Available: https://www.databricks.com/glossary/artificial-neural-network.

[15] J. D. H. Z. X. Y. Yingying Wang, "Mushroom Toxicity Recognition Based on Multigrained Cascade Forest," 2020.

[16] A. Saryoko, E. P. Saputra, S. Nurajizah, M. Maulidah and N. Hidayati, "Product Prediction of Mushroom Agricultural Plants Using Machine Learning Techniques," IEEE, 2022.

[17] M. Bian, "Predicting Poisonous Mushrooms by Using Confusion Matrix Method," 2024.

[18] J. Sun, "Mushroom Poisonous Prediction Based on the Logistic Regression Model," 2023.

[19] Wacharaphol Ketwongsa,Sophon Boonlue, Urachart Kokaew, "A New Deep Learning Model for the Classification of Poisonous and Edible Mushrooms Based on Improved AlexNet Convolutional Neural Network," 2022.

[20] Baiming Zhang, Ying Zhao, Zhixiang Li, "Using deep convolutional neural networks to classify poisonous and edible mushrooms found in China," 2022.