**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN MÔN**

**HỌC MÁY**

**Đề tài : DỰ ĐOÁN CHẤT LƯỢNG RƯỢU VANG**

**Nhóm sinh viên thực hiện : Nhóm 14 - 65KTPM**

**Thành viên nhóm :**

1. Vương Tiến Dũng - 2351170586 - Nhóm trưởng

2. Đinh Thị Hoa – 2351170593 -

3. Vũ Hải Đăng - 2351170580

4. Trần Anh Tuấn - 2351170629

**Giảng viên hướng dẫn : TS Tạ Quang Chiểu**

**Hà Nội, năm** **2025**

# 

# MỤC LỤC

[MỤC LỤC ii](#_Toc212982656)

[LỜI CẢM ƠN iv](#_Toc212982657)

[PHẦN MỞ ĐẦU v](#_Toc212982658)

[PHẦN 1. TỔNG QUAN 1](#_Toc212982659)

[1. Giới thiệu về học máy 1](#_Toc212982660)

[1.1. Lịch sử và vai trò của Machine Learning 1](#_Toc212982661)

[1.2. Học có giám sát 2](#_Toc212982662)

[1.3. Học không có giám sát 2](#_Toc212982663)

[1.4. Bài toán Hồi quy 2](#_Toc212982664)

[1.5. Bài toán Phân lớp 3](#_Toc212982665)

[1.6. Bài toán Phân cụm 3](#_Toc212982666)

[2. Các phương pháp học máy được sử dụng trong đề tài 4](#_Toc212982667)

[2.1. Phương pháp Logistic Regression 4](#_Toc212982668)

[*2.1.1. Xây dựng Logistic Regression* 4](#_Toc212982669)

[*2.1.2. Ưu điểm và nhược điểm* 6](#_Toc212982670)

[2.2. Phương pháp Decision Tree (ID3) 7](#_Toc212982671)

[*2.2.1. Xây dựng Decision Tree (ID3)* 7](#_Toc212982672)

[*2.2.2. Ưu điểm và nhược điểm* 10](#_Toc212982673)

[2.3. Các độ đo để đánh giá mô hình dự đoán 11](#_Toc212982674)

[PHẦN 2. THỰC NGHIỆM 14](#_Toc212982675)

[1. Mô tả bài toán 14](#_Toc212982676)

[2. Mô tả tập dữ liệu của bài toán 14](#_Toc212982677)

[2.1 Các thuộc tính đầu vào bao gồm 15](#_Toc212982678)

[2.2 Thuộc tính đầu ra 15](#_Toc212982679)

[3. Viết ứng dụng 16](#_Toc212982680)

[3.1:Xử lí dữ liệu: 16](#_Toc212982681)

[*3.1.2 :Làm sạch dữ liệu:* 16](#_Toc212982682)

[*3.1.2: Phân chia tập dữ liệu(Train/Test):* 16](#_Toc212982683)

[3.2:Thuật toán Logistic Regression 17](#_Toc212982684)

[*3.2.1: Phân tích toán học* 17](#_Toc212982685)

[*3.2.2: Code triển khai* 17](#_Toc212982686)

[*3.2.3:Ma trận nhầm lẫn và kết quả* 24](#_Toc212982687)

[3.3 Thuật toán Decision Tree ( ID3 ) 27](#_Toc212982688)

[*3.3.1: Phân tích toán học* 27](#_Toc212982689)

[*3.3.2 : Code triển khai* 28](#_Toc212982690)

[*3.3.2.1: Thêm các thư viện cần dùng* 28](#_Toc212982691)

[*3.3.3 : Cây quyết định và kết quả* 32](#_Toc212982692)

[3.4 Demo giao diện người dùng 35](#_Toc212982693)

[*3.4.1. Giới thiệu* 35](#_Toc212982694)

[*3.4.2. Chức năng chính của giao diện* 37](#_Toc212982695)

[*3.4.3. Kết quả demo* 38](#_Toc212982696)

[3.5 Phân tích kết quả chương trình 39](#_Toc212982697)

[PHẦN 3. KẾT LUẬN 40](#_Toc212982698)

[TÀI LIỆU THAM KHẢO 43](#_Toc212982699)

# LỜI CẢM ƠN

Chúng em xin bày tỏ lòng biết ơn sâu sắc đến Ban Giám hiệu Trường Đại học Thủy Lợi cùng toàn thể quý thầy cô thuộc Khoa Công nghệ Thông tin, những người đã tận tình giảng dạy, truyền đạt kiến thức chuyên môn về học máy và trí tuệ nhân tạo, tạo nền tảng vững chắc để nhóm chúng em thực hiện dự án này. Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến TS.Tạ Quang Chiểu đã luôn đồng hành, định hướng, và đưa ra những góp ý quý báu, giúp nhóm hoàn thiện bài tập lớn một cách hiệu quả.

Chúng em cũng xin cảm ơn các nguồn tài liệu mở từ UCI Machine Learning Repository và Kaggle, đã cung cấp bộ dữ liệu Wine Quality chất lượng cao, là cơ sở quan trọng để nhóm triển khai các thuật toán học máy. Đồng thời, chúng em trân trọng sự hỗ trợ từ các tài liệu tham khảo và thư viện lập trình như scikit-learn, pandas, matplotlib, đã giúp đơn giản hóa quá trình phân tích và xây dựng mô hình.

Cuối cùng, chúng em xin ghi nhận tinh thần làm việc nhóm, sự đoàn kết, và nỗ lực không ngừng của các thành viên trong nhóm. Chính sự phối hợp chặt chẽ và chia sẻ trách nhiệm đã giúp chúng em vượt qua những thách thức để hoàn thành dự án này một cách trọn vẹn.

# PHẦN MỞ ĐẦU

Trong bối cảnh học máy ngày càng phát triển, việc áp dụng các thuật toán học máy vào các bài toán thực tiễn đã trở thành một xu hướng quan trọng. Nó ngày càng khẳng định vai trò quan trọng của mình việc giải quyết các bài toán từ y học, tài chính đến công nghiệp thực phẩm. Trong bối cảnh đó, dự án "Chất lượng rượu vang" được thực hiện nhằm khai thác bộ dữ liệu White Wine(Wine Quality) từ [UCI Machine Learning Repository](https://www.kaggle.com/datasets/piyushagni5/white-wine-quality) với 4898 mẫu sử dụng các đặc trưng hóa học như độ cồn, độ pH, hàm lượng đường dư để dự đoán chất lượng rượu vang. Đây là một bài toán phân loại, với mục tiêu phân loại chất lượng rượu thành ba lớp: thấp (dưới 5), trung bình (5-7), và cao (trên 7) trên thang điểm từ 0 đến 10. Nhóm chúng em hướng đến việc xây dựng và so sánh hai mô hình học có giám sát là Decision Tree (ID3) và Logistic Regression, đồng thời áp dụng K-Means Clustering trong khám phá dữ liệu để hiểu rõ hơn về đặc điểm của bộ dữ liệu. Dự án không chỉ nhằm dự đoán chính xác chất lượng rượu mà còn đánh giá hiệu suất các mô hình thông qua các độ đo như accuracy, F1-score, và ma trận nhầm lẫn, từ đó chọn ra phương pháp tối ưu.

Việc lựa chọn đề tài này xuất phát từ cả tính thực tiễn và giáo dục. Dự đoán chất lượng rượu vang có giá trị ứng dụng trong ngành công nghiệp đồ uống, giúp nhà sản xuất tối ưu hóa quy trình và hỗ trợ người tiêu dùng đánh giá sản phẩm. Bộ dữ liệu Wine Quality là một tập dữ liệu chuẩn, phong phú, phù hợp để thực hành các thuật toán học máy. Đồng thời, bài toán này mang tính thách thức với dữ liệu không cân bằng và yêu cầu kỹ năng từ xử lý dữ liệu, phân tích, đến triển khai mô hình. Đề tài cũng giúp nhóm củng cố kiến thức về học có giám sát, học không có giám sát, và rèn luyện kỹ năng lập trình Python với các thư viện như scikit-learn, pandas, matplotlib, cũng như khả năng làm việc nhóm và trình bày khoa học.

Báo cáo được tổ chức để trình bày rõ ràng quá trình thực hiện dự án. Phần đầu tiên cung cấp tổng quan về học máy, các bài toán phân loại, phân cụm, và giới thiệu hai thuật toán chính là Decision Tree và Logistic Regression. Phần thực nghiệm mô tả bài toán, tập dữ liệu, quy trình xử lý dữ liệu, triển khai thuật toán với phân tích toán học, code, và kết quả đánh giá. Phần kết luận tổng hợp hiệu suất mô hình, đề xuất phương pháp tốt nhất, và đưa ra hướng cải tiến như thử nghiệm Random Forest hoặc xử lý dữ liệu không cân bằng. Tài liệu tham khảo liệt kê các nguồn tài liệu và thư viện sử dụng, trong khi phụ lục bao gồm code triển khai và các biểu đồ trực quan như heatmap, scatter plot, và cây quyết định. Báo cáo này thể hiện nỗ lực của nhóm trong việc áp dụng học máy vào một bài toán thực tế, đồng thời phản ánh quá trình học hỏi và hợp tác để đạt được kết quả tốt nhất.

# PHẦN 1. TỔNG QUAN

## **1. Giới thiệu về học máy**

Học máy (Machine Learning) là một nhánh của trí tuệ nhân tạo (AI), tập trung vào việc phát triển các thuật toán cho phép máy tính học hỏi và đưa ra dự đoán hoặc quyết định dựa trên dữ liệu mà không cần được lập trình tường minh. Học máy dựa trên ý tưởng rằng hệ thống có thể tự động tìm ra các mẫu (patterns) trong dữ liệu và sử dụng chúng để giải quyết các bài toán thực tiễn.

Trong dự án "Chất lượng rượu vang", học máy được áp dụng để phân tích các đặc trưng hóa học (như độ cồn, độ pH, hàm lượng đường dư) nhằm dự đoán chất lượng rượu vang thông qua các thuật toán như Decision Tree và Logistic Regression. Học máy không chỉ giúp tự động hóa quá trình dự đoán mà còn cung cấp các công cụ mạnh mẽ để phân tích dữ liệu phức tạp, từ đó hỗ trợ ra quyết định trong nhiều lĩnh vực như công nghiệp, y tế, và tài chính.

### **1.1. Lịch sử và vai trò của Machine Learning**

Học máy ra đời từ những năm 1950, khi Alan Turing đề xuất ý tưởng về "máy học" có khả năng tự cải thiện hiệu suất. Những cột mốc quan trọng bao gồm sự phát triển của mạng nơ-ron nhân tạo (1958), thuật toán hồi quy tuyến tính (1960), và các phương pháp học cây quyết định như ID3 (1986). Từ những năm 2000, với sự bùng nổ của dữ liệu lớn (big data) và sức mạnh tính toán, học máy đã trở thành nền tảng cho các ứng dụng như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, và hệ thống khuyến nghị.

Vai trò của học máy ngày càng quan trọng trong thời đại công nghệ 4.0. Trong công nghiệp thực phẩm và đồ uống, học máy giúp phân tích dữ liệu để tối ưu hóa quy trình sản xuất và đánh giá chất lượng sản phẩm. Trong dự án này, học máy được sử dụng để dự đoán chất lượng rượu vang dựa trên các đặc trưng hóa học, cung cấp thông tin hữu ích cho nhà sản xuất và người tiêu dùng. Các thuật toán như Decision Tree và Logistic Regression cho phép nhóm phân loại chất lượng rượu một cách hiệu quả, trong khi K-Means Clustering giúp khám phá các mẫu ẩn trong dữ liệu.

### **1.2. Học có giám sát**

Học có giám sát (Supervised Learning) là một phương pháp học máy, trong đó mô hình được huấn luyện trên một tập dữ liệu có nhãn (labeled data), tức là mỗi mẫu dữ liệu đi kèm với một giá trị đầu ra đã biết. Mục tiêu là học một hàm ánh xạ từ đầu vào (đặc trưng) đến đầu ra (nhãn). Học có giám sát được chia thành hai loại chính: hồi quy (dự đoán giá trị liên tục) và phân lớp (dự đoán nhãn rời rạc).

Trong dự án "Chất lượng rượu vang", học có giám sát được áp dụng thông qua hai thuật toán phân lớp: Decision Tree (ID3) và Logistic Regression. Bộ dữ liệu Wine Quality cung cấp các đặc trưng hóa học (đầu vào) và điểm chất lượng (nhãn), được chuyển thành ba lớp: thấp (dưới 5), trung bình (5-7), và cao (trên 7). Các thuật toán này học từ dữ liệu huấn luyện để dự đoán nhãn lớp cho các mẫu rượu mới, đảm bảo độ chính xác cao và phù hợp với bài toán thực tiễn.

### **1.3. Học không có giám sát**

Học không có giám sát (Unsupervised Learning) là phương pháp học máy, trong đó mô hình làm việc với dữ liệu không có nhãn, nhằm tìm ra các mẫu hoặc cấu trúc ẩn trong dữ liệu. Các bài toán phổ biến bao gồm phân cụm (clustering) và giảm chiều dữ liệu (dimensionality reduction). Phân cụm nhóm các mẫu dữ liệu tương tự nhau thành các cụm, trong khi giảm chiều giúp đơn giản hóa dữ liệu mà vẫn giữ được thông tin quan trọng.

Trong dự án, học không có giám sát được sử dụng thông qua thuật toán K-Means Clustering để phân cụm dữ liệu rượu vang dựa trên các đặc trưng hóa học. K-Means giúp nhóm khám phá các nhóm chất lượng rượu tiềm năng (ví dụ: thấp, trung bình, cao) mà không cần nhãn, từ đó cung cấp insight cho việc phân tích dữ liệu và hỗ trợ xây dựng các mô hình phân lớp.

### **1.4. Bài toán Hồi quy**

Bài toán hồi quy (Regression) thuộc học có giám sát, nhằm dự đoán một giá trị liên tục dựa trên các đặc trưng đầu vào. Ví dụ, trong hồi quy tuyến tính, mô hình tìm một hàm tuyến tính để dự đoán giá trị đầu ra (như giá nhà hoặc nhiệt độ). Các độ đo đánh giá phổ biến bao gồm Mean Squared Error (MSE) và R².

Trong dự án "Chất lượng rượu vang", bài toán hồi quy không được chọn làm trọng tâm vì điểm chất lượng rượu (từ 0 đến 10) được chuyển thành bài toán phân loại (thấp, trung bình, cao) để phù hợp với mục tiêu phân tích. Tuy nhiên, hiểu biết về hồi quy giúp nhóm nhận thức được các phương pháp khác có thể áp dụng nếu dự đoán điểm số liên tục thay vì lớp chất lượng.

### **1.5. Bài toán Phân lớp**

Bài toán phân lớp (Classification) thuộc học có giám sát, nhằm dự đoán nhãn lớp rời rạc cho các mẫu dữ liệu. Ví dụ, phân loại email thành spam hoặc không spam. Các thuật toán phổ biến bao gồm Logistic Regression, Decision Tree, và Support Vector Machine. Các độ đo đánh giá bao gồm accuracy, F1-score, và ma trận nhầm lẫn.

Dự án này tập trung vào bài toán phân lớp, với mục tiêu phân loại chất lượng rượu vang thành ba lớp: thấp (dưới 5), trung bình (5-7), và cao (trên 7). Decision Tree (ID3) và Logistic Regression được chọn vì tính hiệu quả và phù hợp với dữ liệu. Decision Tree cung cấp khả năng trực quan hóa quy tắc phân loại, trong khi Logistic Regression đơn giản và hiệu quả với dữ liệu tuyến tính.

### **1.6. Bài toán Phân cụm**

Bài toán phân cụm (Clustering) thuộc học không có giám sát, nhằm nhóm các mẫu dữ liệu tương tự nhau thành các cụm mà không cần nhãn. Thuật toán phổ biến nhất là K-Means Clustering, trong đó các mẫu được gán vào các cụm dựa trên khoảng cách đến tâm cụm. Các độ đo như Silhouette Score được dùng để đánh giá chất lượng phân cụm.

Trong dự án, K-Means Clustering được áp dụng trong giai đoạn khám phá dữ liệu (EDA) để phân tích các mẫu rượu vang thành các cụm dựa trên đặc trưng hóa học. Kết quả phân cụm giúp nhóm hiểu rõ hơn về sự phân bố của dữ liệu, xác định các nhóm chất lượng tiềm năng, và cung cấp insight để hỗ trợ các mô hình phân lớp như Decision Tree và Logistic Regression.

## **2. Các phương pháp học máy được sử dụng trong đề tài**

### **2.1. Phương pháp Logistic Regression**

* Logistic Regression là một thuật toán phân loại có giám sát phổ biến, dùng để dự đoán xác suất thuộc về các lớp rời rạc dựa trên tổ hợp tuyến tính của các đặc trưng đầu vào. Trong bài toán “Dự đoán chất lượng rượu vang”, Logistic Regression được sử dụng để phân loại chất lượng rượu (đã được ánh xạ thành các lớp: thấp / trung bình / cao) dựa trên các đặc trưng hóa học như volatile acidity, residual sugar, pH, sulphates, alcohol,...

#### ***2.1.1. Xây dựng Logistic Regression***

* ***Mô hình cơ bản***
* Với bài toán nhị phân, Logistic Regression mô tả xác suất lớp dương như sau:
* Trong đó là vector trọng số, là bias, là vector đặc trưng, và là hàm sigmoid.
* Với bài toán **đa lớp** (ở đề tài ta có 3 lớp: thấp/trung bình/cao), ta dùng **one-vs-rest (OvR)** hoặc **softmax (multinomial)**. Softmax cho xác suất của lớp k:

* ***Hàm mất mát và tối ưu***
* Hàm mất mát (đa lớp) thường dùng là **cross-entropy loss**:

* Tối ưu hóa bằng các thuật toán như **l-bfgs**, **liblinear**, **saga** hoặc **newton-cg** (thư viện scikit-learn đã triển khai sẵn). Thường kết hợp **regularization** để tránh overfitting — L2 mặc định (có thể chọn L1).
* ***Tiền xử lý cần thiết***
* **Xử lý missing values**: mean imputation hoặc drop (nếu số record rất ít).
* **Loại bỏ outliers** (nếu cần) — IQR hoặc z-score để giảm ảnh hưởng.
* **Chuẩn hóa/scale**: dùng StandardScaler hoặc MinMaxScaler — bắt buộc/nên làm để các feature có cùng thang đo và giúp thuật toán hội tụ tốt.
* **Chia tập**: train/test (ví dụ 80/20) và có thể dùng stratified split nếu dữ liệu mất cân bằng giữa các lớp.
* **Xử lý imbalance** : oversampling (SMOTE), class\_weight='balanced' trong LogisticRegression, hoặc dùng metric phù hợp (F1\_macro).
* ***Siêu tham số quan trọng (sklearn)***
* C: inverse regularization strength (giá trị nhỏ → phạt mạnh → độ phức tạp mô hình giảm).
* penalty: 'l2' (mặc định) hoặc 'l1' (nếu dùng solver hỗ trợ).
* solver: 'lbfgs', 'liblinear', 'saga', 'newton-cg'. (Với multiclass='multinomial' thường dùng 'l-bfgs' hoặc 'saga'.)
* multi\_class: 'ovr' hoặc 'multinomial'.
* max\_iter: tăng nếu không hội tụ.
* ***Tối ưu hóa***
* Dùng **GridSearchCV** hoặc **RandomizedSearchCV** để tìm C, solver, penalty, multi\_class tốt nhất theo metric mong muốn (ví dụ f1\_macro hoặc accuracy).
* Dùng **cross-validation (k=5)** để đánh giá ổn định.
* ***Quy trình triển khai***
* Load dữ liệu (UCI/Kaggle) → kiểm tra missing/outlier.
* Map label quality → 3 lớp (ví dụ <5: thấp, 5-7: trung bình, >7: cao).
* Chia train/test (stratified).
* Scale features bằng StandardScaler.
* Huấn luyện LogisticRegression với GridSearchCV (scoring=f1\_macro hoặc accuracy).
* Đánh giá trên test: accuracy, precision, recall, f1-score, confusion matrix.
* Nếu imbalance nặng → thử class\_weight='balanced' hoặc SMOTE.

#### ***2.1.2. Ưu điểm và nhược điểm***

* ***Ưu điểm***
* **Đơn giản, giải thích được**: hệ số w cho biết đóng góp của từng đặc trưng đến xác suất — dễ giải thích.
* **Tính ổn định và nhanh**: huấn luyện nhanh với dữ liệu kích thước vừa phải.
* **Ít overfitting hơn** so với mô hình phức tạp nếu dùng regularization.
* **Dễ tích hợp** (cả trong pipeline và triển khai web/Streamlit).
* ***Nhược điểm***
* **Giả sử tuyến tính**: hiệu quả nhất khi ranh giới giữa các lớp tương đối tuyến tính; nếu dữ liệu phức tạp, phi tuyến, Logistic có thể kém.
* **Nhạy với outliers**: cần tiền xử lý.
* **Với lớp nhiều và không cân bằng**, cần điều chỉnh (class\_weight/SMOTE) hoặc kết hợp mô hình khác (Random Forest, XGBoost) để cải thiện.
* **Không tự động bắt tương tác phức tạp giữa features** — cần feature engineering hoặc model phức tạp hơn.

### **2.2. Phương pháp Decision Tree (ID3)**

* Decision Tree (ID3) là một thuật toán học máy có giám sát, thuộc nhóm phân loại và hồi quy, hoạt động theo nguyên lý chia để trị (divide and conquer). Mô hình xây dựng một cây quyết định với các nút quyết định (dựa trên đặc trưng), nhánh (giá trị của đặc trưng), và nút lá (dự đoán lớp).
* Trong bài toán chất lượng rượu vang, Decision Tree được dùng để phân loại rượu thành 3 lớp: thấp, trung bình, cao dựa trên 11 đặc trưng hóa học. Thuật toán ID3 sử dụng Information Gain làm tiêu chí chọn đặc trưng để chia nhánh, giúp xây dựng cây tối ưu.

#### ***2.2.1. Xây dựng Decision Tree (ID3)***

* ***Mô hình cơ bản*** 
  + Với bài toán **phân loại đa lớp** (3 lớp: thấp / trung / bình, cao), Decision Tree (ID3) mô tả quá trình dự đoán như một **cây nhị phân hoặc đa hướng**
* Trong đó:
* Mỗi **nút quyết định** kiểm tra một điều kiện trên một đặc trưng.
* Mỗi **nhánh** đại diện cho một giá trị (hoặc khoảng giá trị) của đặc trưng đó.
* Mỗi **nút lá** chứa **quyết định cuối cùng** – nhãn lớp có tần suất cao nhất trong tập con dữ liệu đến được nút đó.
* Cụ thể, tại mỗi nút , thuật toán sẽ lựa chọn **đặc trưng A** và **ngưỡng chia**  sao cho **Information Gain** đạt giá trị **lớn nhất**
* Công thức tính Information Gain được xác định như sau:

IG (A, ) = () -

* Trong đó:
* tập con thỏa mãn *A .*
* tập con thỏa mãn *A> .*
* entropy của nút .
* ***Phân tích toán học***
* **Entropy** của tập dữ liệu S:
* Trong đó:
* số lượng lớp (class) trong tập dữ liệu.
* xác suất mẫu thuộc lớp trong tập .
* **Information Gain** khi chia theo **đặc trưng A:**
* Trong đó:
* tập các giá trị có thể có của thuộc tính .
* tập con của mà tại đó .
* Quy trình chia nhánh:

Tại mỗi nút, thuật toán chọn đặc trưng A có Information Gain cao nhất để tách dữ liệu, nó lặp lại cho đến khi:

* Tất cả các mẫu trong nút thuộc cùng một lớp, hoặc.
* Không còn đặc trưng nào có thể chia tiếp, hoặc.
* Đạt đến độ sâu nút tối đa (max\_deep) được quy định.
* ***Tiền xử lý cần thiết***
* **Kiểm tra missing values**: Dùng data.isnull().sum() → Không có giá trị thiếu trong bộ dữ liệu Wine Quality.
* **Xử lý missing values**: Thay giá trị trung bình của cột → data.fillna(data.mean(), inplace=True) → đảm bảo không mất mẫu.
* **Loại bỏ outliers**: Dùng phương pháp **IQR** → loại các mẫu nằm ngoài khoảng (Q1 - 1.5\*IQR, Q3 + 1.5\*IQR).
* **Chuẩn hóa dữ liệu**: Dùng **StandardScaler** → đưa tất cả đặc trưng về phân phối chuẩn (mean=0, std=1).
* **Chuyển nhãn thành 3 lớp**: Dùng pd.cut với ngưỡng <5, 5-7, >7 → thấp, trung bình, cao.
* **Chia tập train/test**: tỷ lệ 80/20, dùng stratify=y\_class để giữ nguyên lớp tỷ lệ.
* **Xử lý imbalance** (nếu cần): Dùng class\_weight=’balanced’ trong DecisionTreeClassifier → tự động tăng trọng số lớp thiểu số (low, high).
* ***Siêu tham số quan trọng (sklearn)***
* **Criterion**: tiêu chí đánh giá chia nhánh ‘entropy’ (ID3) hoặc ‘gini’ (CART).
* **Max\_depth**: độ sâu tối đa của cây (nhỏ giúp giảm độ phức tạp, tránh overfitting).
* **Min\_samples\_split**: số lượng mẫu tối thiểu để chia một nút (kiểm soát kích thước cây).
* **Min\_samples\_leaf**: số lượng mẫu tối thiểu ở nút lá (ngăn cây quá phân mảnh).
* **Class\_weight**: cân bằng trọng số giữa các lớp
* **Random\_state**: cố định ngẫu nhiên (tái lập kết quả thử nghiệm).
* ***Tối ưu hóa***
* Sử dụng **GridSearchCV** hoặc **RandomizedSearchCV** để tìm bộ siêu tham số tốt nhất:
* **Các tham số thường tối ưu**: max\_depth, min\_samples\_split, min\_samples\_leaf, criterion
* **Thước đo đánh giá** (scoring metric): f1\_macro hoặc accuracy.
* **Áp dụng cross-validation** (k=5) để đảm bảo tính ổn định và tổng quát hóa.
* ***Quy trình triển khai***
* **Nạp dữ liệu**: Sử dụng tập dữ liệu có sẵn trong thư viện scikit-learn.
* **Huấn luyện mô hình**: Áp dụng DecisionTreeClassifier với tiêu chí chia nhánh là entropy (ID3) để xây dựng mô hình trên tập huấn luyện.
* **Tối ưu siêu tham số**: Sử dụng **GridSearchCV** với thang đo đánh giá F1-score macro để tìm ra tổ hợp tham số tối ưu, bao gồm: max\_depth, min\_samples\_split, criterion, min\_samples\_leaf.
* **Trực quan hóa mô hình**: Dùng hàm plot\_tree trong scikit-learn để biểu diễn cây quyết định, giúp quan sát rõ quá trình chia nhánh và đặc trưng được chọn.
* **Đánh giá mô hình:** Thực hiện kiểm tra thông qua các chỉ số: Accuracy, Precision, Recall, F1-score, và Confusion Matrix.
* **Xử lý dữ liệu mất cân bằng** (nếu có): Áp dụng class\_weight=’balanced’ hoặc giới hạn độ sâu tối đa (max\_depth) để giảm hiện tượng overfitting.

#### ***2.2.2. Ưu điểm và nhược điểm***

* ***Ưu điểm***
* **Đơn giản và dễ biểu diễn :** Mô hình dạng cây rất trực quan, có thể chuyển thành các câu “Nếu... thì...” giúp người dùng không chuyên dễ hiểu.
* **Không yêu cầu chuẩn hoá dữ liệu** : Decision Tree có thể làm việc với dữ liệu ở nhiều dạng (số, danh mục) mà không cần chuyển đổi phức tạp.
* **Xử lý tốt dữ liệu phi tuyến:** Có thể mô tả mối quan hệ phức tạp giữa các biến mà không cần giả định tuyến tính.
* **Phù hợp cho bài toán phân loại và hồi quy:** Cây quyết định linh hoạt, có thể áp dụng cho cả hai loại bài toán.
* ***Nhược điểm***
* **Dễ bị overfitting:** Khi cây quá sâu, mô hình có thể học chi tiết nhiễu của dữ liệu huấn luyện → giảm khả năng tổng quát hóa.
* **Không ổn định:** Một thay đổi nhỏ trong dữ liệu đầu vào có thể làm thay đổi cấu trúc cây đáng kể.
* **Thiên lệch với thuộc tính có nhiều giá trị:** ID3 có xu hướng chọn thuộc tính có nhiều nhánh hơn, dẫn đến chia nhỏ dữ liệu không cần thiết.
* **Khó xử lý dữ liệu liên tục chính xác:** Cần phải chia khoảng giá trị liên tục thành từng vùng rời rạc, có thể làm giảm độ chính xác.

### **2.3. Các độ đo để đánh giá mô hình dự đoán**

* Trong học máy, việc đánh giá mô hình là bước quan trọng để xác định mức độ hiệu quả của thuật toán trên dữ liệu thực tế. Đối với bài toán phân loại chất lượng rượu vang, nhóm sử dụng bốn độ đo chính: **Accuracy**, **F1-score**, **Confusion Matrix**, và **Cross-validation**. Mỗi độ đo mang ý nghĩa riêng, giúp đánh giá mô hình một cách toàn diện và khách quan hơn.
* ***Accuracy (Độ chính xác)***
* **Accuracy** là tỷ lệ số lượng dự đoán đúng trên tổng số mẫu kiểm tra.
* Công thức :
* Trong đó :
* **TP (True Positive)**: số mẫu dương dự đoán đúng
* **TN (True Negative)**: số mẫu âm dự đoán đúng
* **FP (False Positive)**: số mẫu âm bị dự đoán nhầm thành dương
* **FN (False Negative)**: số mẫu dương bị dự đoán nhầm thành âm
* Accuracy cho biết mô hình dự đoán đúng bao nhiêu phần trăm tổng thể. Tuy nhiên, trong trường hợp dữ liệu **mất cân bằng giữa các lớp** (ví dụ: phần lớn rượu ở mức trung bình), chỉ số này có thể **đánh lừa** vì mô hình chỉ cần đoán lớp chiếm đa số cũng đạt Accuracy cao.
* ***F1-score (Điểm cân bằng Precision và Recall)***
* **F1-score** là trung bình điều hòa giữa **Precision** (độ chính xác) và **Recall** (độ bao phủ), giúp đánh giá mô hình trong trường hợp dữ liệu không cân bằng.
* Trong đó :
* **Precision** đo lường tỷ lệ mẫu được dự đoán là dương thật sự đúng.
* **Recall** đo lường khả năng mô hình phát hiện hết các mẫu dương.
* Chỉ số **F1-score** càng cao (gần 1) cho thấy mô hình vừa chính xác vừa toàn diện. Trong bài toán này, nhóm sử dụng **F1-macro** (trung bình đều giữa các lớp) để đảm bảo công bằng giữa rượu “thấp”, “trung bình”, và “cao”.
* ***Confusion Matrix (Ma trận nhầm lẫn)***
* **Confusion Matrix** thể hiện số lượng mẫu mà mô hình dự đoán đúng hoặc nhầm giữa các lớp.
* Ví dụ với 3 lớp “thấp”, “trung bình”, “cao”, ma trận có dạng 3×3:

|  |  |  |  |
| --- | --- | --- | --- |
| Thực tế \ Dự đoán | Thấp | Trung bình | Cao |
| Thấp |  |  | FP₁₃ |
| Trung bình |  |  | FP₂₃ |
| Cao |  |  | TP₃ |

* Đường chéo chính (TP) thể hiện số mẫu dự đoán đúng.
* Các phần tử ngoài đường chéo biểu thị số lượng nhầm lẫn giữa các lớp.
* Ma trận này cho phép nhóm **quan sát trực quan mô hình đang nhầm ở đâu** (ví dụ: rượu “cao” bị đoán nhầm thành “trung bình”), từ đó cải thiện bằng cách cân bằng lớp hoặc chọn mô hình khác.
* ***Cross-validation (Kiểm định chéo)***
* **Cross-validation (k-fold)** là kỹ thuật đánh giá mô hình bằng cách chia dữ liệu thành *k phần* bằng nhau (thường k=5 hoặc 10).
* Quy trình:

1. Chia dữ liệu thành *k* phần.
2. Huấn luyện mô hình trên *k−1* phần, kiểm tra trên phần còn lại.
3. Lặp lại *k* lần, mỗi lần đổi phần kiểm tra.
4. Lấy **điểm trung bình** của các lần làm kết quả cuối cùng.

* Cross-validation giúp đánh giá mô hình **ổn định và tránh phụ thuộc vào cách chia train/test ngẫu nhiên**, đảm bảo độ tin cậy của kết quả.
* ***Ứng dụng trong dự án***
* Trong dự án “**Dự đoán chất lượng rượu vang**”, các độ đo trên được sử dụng như sau:
* **Accuracy**: dùng để đo tổng thể mức độ chính xác mô hình Logistic Regression và Decision Tree.
* **F1-macro**: dùng làm **thước đo chính trong GridSearchCV** để tìm siêu tham số tối ưu cho Logistic Regression, đảm bảo đánh giá công bằng giữa các lớp.
* **Confusion Matrix**: giúp nhóm quan sát mô hình đang dự đoán sai ở lớp nào — ví dụ rượu “cao” bị nhầm thành “trung bình”.
* **Cross-validation (5-fold)**: dùng trong GridSearchCV để đảm bảo mô hình **tổng quát hóa tốt**, không phụ thuộc vào một lần chia dữ liệu.
* Nhờ kết hợp các độ đo trên, nhóm có thể đánh giá toàn diện cả **hiệu suất (Accuracy)** lẫn **sự cân bằng (F1-score)** của mô hình, từ đó xác định thuật toán phù hợp nhất cho việc dự đoán chất lượng rượu vang trắng.

# PHẦN 2. THỰC NGHIỆM

## **1. Mô tả bài toán**

Bài toán "Chất lượng rượu vang" tập trung vào việc dự đoán chất lượng của rượu vang dựa trên các đặc trưng hóa học, sử dụng bộ dữ liệu Wine Quality từ [UCI Machine Learning Repository](https://www.kaggle.com/datasets/piyushagni5/white-wine-quality). Đây là một bài toán phân loại có giám sát, trong đó mục tiêu là dự đoán chất lượng rượu vang thuộc một trong ba lớp: thấp (điểm chất lượng dưới 5), trung bình (điểm từ 5 đến 7), và cao (điểm trên 7), dựa trên thang điểm từ 0 đến 10. Bộ dữ liệu bao gồm 11 đặc trưng hóa học như độ cồn (alcohol), độ pH, hàm lượng đường dư (residual sugar), và hàm lượng axit citric, cùng với nhãn chất lượng. Nhóm sử dụng hai thuật toán học máy là Decision Tree (ID3) và Logistic Regression để xây dựng mô hình dự đoán, đồng thời áp dụng K-Means Clustering trong giai đoạn khám phá dữ liệu (EDA) để phân tích các mẫu chất lượng tiềm năng. Mục tiêu là đạt được độ chính xác cao trong dự đoán và so sánh hiệu suất giữa các mô hình để chọn ra phương pháp tối ưu.

## **2. Mô tả tập dữ liệu của bài toán**

Nguồn dữ liệu sử dụng cho bài toán dự đoán chất lượng rượu vang được lấy từ bộ dữ liệu Wine Quality của [UCI Machine Learning Repository](https://www.kaggle.com/datasets/piyushagni5/white-wine-quality) (Wine Quality Data Set). Bộ dữ liệu gốc gồm hai phần : **rượu vang đỏ** (Red Wine) và **rượu vang trắng** (White Wine)

Tuy nhiên,trong đề tài này nhóm chỉ sử dụng phần dữ liệu rượu vang trắng (White Wine) gồm 4898 mẫu để đảm bảo tính đồng nhất,độ ổn định và độ chính xác cao hơn trong quá trình huấn luyện và đánh giá mô hình dự đoán chất lượng rượu.

### **2.1 Các thuộc tính đầu vào bao gồm**

* **Fixed acidity** : Lượng axit cố định (đơn vị g/dm³)
* **Volatile acidity** : Lượng axit bay hơi ( đơn vị g/dm³)
* **Citric acid** : Hàm lượng axit citric (đơn vị g/dm³)
* **Residual sugar** : Lượng đường còn lại sau khi lên men (đơn vị g/dm³)
* **Chlorides** : Hàm lượng muối (NaCl) trong rượu (đơn vị g/dm³)
* **Free sulfur dioxide** : Lượng SO₂ tự do (đơn vị mg/dm³)
* **Total sulfur dioxide** : Tổng lượng SO₂ (đơn vị mg/dm³)
* **Density** : Khối lượng riêng của rượu (đơn vị g/cm³)
* **pH** : Độ axit của rượu
* **Sulphates** : Mức độ sulfat (đơn vị g/dm³)
* **Alcohol** : Nồng độ cồn (đơn vị % vol )

### **2.2 Thuộc tính đầu ra**

* **Quality** : điểm chất lượng của rượu (thang điểm 0-10) được đánh giá bởi chuyên gia ( dữ liệu gốc )
* Để đơn giản hóa bài toán phân loại, giá trị này sẽ được chia thành 3 nhóm :
* Rượu kém chất lượng ( điểm < 5 )
* Rượu trung bình ( điểm 5-7 )
* Rượu chất lượng cao ( điểm > 7 )
* Trong dự án này,cả hai mô hình Logistics Regression và ID3 được áp dụng cho cùng bài toán phân loại chất lượng rượu trắng thành ba mức: Thấp, Trung bình và Cao.Dù khác biệt về bản chất (một mô hình **tuyến tính** và một mô hình **cây quyết định phi tuyến**), nhưng cả hai đều có cùng đầu ra phân loại nên hoàn toàn có thể so sánh các chỉ số đánh giá hiệu năng như Accuracy, Precision, Recall và F1-score.”

## **3. Viết ứng dụng**

Chương trình dự đoán chất lượng rượu vang được phát triển bằng ngôn ngữ lập trình Python, sử dụng các thư viện học máy và xử lý dữ liệu phổ biến như scikit-learn, pandas, và matplotlib. Ứng dụng bao gồm các bước chính: xử lý dữ liệu (làm sạch, chuẩn hóa, giảm chiều), khám phá dữ liệu (EDA), xây dựng và huấn luyện mô hình (Decision Tree và Logistic Regression), và đánh giá hiệu suất mô hình thông qua các độ đo như accuracy, F1-score, và ma trận nhầm lẫn.

Quy trình triển khai được thực hiện trên môi trường Jupyter Notebook để dễ dàng tích hợp code, trực quan hóa dữ liệu, và trình bày kết quả. Dữ liệu được chia thành tập huấn luyện (80%) và tập kiểm tra (20%) để đảm bảo đánh giá khách quan. Mô hình Decision Tree (ID3) được xây dựng để tạo ra các quy tắc phân loại trực quan, trong khi Logistic Regression được tối ưu hóa bằng Grid Search để cải thiện hiệu suất. Kết quả được trình bày dưới dạng biểu đồ (heatmap, scatter plot, cây quyết định) và báo cáo số liệu, giúp nhóm phân tích và so sánh hiệu quả của các mô hình.

### **3.1:Xử lí dữ liệu:**

#### ***3.1.2 :Làm sạch dữ liệu:***

* Loại bỏ các kí tự đặc biệt hoặc khoảng trắng trong tên cột để thống nhất định dạng
* Loại bỏ những bản ghi bị thiếu dữ liệu hoặc chứa giá trị không hợp lệ
* Chuyển toàn bộ dữ liệu về dạng số thực(numeric) để thuận tiện cho các phép tính toán và mô hình học máy

#### ***3.1.2: Phân chia tập dữ liệu(Train/Test):***

* Chia tập dữ liệu thành tập train và tập test để đánh giá hiệu suất của mô hình.Dữ liệu thường được chia với tỉ lệ 80/20 hoặc 70/30. Trong bài toán dự đoán chất lượng rượu vang sẽ chia với tỷ lệ 80/20 trong đó 80% mẫu được sử dụng để huấn luyện mô hình và 20% dùng để kiểm tra.

### **3.2:Thuật toán Logistic Regression**

#### ***3.2.1: Phân tích toán học***

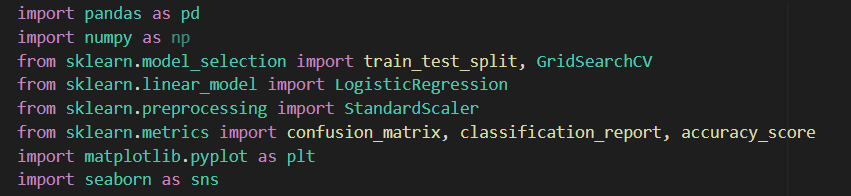
* Áp dụng Logistic Regression để phân loại chất lượng rượu trắng thành 3 nhóm: thấp( q
* Mô hình hóa xác suất bằng hàm sigmoid:

Trong đó:

* x=[x1​,x2​,...,xn​] là vector các đặc trưng
* w=[w1,w2,...,wn] là vector trọng số cần học
* b là hệ số sai lệch (bias)
* Mô hình được huấn luyện tối ưu hàm mất mát(loss function) - đo lường sai lệch giữa dự đoán các đặc trưng và nhãn thật(quality). Nếu mô hình dự đoán đúng(xác suất cao cho lớp tốt) thì hàm mất mát nhỏ và nếu mô hình dự đoán sai(xác suất thấp cho lớp tốt) thì hàm mất mát lớn.
* Để tránh overfitting cho mô hình ta thêm vào hàm mất mát hệ số phạt L2 với λ là hệ số điều chỉnh mức phạt(nếu λ lớn thì mô hình phạt các trọng số lớn thì sẽ đơn giản hơn và tránh overfitting). Trong sklearn thì tham số được biểu diễn bằng C = 1/λ.

#### ***3.2.2: Code triển khai***

*3.2.2.1: Thêm các thư viện cần dùng*

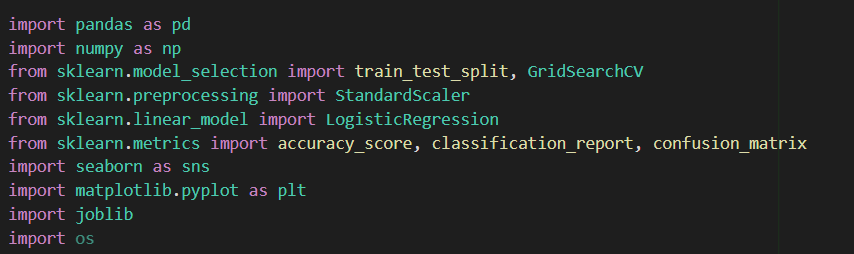
****

* Thư viện pandas: xử lí dữ liệu dạng bảng, đọc file “.csv”.
* Thư viện numpy: hỗ trợ tính toán số học, mảng.
* Thư viện học máy train\_test\_split: chia dữ liệu thành 2 tập train và test.
* Thư viện học máy GridSearchCV: tìm bộ tham số tối ưu cho mô hình bằng cách thử nhiều tổ hợp.
* Thư viện học máy StandardScaler: chuẩn hóa dữ liệu.
* Thư viện học máy confusion\_matrix: ma trận nhầm lẫn.
* Thư viện học máy classification\_report: báo cáo về độ chính xác, độ nhạy,F1-score.
* Thư viện học máy accuracy\_score: tính độ chính xác tổng thể của mô hình.
* Thư viện matplotlib.pyplot: để vẽ biểu đồ đơn giản như biểu đồ cột, đường, scatter.
* Thư viện seaborn: vẽ biểu đồ phức tạp hơn như heatmap, boxplot,...

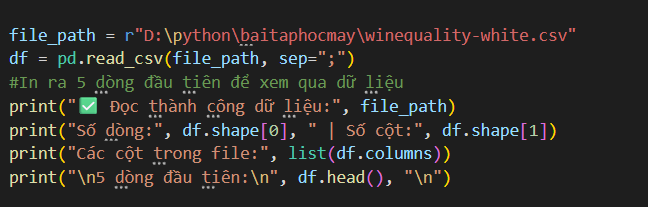
*3.2.2.2: Phần thân chương trình:*

#### ***Dùng thư viện :***

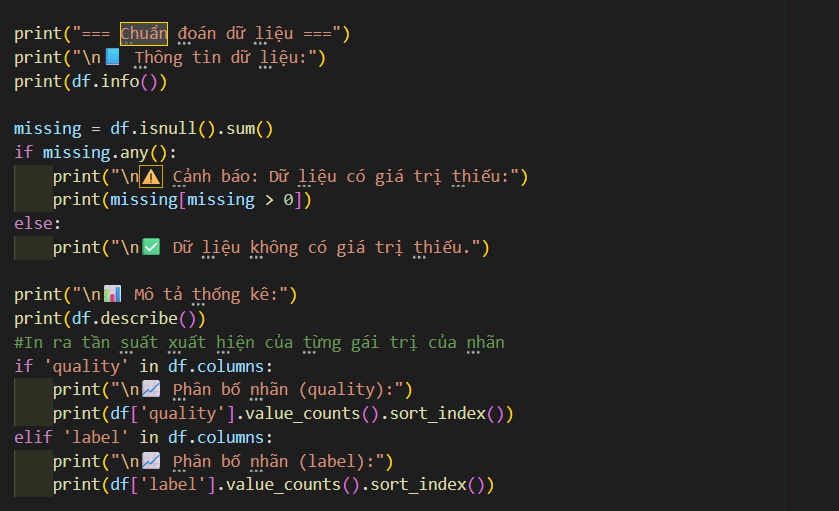
* Import thư viện:



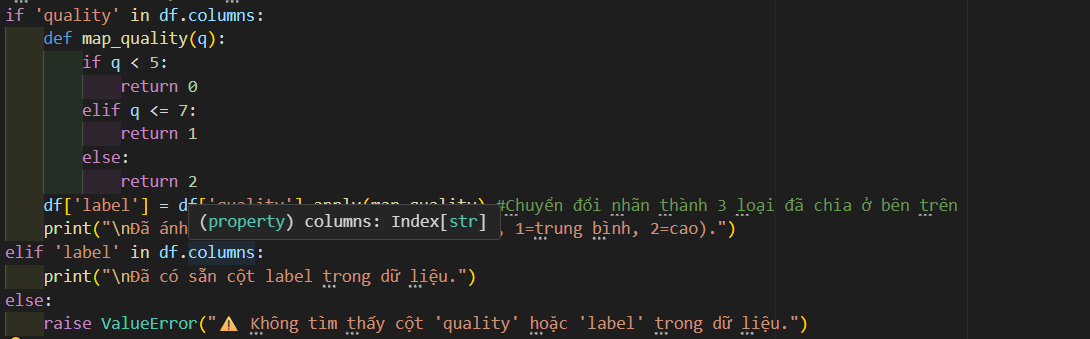
* Đọc dữ liệu



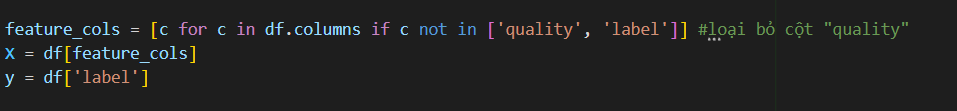
* Chuẩn hóa dữ liệu



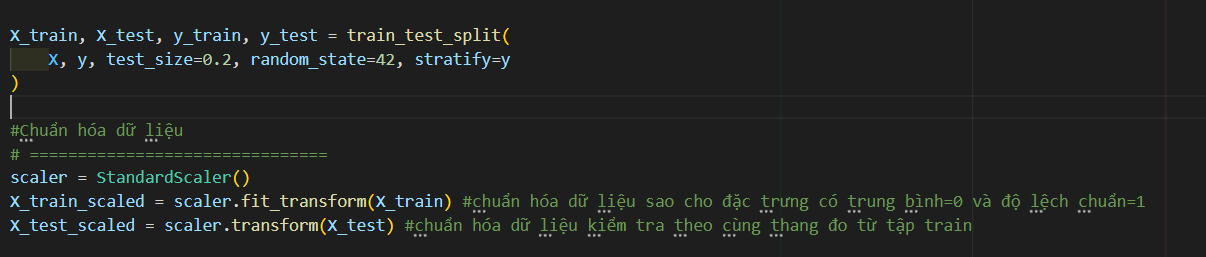
* Xử lí nhãn



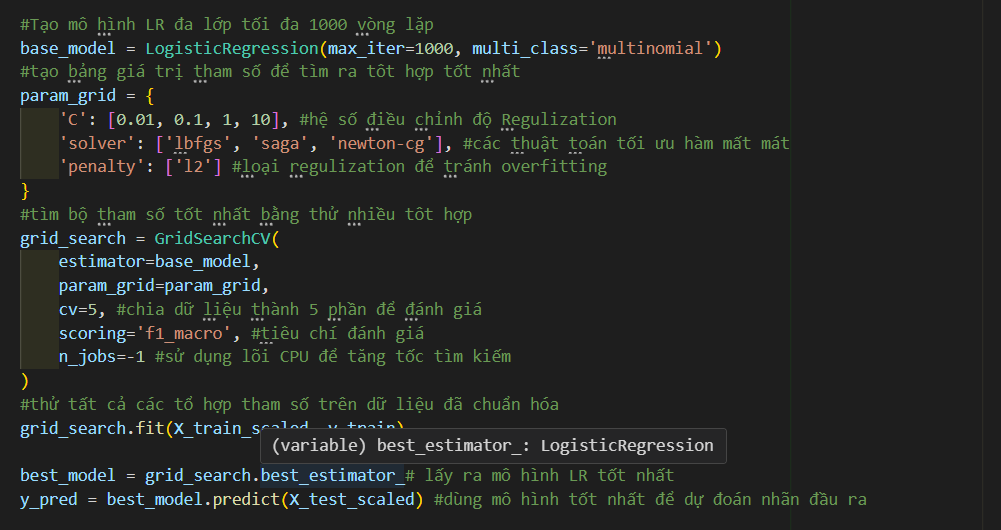
* Chuẩn bị dữ liệu đầu vào



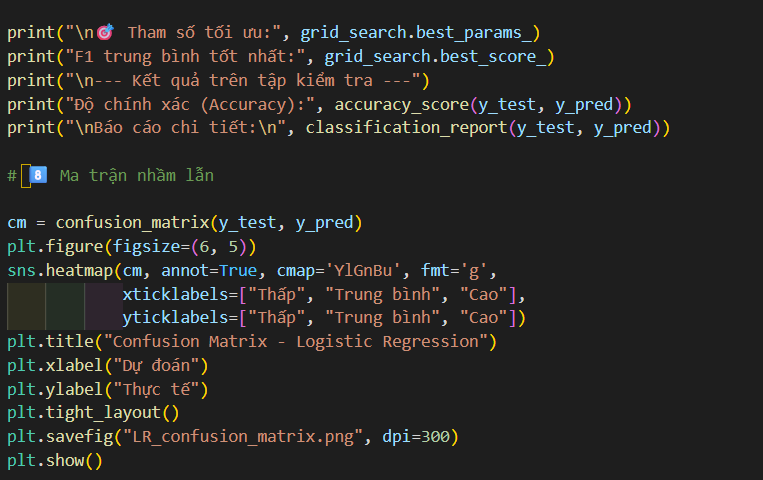
* Chia tập dữ liệu



* Huấn luyện mô hình

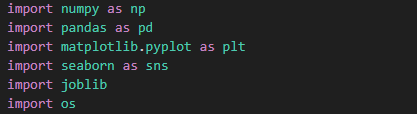


* Kết quả và ma trận nhầm lẫn

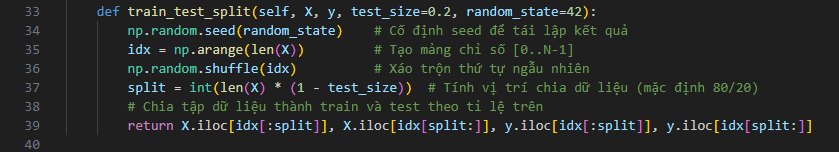
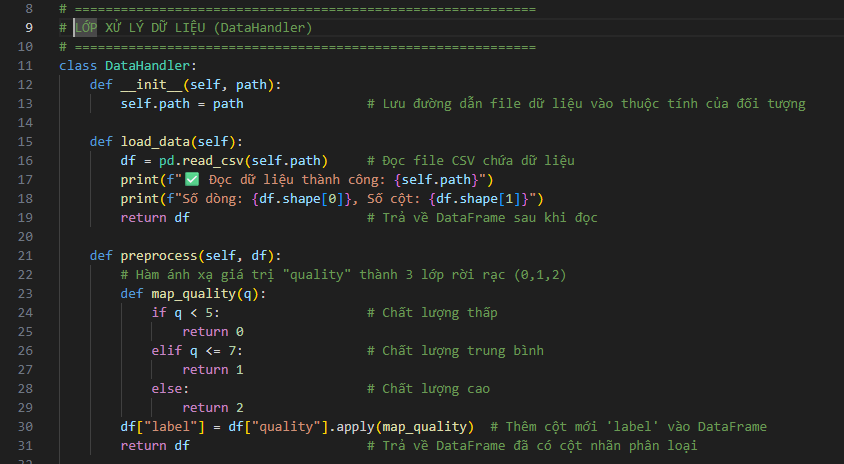


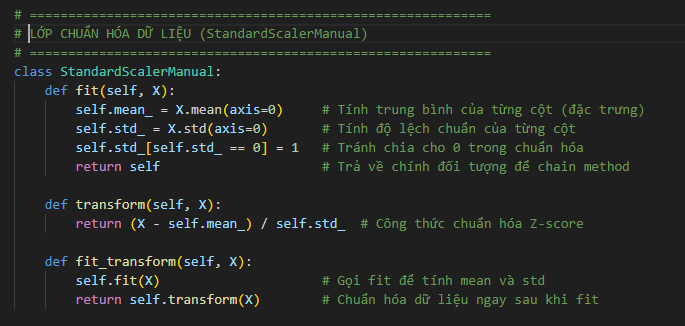
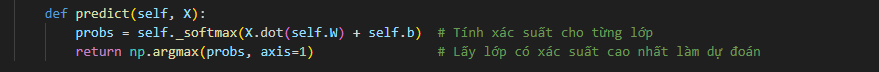
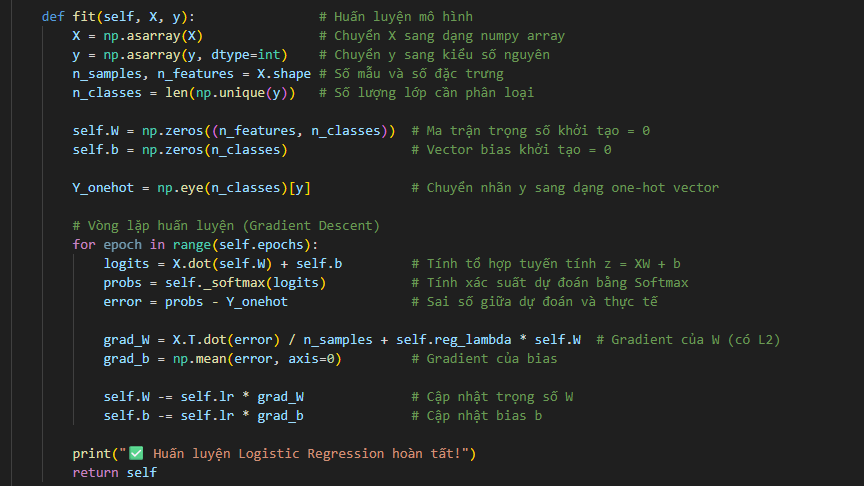
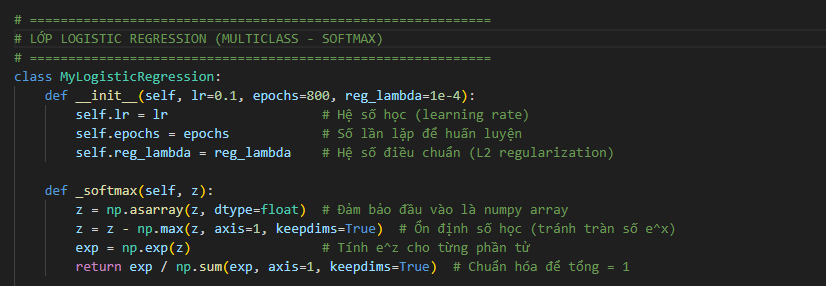
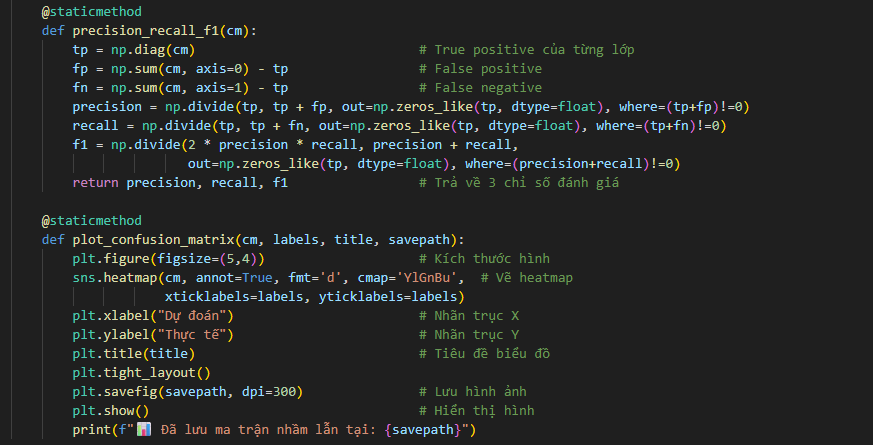
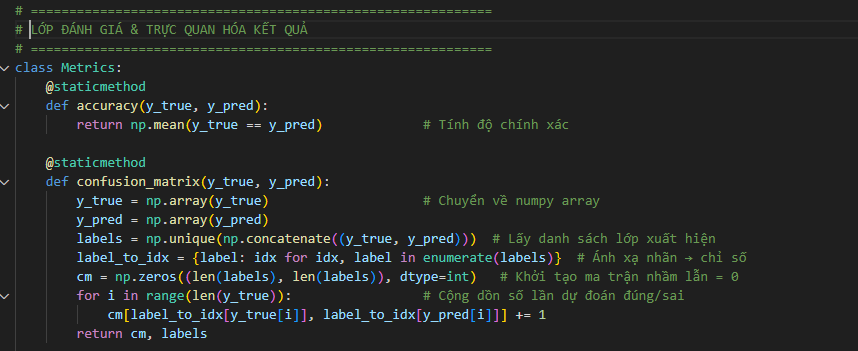
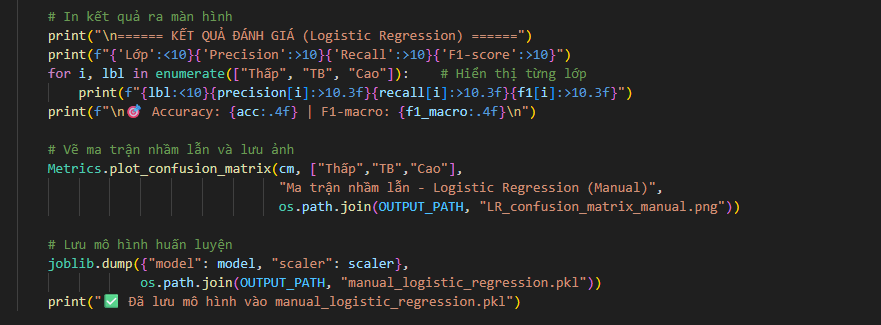
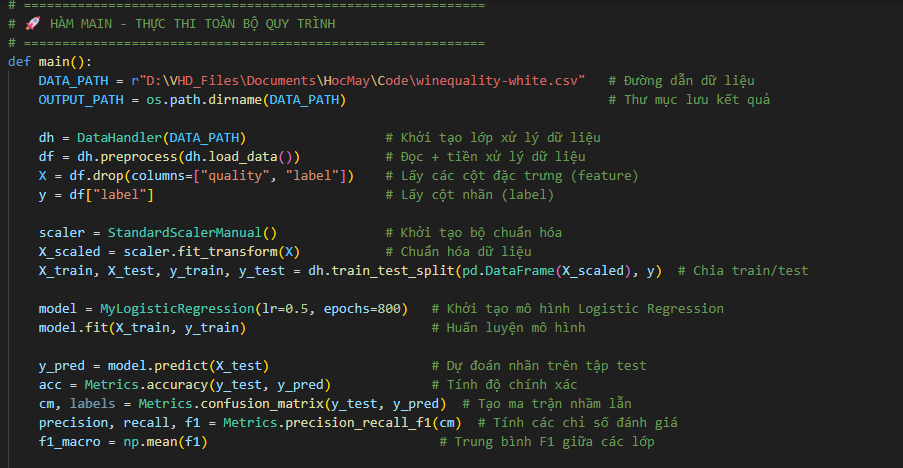
#### ***Code thủ công:***

* Import thư viện



* Định nghĩa lớp để xử lí dữ liệu

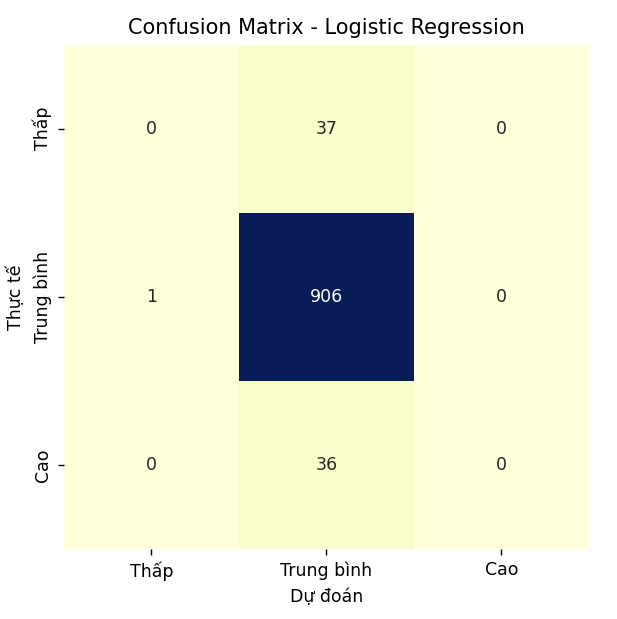


* Lớp chuẩn hóa dữ liệu
* Thuật toán Logistic Regression
* Hàm đánh giá, vẽ ma trận nhầm lẫn
* Hàm Main - Thực thi chương trình

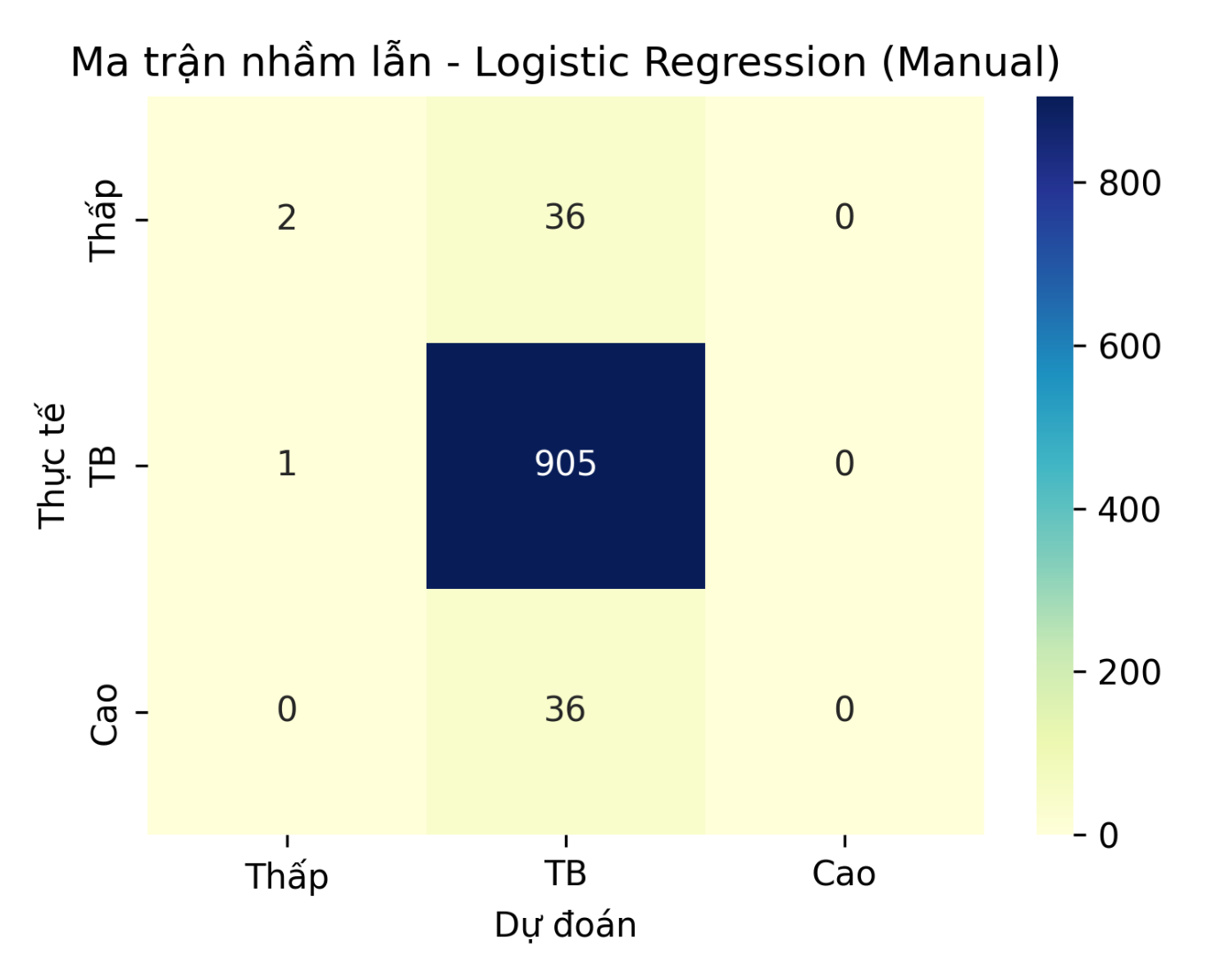
#### ***3.2.3:Ma trận nhầm lẫn và kết quả***

*3.2.3.1: Ma trận nhầm lẫn*

#### ***Dùng thư viện***

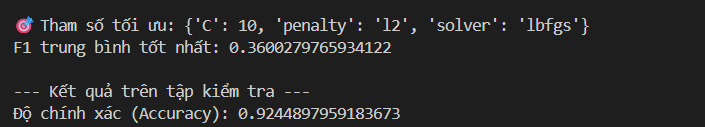


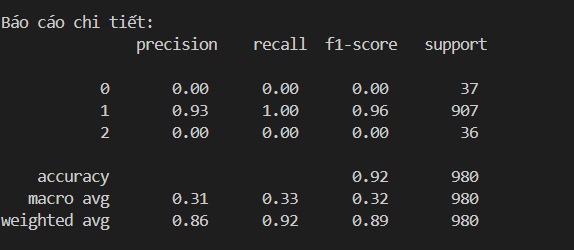
#### ***Code thủ công***



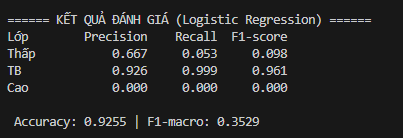
*3.2.3.2: Kết quả:*

#### ***Dùng thư viện:***





#### ***Code thủ công:***



### **3.3 Thuật toán Decision Tree ( ID3 )**

#### ***3.3.1: Phân tích toán học***

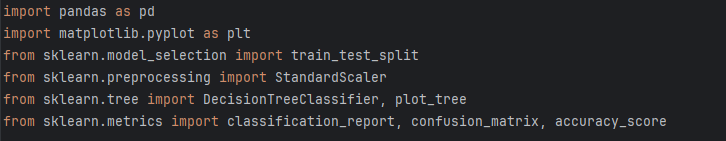
* Thuật toán ID3(Iterative Dichotomiser) xây dựng cây quyết định bằng cách chọn thuộc tính có khả năng phân chia dữ liệu tốt nhất tại mỗi bước
* Mức độ “**tốt nhất**” được đo bằng chỉ số Information Gain dựa trên khái niệm Entropy
* ***Entropy - Độ hỗn loạn của dữ liệu :***

- Công thức tính Entropy cho một tập dữ liệu S :

* Trong đó :
* số lượng lớp (class) trong tập dữ liệu.
* xác suất mẫu thuộc lớp trong tập .
* ***Information Gain - Lượng thông tin thu được***
  + Sau khi chọn thuộc tính A để chia dữ liệu,ta tính được lượng thông tin thu được bằng :
  + Trong đó :
* tập các giá trị có thể có của thuộc tính .
* tập con của mà tại đó .
* Thuộc tính có Information Gain cao nhất sẽ được chọn làm nút chia tại bước đó

#### ***3.3.2 : Code triển khai***

#### *3.3.2.1: Thêm các thư viện cần dùng*

****

* Thư viện pandas: xử lý dữ liệu dạng bảng, đọc file “.csv”.
* Thư viện matplotlib.pyplot: vẽ biểu đồ cơ bản như đường, cột, scatter, cây quyết định.
* Thư viện học máy train\_test\_split: chia dữ liệu thành tập huấn luyện (train) và tập kiểm tra (test).
* Thư viện học máy StandardScaler: chuẩn hóa dữ liệu về trung bình = 0, độ lệch chuẩn = 1.
* Thư viện học máy DecisionTreeClassifier: xây dựng mô hình cây quyết định phân loại.
* Thư viện học máy plot\_tree: trực quan hóa cây quyết định dưới dạng đồ họa.
* Thư viện học máy classification\_report: báo cáo chi tiết precision, recall, F1-score cho từng lớp.
* Thư viện học máy confusion\_matrix: ma trận nhầm lẫn, so sánh nhãn thật với nhãn dự đoán.
* Thư viện học máy accuracy\_score: tính độ chính xác tổng thể của mô hình.

*3.3.2.2: Phần thân chương trình:*

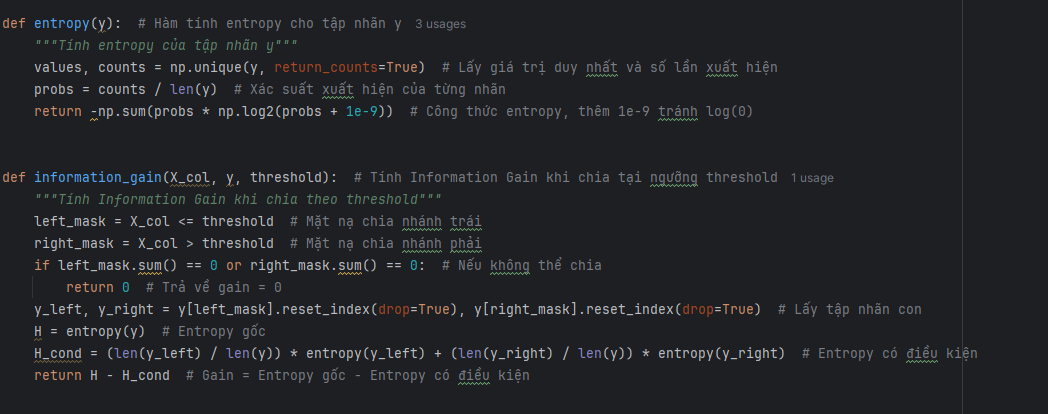
* *Dùng thư viện*

A screenshot of a computer program

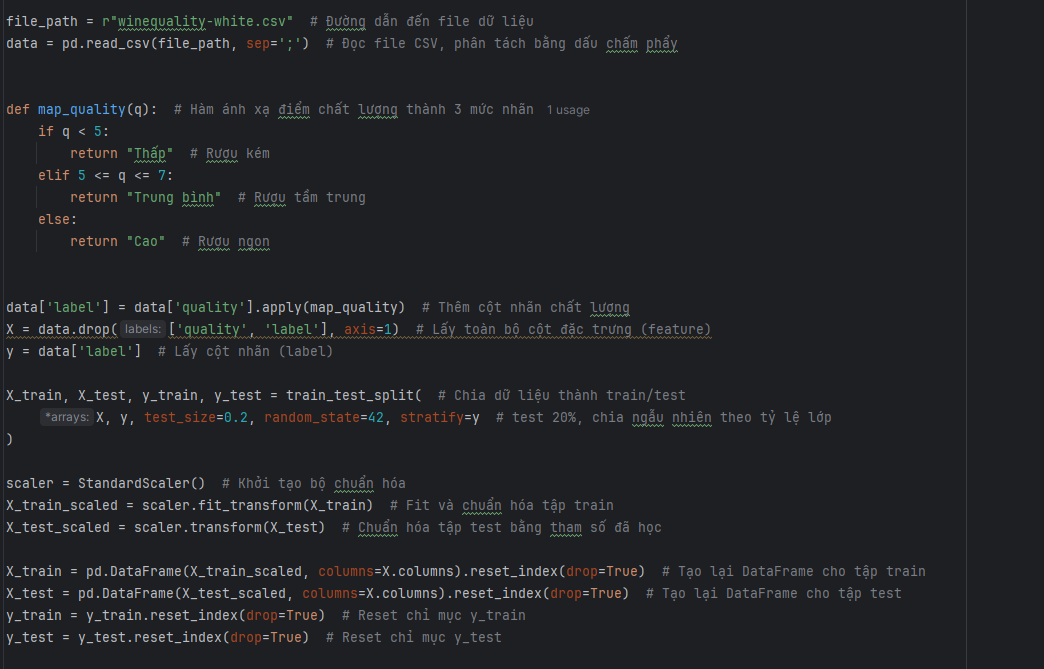
AI-generated content may be incorrect.A computer screen with colorful text

AI-generated content may be incorrect.

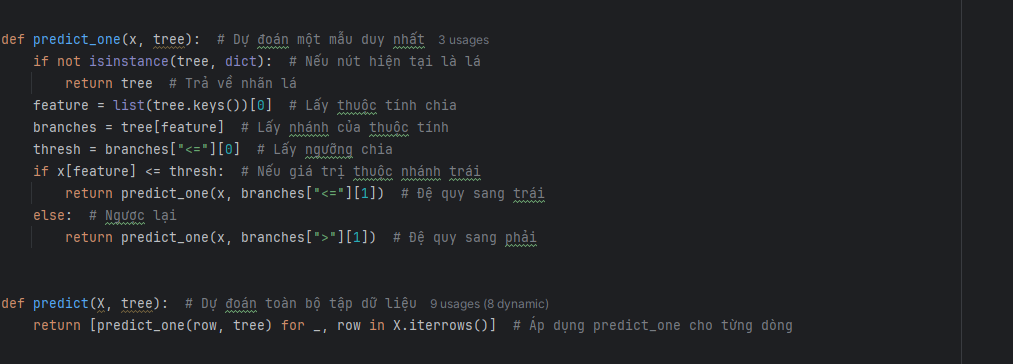
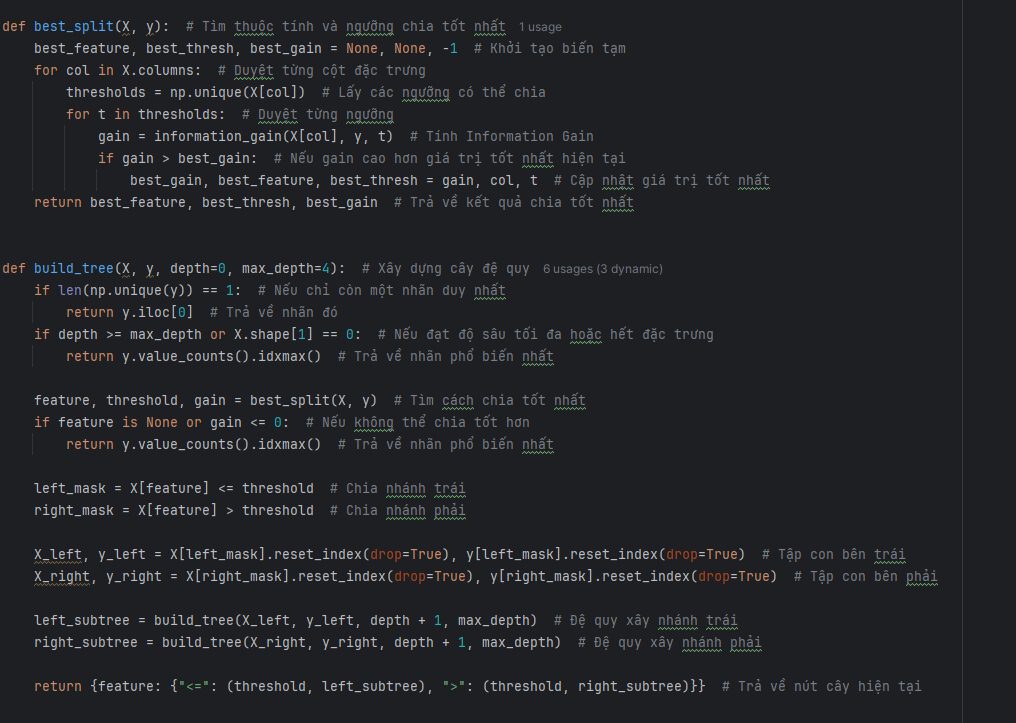
* *Code thủ công*
* *Xây dựng các đánh giá thuật toán*



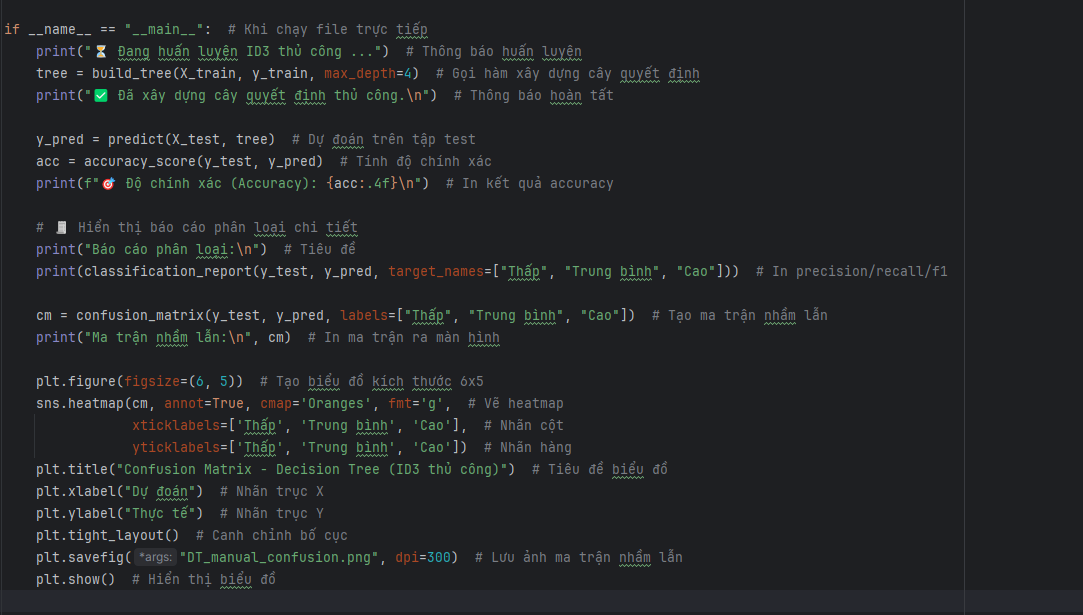
* *Định nghĩa lớp xử lý dữ liệu*



* *Thuật toán decision tree (thủ công)*



* *Hàm main*



#### ***3.3.3 : Cây quyết định và kết quả***

*3.3.3.1: Ma trận nhầm lẫn:*

*A computer screen with colorful text

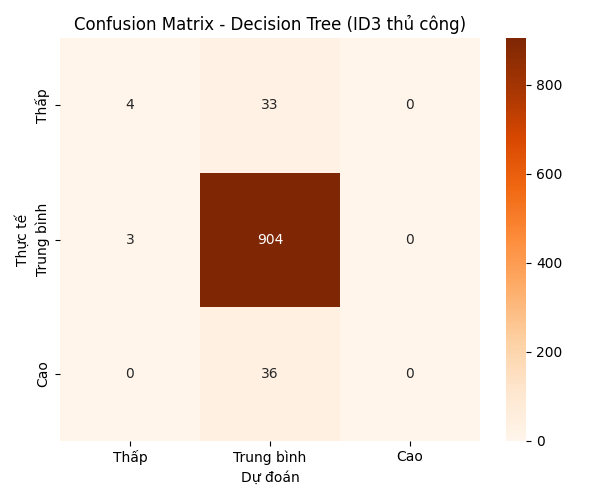
AI-generated content may be incorrect.*

* *Kết quả ma trận nhầm lẫn dùng thư viện*

A diagram of a decision tree

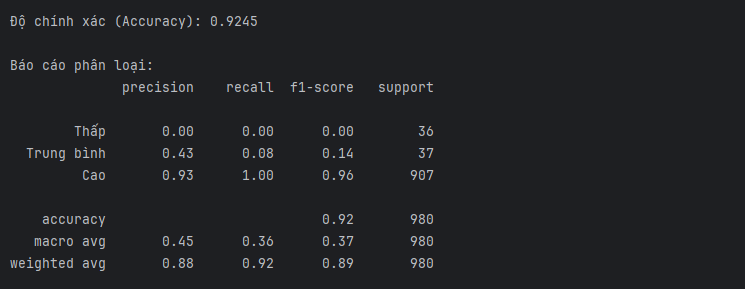
AI-generated content may be incorrect.

* *Kết quả ma trận nhầm lẫn code thủ công*

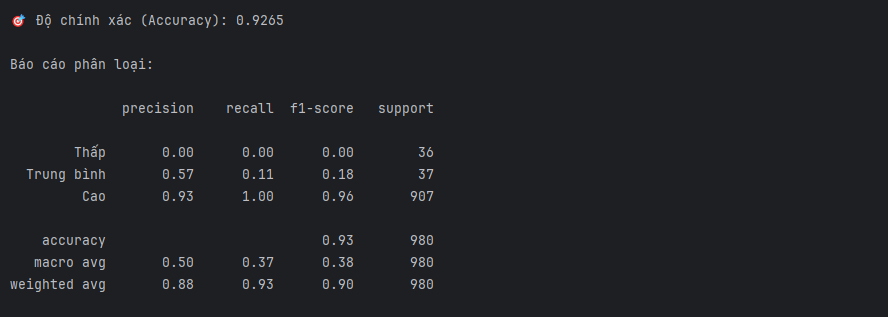


*3.3.3.2: Kết quả:*

* *Dùng thư viện*



* *Code thủ công*



*3.3.3.3 : Cây quyết định dùng thư viện*

**A diagram of a diagram

AI-generated content may be incorrect.**

### **3.4 Demo giao diện người dùng**

#### ***3.4.1. Giới thiệu***

* Nhóm đã xây dựng một **ứng dụng giao diện đồ họa (UI)** bằng **Python – thư viện Tkinter**, giúp người dùng có thể **nhập các chỉ số hóa học của rượu** và **nhận kết quả dự đoán chất lượng** dựa trên mô hình Machine Learning đã huấn luyện.
* Ứng dụng được thiết kế trực quan, dễ sử dụng, có thể chuyển đổi giữa **hai mô hình dự đoán khác nhau**:
* ***Logistic Regression (Hồi quy Logistic)***  
  A screenshot of a computer

  AI-generated content may be incorrect.
* ***Decision Tree (Cây quyết định – ID3)***

**A screenshot of a computer

AI-generated content may be incorrect.**

#### ***3.4.2. Chức năng chính của giao diện***

* + - 1. *Nhập dữ liệu đầu vào*
* Người dùng nhập 11 chỉ số hóa học của rượu trắng, bao gồm:
* Fixed Acidity, Volatile Acidity, Citric Acid, Residual Sugar, Chlorides, Free Sulfur Dioxide, Total Sulfur Dioxide, Density, pH, Sulphates, Alcohol.
  + - 1. *Chọn mô hình học máy*
* Giao diện có một **combobox** cho phép người dùng chọn mô hình dự đoán:
* Logistic Regression
  + Decision Tree
    - 1. *Dự đoán chất lượng rượu*
* Khi nhấn nút **“DỰ ĐOÁN”**, chương trình sẽ:
* Lấy dữ liệu nhập → Chuẩn hóa (nếu cần).
  + - * + Nạp mô hình .pkl tương ứng.
        + Dự đoán **chất lượng rượu** (Thấp, Trung bình, hoặc Cao).
* Tính toán và hiển thị **độ tin cậy (%)** của dự đoán.
* Hiển thị lại toàn bộ các chỉ số mà người dùng đã nhập.
  + - 1. *Minh họa mô hình học máy*
* Ứng dụng hiển thị hình ảnh minh họa cho từng mô hình:
* Nếu chọn **Decision Tree** → hiển thị **sơ đồ cây quyết định** (decision\_tree\_white\_fixed.png).
* Nếu chọn **Logistic Regression** → hiển thị **biểu đồ trọng số đặc trưng** (LR\_feature\_importance.png).
* Người dùng có thể nhấn nút **“MỞ ẢNH LỚN”** để xem hình minh họa chi tiết đối với cả 2 mô hình.

#### ***3.4.3. Kết quả demo***

* Khi nhập các chỉ số rượu thực tế, hệ thống đưa ra kết quả dự đoán như sau:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

* Giao diện giúp nhóm chứng minh khả năng ứng dụng thực tế của mô hình học máy vào việc đánh giá chất lượng sản phẩm.
* Người dùng có thể dễ dàng thử nghiệm, nhập dữ liệu mới và so sánh kết quả giữa hai mô hình.

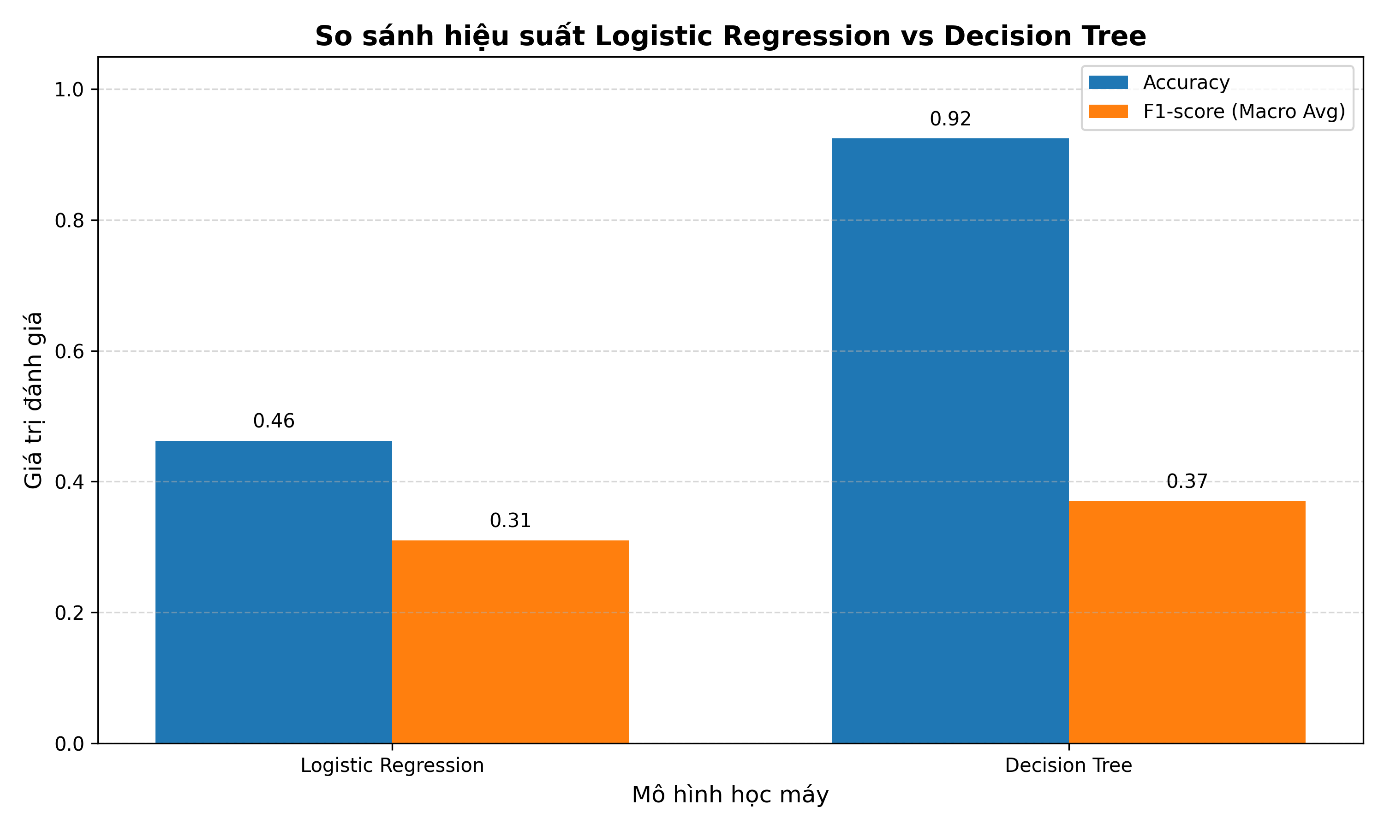
### **3.5 Phân tích kết quả chương trình**

* So sánh hiệu suất giữa **Decision Tree** và **Logistic Regression**:
* Sau khi huấn luyện hai mô hình trên cùng tập dữ liệu **rượu vang trắng (Wine Quality - White)**, nhóm thu được kết quả như sau:

*Bảng 3.1 So sánh hiệu suất giữa Decision Tree và Logistic Regresion*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mô hình | Accuracy | F1-macro | Ưu điểm | Hạn chế |
| Decision Tree (ID3 - Entropy) | 0.9245 | 0.37 | Hiệu suất dự đoán cao, mô hình dễ hiểu, trực quan, biểu diễn được mối quan hệ phi tuyến giữa các đặc trưng | Có nguy cơ overfitting nếu không giới hạn độ sâu cây; nhạy với thay đổi nhỏ trong dữ liệu |
| Logistic Regression (Multinomial) | 0.4612 | 0.31 | Đơn giản, huấn luyện nhanh, phù hợp với dữ liệu tuyến tính | Hiệu suất thấp với dữ liệu phi tuyến; khó mô tả mối quan hệ phức tạp giữa các biến |

* Nhận xét chung:
* Decision Tree cho kết quả vượt trội hơn Logistic Regression cả về độ chính xác (Accuracy) lẫn khả năng tổng quát hóa trên dữ liệu kiểm thử.
* Logistic Regression gặp khó khăn do giả định tuyến tính, trong khi dữ liệu rượu vang có nhiều mối quan hệ phi tuyến giữa các đặc trưng (ví dụ giữa độ cồn và độ chua)
* Việc chuẩn hóa dữ liệu vẫn có ích, nhưng trong Decision Tree thì không ảnh hưởng đáng kể đến kết quả - cho tháy mô hình cây xử lý tốt các đặc trưng có thang đo khác nhau.
* Biểu đồ so sánh:



# PHẦN 3. KẾT LUẬN

* **Tổng hợp kết quả** 
  + Sau khi tiến hành huấn luyện và đánh giá lại hai mô hình Logistic Regression và Decision Tree trên cùng bộ dữ liệu winequality-white.csv, nhóm nhận thấy sự khác biệt đáng kể về hiệu suất giữa hai mô hình
  + Kết quả so sánh hiệu suất:
* Logistic Regression: Accuracy = 0.4612, F1-score = 0.31
* Decision Tree: Accuary = 0.9245, F1-score = 0.37
* Cả Hai mô hình đều được huấn luyện trên cùng một tập dữ liệu, cùng cách chia train/test, cùng phương pháp chuẩn hóa (scaling) và sử dụng các thước đo đánh giá giống nhau để đảm bảo tính công bằng trong quá trình so sánh.
* **Phân tích và đánh giá**
* Mặc dù Logistic Regression đạt độ chính xác thấp hơn đáng kể, điều này phù hợp về mặt bản chất của mô hình, bởi:
* Logistic Regression là mô hình **tuyến tính**, thích hợp với dữ liệu có quan hệ tuyến tính giữa các đặc trưng.
* Dữ liệu rượu vang có **mối quan hệ phi tuyến và phức tạp**, dẫn đến việc Logistic Regression không thể biểu diễn đầy đủ cấu trúc dữ liệu.
* Ngược lại, Decision Tree có khả năng chia không gian dữ liệu thành nhiều vùng nhỏ bằng các **điều kiện nhánh (if-else)**, giúp mô hình hóa tốt hơn các mối quan hệ phi tuyến. Do đó, mô hình này đạt **độ chính xác cao hơn (≈93%)** và thể hiện khả năng phân loại tốt hơn rõ rệt.
* Tuy nhiên, hiệu suất cao của Decision Tree cũng **tiềm ẩn nguy cơ overfitting**, vì mô hình có thể học quá sát dữ liệu huấn luyện.  
   Để kiểm soát hiện tượng này, cần:
* Giới hạn **độ sâu cây (max\_depth)**,
* Áp dụng **pruning (cắt tỉa cây)**,
* Hoặc kết hợp **cross-validation** để đánh giá ổn định mô hình.\
* **Đề xuất hướng cải thiện**
* Dựa trên kết quả thực nghiệm, nhóm đưa ra một số hướng phát triển trong tương lai:
* Sử dụng **Random Forest** để kết hợp nhiều cây quyết định, giảm overfitting và tăng độ chính xác.
* Áp dụng **SVM (Support Vector Machine)** nhằm so sánh thêm với mô hình tuyến tính mạnh hơn Logistic Regression.
* Dùng kỹ thuật **SMOTE** để cân bằng lại dữ liệu giữa các lớp chất lượng thấp – trung bình – cao, giúp cải thiện **F1-score**.
* Kết hợp thêm **Grid Search mở rộng** để tối ưu tham số chi tiết hơn, đặc biệt cho Decision Tree.
* **Kết luận tổng quan**
* Dự án “Dự đoán chất lượng rượu vang” đã được nhóm 14 thực hiện thành công, áp dụng các thuật toán học máy có giám sát (**Decision Tree (ID3)** và **Logistic Regression**) trên bộ dữ liệu **Wine Quality (White Wine)** với 4898 mẫu và 11 đặc trưng hóa học. Kết quả thực nghiệm cho thấy **Decision Tree (ID3)** vượt trội hơn so với **Logistic Regression**, nhờ khả năng học mối quan hệ phi tuyến giữa các đặc trưng và trực quan hóa qua cây quyết định, trong khi **Logistic Regression** phù hợp hơn với dữ liệu tuyến tính nhưng hiệu quả thấp hơn trong trường hợp này.
* Dựa trên kết quả, **Decision Tree (ID3)** được đánh giá là phương pháp tối ưu hơn cho bài toán này nhờ khả năng giải thích trực quan và hiệu suất dự đoán tốt, trong khi **Logistic Regression** phù hợp hơn với dữ liệu tuyến tính nhưng không tối ưu trong bối cảnh này. Tuy nhiên, độ chính xác của dự án còn hạn chế do dữ liệu thiếu đa dạng và thời gian thực hiện gấp rút. Để cải thiện, nhóm đề xuất thử nghiệm **Random Forest** để giảm nguy cơ overfitting của Decision Tree hoặc áp dụng **SMOTE** để xử lý mất cân bằng lớp. Dự án đã củng cố kiến thức về học máy, rèn luyện kỹ năng lập trình Python, xử lý dữ liệu, và làm việc nhóm, tạo nền tảng quý giá cho các nghiên cứu thực tiễn trong tương lai.

# TÀI LIỆU THAM KHẢO

1. Slide bài giảng của thầy Tạ Quang chiểu

2. [Blog: Machine learning cơ bản](https://machinelearningcoban.com/)

3. [Kaggal](https://www.kaggle.com/)