**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**



A blue and white logo

AI-generated content may be incorrect.

**BÀI TẬP LỚN**

**KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI: PHÂN LOẠI CHẤT LƯỢNG NGUỒN NƯỚC DỰA TRÊN THUẬT TOÁN ID3 VÀ C4.5**

**Giáo viên hướng dẫn: TS. Nguyễn Tu Trung**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 | 2251061923 | Bùi Hữu Việt | 64CNTT2 |
| 2 | 2251061846 | Nguyễn Trọng Nam | 64CNTT2 |
| 3 | 2251061803 | Nguyễn Ngọc Huỳnh | 64CNTT2 |
| 4 | 2251061822 | Nguyễn Thị Khánh Linh | 64CNTT2 |

**Hà Nội, năm 2025**

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**



**BÀI TẬP LỚN**

**KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI: PHÂN LOẠI CHẤT LƯỢNG NGUỒN NƯỚC DỰA TRÊN THUẬT TOÁN ID3 VÀ C4.5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 2251061923 | Bùi Hữu Việt | 12/12/2004 |  |  |
| 2 | 2251061846 | Nguyễn Trọng Nam | 31/01/2004 |  |  |
| 3 | 2251061803 | Nguyễn Ngọc Huỳnh | 02/03/2004 |  |  |
| 4 | 2251061822 | Nguyễn Thị Khánh Linh | 28/11/2004 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2025**

# **LỜI NÓI ĐẦU**

Nước là tài nguyên thiết yếu, đóng vai trò vô cùng quan trọng trong mọi mặt của đời sống con người, từ sinh hoạt hàng ngày đến các hoạt động sản xuất công nghiệp và nông nghiệp. Tuy nhiên, cùng với sự phát triển kinh tế - xã hội, vấn đề ô nhiễm nguồn nước do các hoạt động của con người đang ngày càng trở nên nghiêm trọng, đe dọa trực tiếp đến sức khỏe cộng đồng và môi trường sinh thái. Do đó, việc đánh giá chính xác chất lượng nguồn nước, đặc biệt là khả năng sử dụng cho mục đích ăn uống, dựa trên các chỉ số hóa học và vật lý có thể đo đạc được, đã trở thành một nhu cầu cấp thiết.

Xuất phát từ thực tiễn đó, trong khuôn khổ môn học Khai phá dữ liệu, nhóm sinh viên chúng em đã lựa chọn thực hiện đề tài “Phân loại chất lượng nguồn nước dựa trên thuật toán ID3 và C4.5”. Mục tiêu chính của đề tài là ứng dụng hai thuật toán cây quyết định phổ biến là ID3 (triển khai bằng Java) và C4.5 (sử dụng công cụ Weka) để xây dựng mô hình dự đoán khả năng uống được của nước dựa trên 9 đặc trưng đầu vào. Đồng thời, đề tài cũng thực hiện so sánh hiệu quả của hai thuật toán này trên cùng một tập dữ liệu, từ đó rút ra những quy tắc phân loại hữu ích và xác định các yếu tố ảnh hưởng chính đến chất lượng nước.

Để hoàn thành bài tập lớn này, nhóm sinh viên chúng em xin cảm ơn đến thầy TS. Nguyễn Tu Trung, giảng viên hướng dẫn môn học. Thầy đã tận tình chỉ bảo, định hướng và cung cấp những kiến thức chuyên môn giúp chúng em từng bước tiếp cận và giải quyết các vấn đề trong quá trình thực hiện đề tài.

Trong quá trình thực hiện, nhóm chúng em đã nỗ lực vận dụng kiến thức đã học, tuân thủ quy trình khai phá dữ liệu CRISP-DM, từ việc tìm hiểu bài toán, khám phá và tiền xử lý dữ liệu (bao gồm xử lý giá trị thiếu, rời rạc hóa, cân bằng dữ liệu) đến xây dựng, đánh giá mô hình và rút ra kết luận. Tuy nhiên, do hạn chế về thời gian và kinh nghiệm thực tế, bài báo cáo chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ Thầy để báo cáo được hoàn thiện hơn.

# **MỤC LỤC**

[**LỜI NÓI ĐẦU** 3](#_Toc195901145)

[**MỤC LỤC** 4](#_Toc195901146)

[**I. Mô tả bài toán** 5](#_Toc195901147)

[**1. Lý do chọn đề tài** 5](#_Toc195901148)

[**2. Tổng quan bài toán** 5](#_Toc195901149)

[**3. Quy trình thực hiện** 5](#_Toc195901150)

[**4. Phân tích dữ liệu thô** 7](#_Toc195901157)

[**II. Quy trình khai phá dữ liệu** 11](#_Toc195901163)

[**1. Tiền xử lý dữ liệu** 11](#_Toc195901164)

[**2. Biến đổi dữ liệu** 15](#_Toc195901168)

[**3. Phân tách dữ liệu** 16](#_Toc195901169)

[**III. Thuật toán ID3 và thuật toán C4.5** 16](#_Toc195901170)

[**1. Thuật toán ID3** 16](#_Toc195901171)

[**2. Thuật toán C4.5** 23](#_Toc195901175)

[**SO SÁNH KẾT QUẢ CỦA 2 THUẬT TOÁN** 26](#_Toc195901179)

[**TỔNG KẾT** 28](#_Toc195901180)

[**TÀI LIỆU THAM KHẢO** 30](#_Toc195901181)

# 

# **I. Mô tả bài toán**

## **1. Lý do chọn đề tài**

Đối với mỗi con người chúng ta, nước đóng một vai trò vô cùng quan trọng, nước hiện diện tử khắp mọi nơi, từ sinh hoạt hằng ngày đến các hoạt động công nghiệp, tất cả đều cần đến nước. Và khi cuộc sống của con người được cải thiện, chất lượng sinh hoạt được đi lên, các nhà máy công nghiệp cũng được mọc ra như nấm để đáp ứng được nhu cầu cuộc sống hằng ngày của con người. Do đó, sự gia tăng không ngừng về mặt ô nhiễm nguồn nước do công nghiệp, nông nghiệp, hoạt động sinh hoạt của con người hằng ngày. Tất cả các hoạt động trên đều góp phần làm ảnh hưởng đến chất lượng của nguồn nước, do đó, việc đánh giá được khả năng có thể uống được của nguồn nước dựa trên các thông số hóa học và vật lý có thể đo đạc được trở thành một nhu cầu ngày càng cấp thiết và cần được ứng dụng rộng rãi.

## **2. Tổng quan bài toán**

Bài toán thuộc loại phân loại nhị phân, mới đích đến là dự đoán khả năng uống được của nước dựa trên 9 đặc trưng về mặt hóa học và vật lý

* Các biến đầu vào: ph, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic\_carbon, Trihalomethanes, Turbidity
* Biến mục tiêu: Potability: Có thể uống được hay không?

Các thuật toán sẽ được đem ra để sử dụng: ID3 - Code với ngôn ngữ Java và C4.5 sẽ chạy với phần mềm Weka, mục tiêu cuối cùng là để phân loại nước có thể sử dụng được hay không và cũng nhằm để so sánh sự khác biệt giữa 2 thuật toán trên cho cùng một tập dữ liệu

Mục tiêu:

* Xây dựng mô hình cây quyết định chính xác, dễ diễn giải.
* So sánh hiệu suất của ID3 và C4.5 trên tập dữ liệu.
* Đưa ra các quy tắc phân loại giúp xác định yếu tố nào ảnh hưởng mạnh đến chất lượng nước.

## **3. Quy trình thực hiện**

Quy trình khai phá dữ liệu được xây dựng dựa trên mô hình CRISP-DM (Cross Industry Standard Process for Data Mining), bao gồm các bước sau:

### 3.1. Hiểu biết về nghiệp vụ (Business Understanding)

* **Ứng dụng vào bài toán:**
* Vấn đề: Ô nhiễm nguồn nước ngày càng gia tăng do các hoạt động công nghiệp, nông nghiệp và sinh hoạt, dẫn đến nhu cầu đánh giá chất lượng nước (có thể uống được hay không).
* Mục tiêu: Xây dựng mô hình phân loại nhị phân để dự đoán khả năng uống được của nước dựa trên 9 đặc trưng hóa học và vật lý (pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic\_carbon, Trihalomethanes, Turbidity).

### 3.2. Hiểu biết về dữ liệu (Data Understanding)

* **Mục tiêu**: Thu thập và khám phá dữ liệu để hiểu đặc điểm và chất lượng của nó.
* **Ứng dụng vào bài toán**:
* **Tập dữ liệu**: Dữ liệu gồm 9 đặc trưng đầu vào và 1 biến mục tiêu (Potability: 0 - không uống được, 1 - uống được).
* **Khám phá dữ liệu**:
* Kiểm tra phân phối của các đặc trưng (ví dụ: pH có nằm trong khoảng 0-14 không? Hardness, Solids có giá trị bất thường không?).
* Xác định giá trị thiếu (missing values) trong các cột như Sulfate hoặc Trihalomethanes.
* Phân tích tương quan sơ bộ giữa các đặc trưng và biến mục tiêu (ví dụ: Chloramines hoặc Turbidity có ảnh hưởng rõ rệt đến Potability không?).
  + **Công cụ**: Sử dụng Java hoặc Weka để trực quan hóa dữ liệu

### 3.3. Chuẩn bị dữ liệu (Data Preparation)

* **Mục tiêu**: Làm sạch và chuyển đổi dữ liệu để sẵn sàng cho việc xây dựng mô hình.
* **Ứng dụng vào bài toán**:
* **Xử lý giá trị thiếu**:
* Có các giá trị thiếu trong thuộc tính PH, Sulfate, Trihalomethanes:
  + Thay thế bằng giá trị trung bình (mean) hoặc trung vị (median) nếu dữ liệu không lệch nhiều.
  + Loại bỏ mẫu (row) nếu số lượng giá trị thiếu không đáng kể.
  + Sử dụng kỹ thuật nội suy hoặc mô hình dự đoán giá trị thiếu (nếu phức tạp hơn).
* **Chuẩn hóa dữ liệu**:
* Các đặc trưng như Solids, Conductivity có thang đo khác nhau, nên cần chuẩn hóa (normalization/standardization) nếu thuật toán nhạy cảm với thang đo.
* Đối với ID3 và C4.5, chuẩn hóa không bắt buộc vì cây quyết định không phụ thuộc vào thang đo, nhưng vẫn cần đảm bảo dữ liệu đồng nhất.
* **Rời rạc hóa dữ liệu**:
* ID3 và C4.5 hoạt động tốt với dữ liệu rời rạc. Nếu các đặc trưng như pH, Hardness là liên tục, cần chia thành các khoảng (binning) dựa trên ngưỡng hợp lý (ví dụ: pH < 6.5, 6.5-8.5, > 8.5).
* **Chia dữ liệu**:
* Chia tập dữ liệu thành tập huấn luyện (training set, ví dụ 70%) và tập kiểm tra (test set, 30%) để đánh giá hiệu suất mô hình.
* **Định dạng dữ liệu**:
* Chuyển dữ liệu sang định dạng phù hợp cho Java (ID3) và Weka (C4.5).

### 3.4. Xây dựng mô hình (Modeling)

* **Mục tiêu**: Chọn và áp dụng các thuật toán để xây dựng mô hình.
* **Ứng dụng vào bài toán**:
* **Thuật toán ID3**: Triển khai bằng Java
* **Thuật toán C4.5**: Chạy trên Weka

### 3.5. Đánh giá mô hình (Evaluation)

* **Mục tiêu**: Đánh giá hiệu suất mô hình và kiểm tra xem nó có đáp ứng mục tiêu nghiệp vụ không.
* **Ứng dụng vào bài toán**:
* **Độ đo hiệu suất**:
* **Accuracy**: Tỷ lệ dự đoán đúng trên tập kiểm tra
* **Precision, Recall, F1-Score**: Đặc biệt quan trọng nếu tập dữ liệu không cân bằng (ví dụ: số mẫu “uống được” ít hơn “không uống được”).
* **Confusion Matrix**: Kiểm tra số lượng True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN).
* **AUC-ROC**: Đánh giá khả năng phân biệt giữa hai lớp.

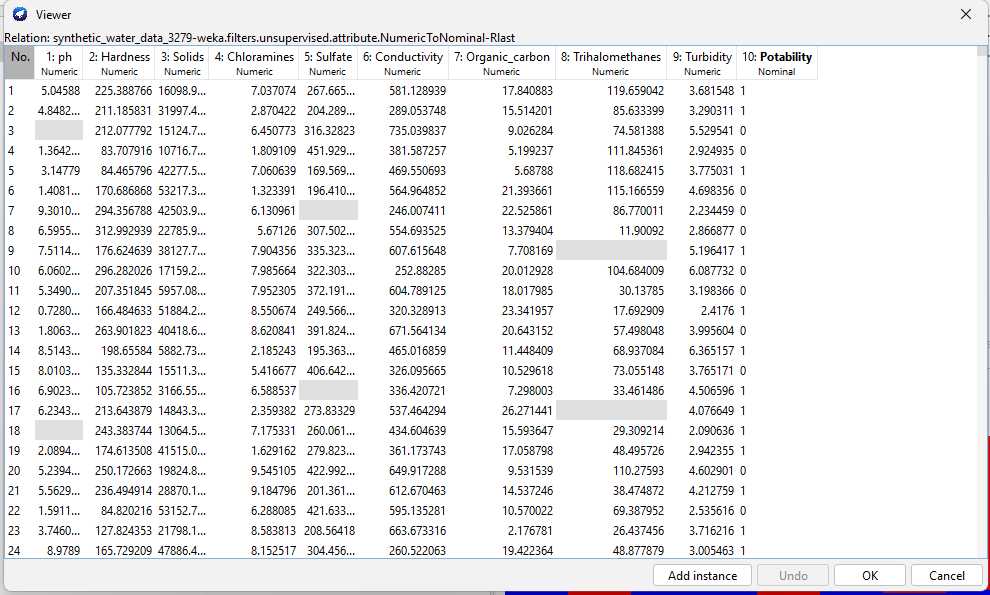
### 3.6. Triển khai (Deployment)

* **Mục tiêu**: Áp dụng mô hình vào thực tế và duy trì nó.
* **Ứng dụng vào bài toán**:
* **Triển khai mô hình**:
* Tích hợp code ID3 (Java) vào một ứng dụng để dự đoán chất lượng nước theo thời gian thực.
* Xuất mô hình C4.5 từ Weka thành file (hoặc code lại) để sử dụng trong hệ thống.
* **Báo cáo kết quả**:
* Tạo báo cáo chi tiết về hiệu suất mô hình, các quy tắc phân loại, và yếu tố ảnh hưởng mạnh nhất đến chất lượng nước.
* Đề xuất cải tiến: Thu thập thêm dữ liệu, thử nghiệm các thuật toán khác (Random Forest, XGBoost).
* **Duy trì**:
* Cập nhật mô hình khi có dữ liệu mới (ví dụ: chất lượng nước thay đổi theo mùa).
* Theo dõi hiệu suất mô hình trong thực tế để đảm bảo độ chính xác.

## **4. Phân tích dữ liệu thô**

### 4.1. Mô tả tổng quan tập dữ liệu

* **Nguồn dữ liệu**: [Water Potability](https://www.kaggle.com/code/nimapourmoradi/water-potability)



* **Kích thước tập dữ liệu**:
* Số lượng mẫu (rows): 3279 bản ghi
* Số lượng đặc trưng (columns): 9 đặc trưng + 1 biến mục tiêu (Potability).
* **Định dạng dữ liệu**: File CSV, ARFF (dùng cho Weka)
* **Biến mục tiêu**:
* Potability: Nhị phân (0 - không uống được, 1 - uống được)
* Kiểm tra phân phối của Potability để xem dữ liệu có cân bằng không
* **Đặc trưng đầu vào**:
* **pH**: Độ pH của nước
* **Hardness**: Độ cứng
* **Solids**: Tổng chất rắn hòa tan
* **Chloramines**: Hàm lượng chloramine
* **Sulfate**: Hàm lượng sulfate
* **Conductivity**: Độ dẫn điện
* **Organic\_carbon**: Hàm lượng carbon hữu cơ
* **Trihalomethanes**: Hàm lượng trihalomethanes
* **Turbidity**: Độ đục

### 4.2. Kiểm tra chất lượng dữ liệu

Phân tích chất lượng dữ liệu để phát hiện các vấn đề như giá trị thiếu, giá trị bất thường, hoặc dữ liệu không nhất quán.

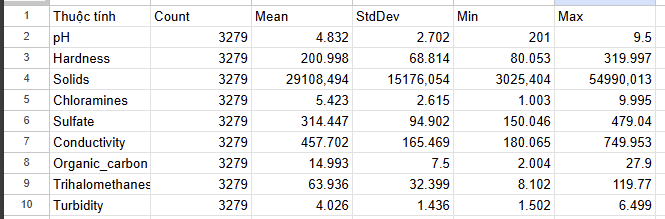
* **Giá trị thiếu (Missing Values)**:
* Kiểm tra từng cột xem có giá trị null, NaN, hoặc trống không.
* **Trong file dữ liệu hiện có**:
* Các cột PH, Sulfate và Trihalomethanes có giá trị bị thiếu
* **Giá trị bất thường (Outliers)**:
* Kiểm tra xem các đặc trưng có giá trị nằm ngoài phạm vi hợp lý không.
* **Trong file dữ liệu hiện có**:
* Không có các giá trị bất thường, các giá trị đều nằm trong khoảng được xác định, không tồn tại giá trị ngoại lai



* **Dữ liệu không nhất quán**:
* Kiểm tra xem có giá trị không hợp lý hoặc mâu thuẫn không (ví dụ: Conductivity âm).
* **Trong file dữ liệu hiện có**:
* Không có các giá trị bất thường, các giá trị đều nằm trong khoảng được xác định

### 4.3. Thống kê mô tả (Descriptive Statistics)

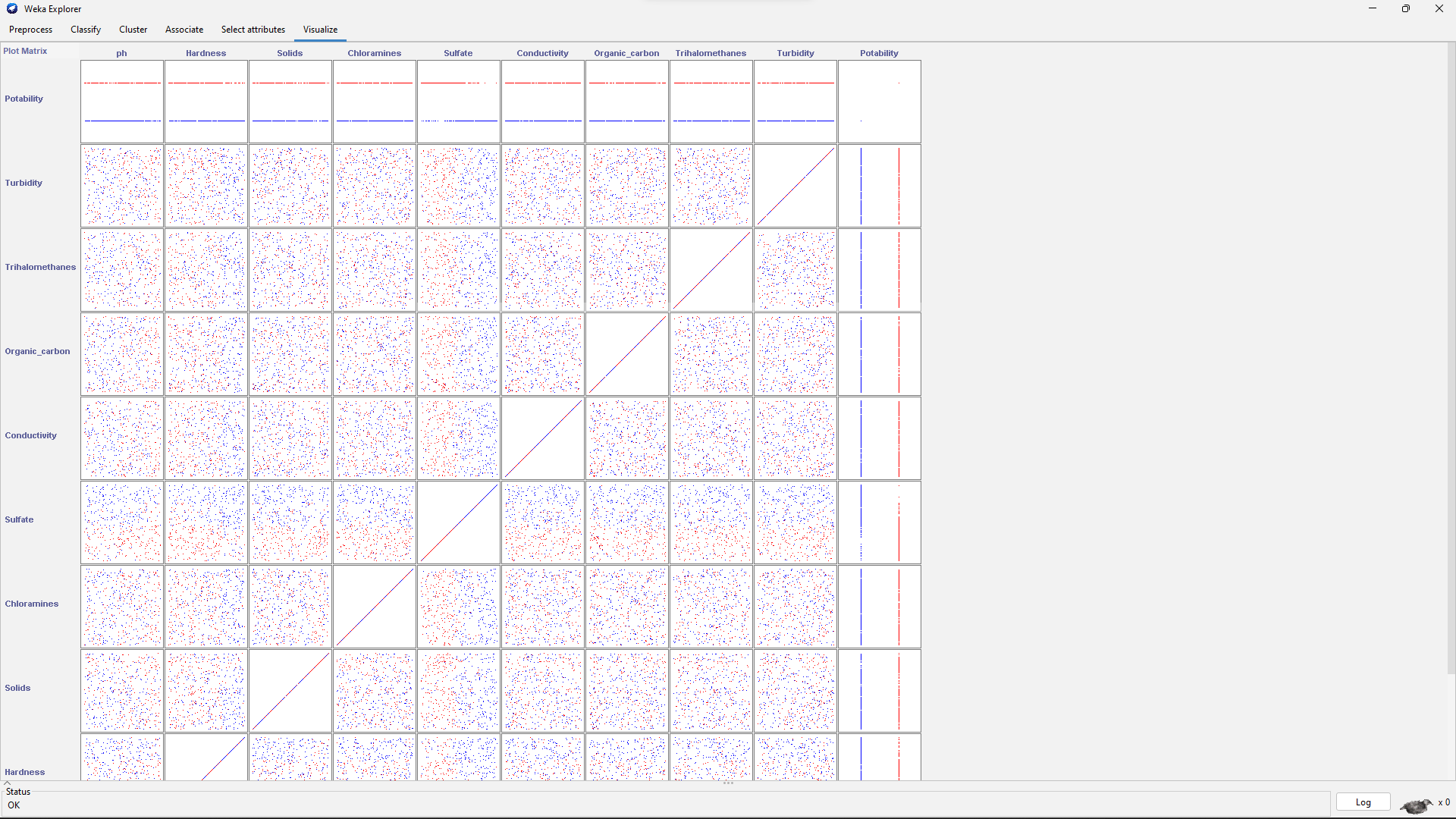
Tính toán các chỉ số thống kê để hiểu phân phối và đặc điểm của từng đặc trưng.

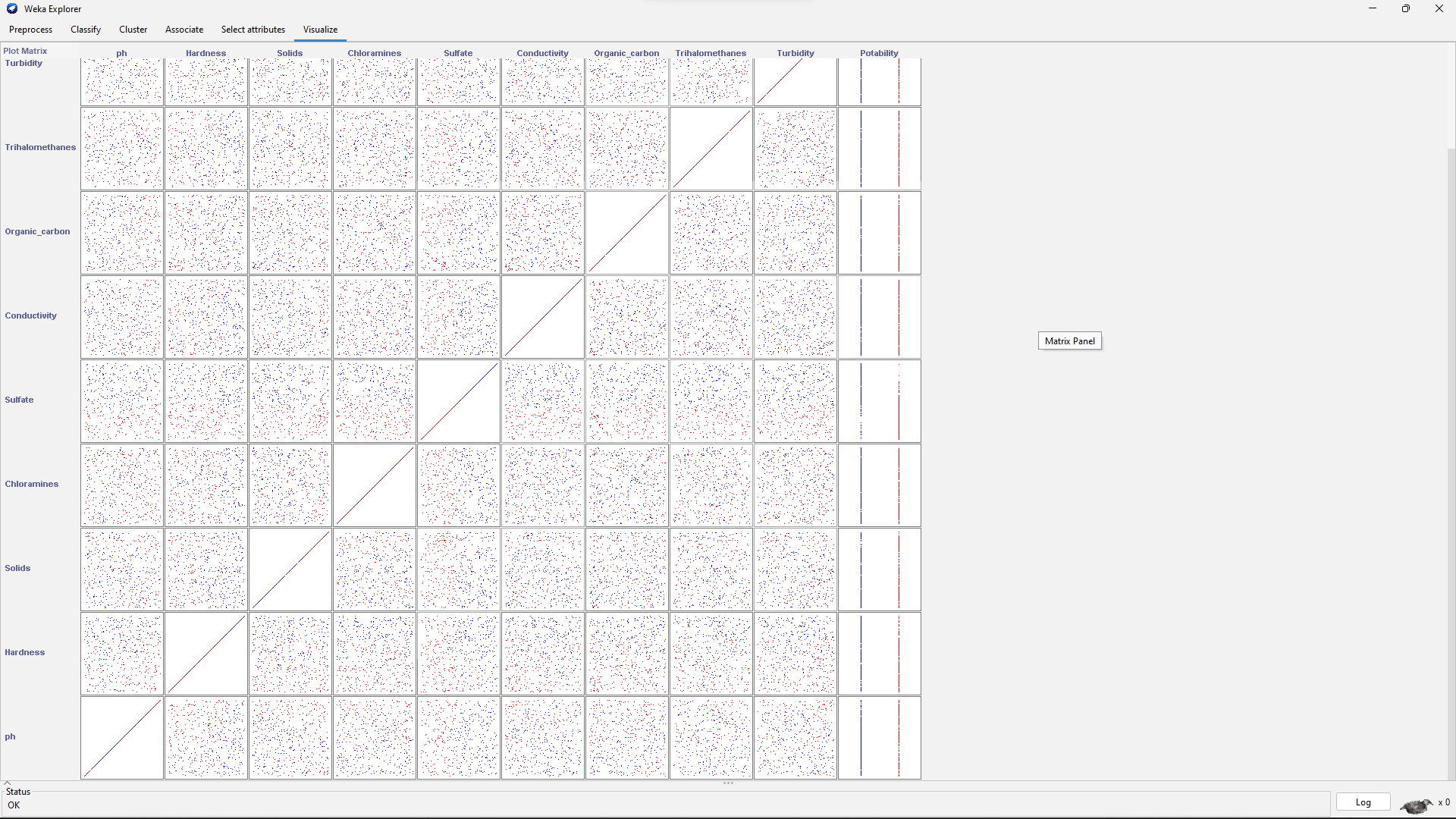


* **Đặc trưng liên tục** (pH, Hardness, Solids, v.v.):
* **Mean**: Giá trị trung bình.
* **Standard Deviation**: Độ lệch chuẩn, thể hiện mức độ phân tán.
* **Min, Max**: Giá trị nhỏ nhất, lớn nhất.
* **Biến mục tiêu (Potability)**:
* Đếm số lượng mẫu thuộc mỗi lớp (0 và 1).
* Tính tỷ lệ % của từng lớp.
* **Potability**:
* 0 (không uống được)
* 1 (uống được)

### 4.4. Phân tích tương quan

* Kiểm tra mối quan hệ giữa các đặc trưng và giữa đặc trưng với biến mục tiêu.
* **Tương quan giữa các đặc trưng**:
* Tính ma trận tương quan (Pearson) để xem các đặc trưng có liên quan mạnh không.
* Như trong ma trận dưới, hầu hết các biểu đồ phân tán cho thấy các điểm dữ liệu phân bố rải rác, không tạo thành một đường thẳng hoặc đường cong rõ ràng. Điều này cho thấy không có mối quan hệ tuyến tính mạnh giữa các biến.





### 4.5. Nhận xét và vấn đề phát hiện

Hành động đề xuất:

* Xử lý giá trị thiếu trong bước tiền xử lý.
* Loại bỏ hoặc sửa các giá trị bất thường.
* Rời rạc hóa dữ liệu cho ID3.
* Sử dụng kỹ thuật cân bằng dữ liệu (như oversampling lớp thiểu số) nếu cần.

# **II. Quy trình khai phá dữ liệu**

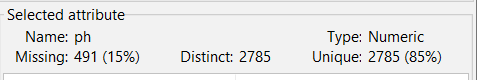
## **1. Tiền xử lý dữ liệu**

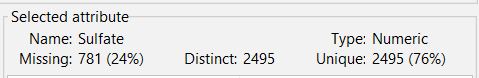
Tiền xử lý dữ liệu là bước quan trọng để đảm bảo chất lượng dữ liệu trước khi đưa vào mô hình ID3 (Java) và C4.5 (Weka). Mục tiêu là làm sạch dữ liệu, loại bỏ các vấn đề như nhiễu, giá trị thiếu, hoặc thuộc tính không cần thiết, đồng thời đảm bảo dữ liệu phù hợp với yêu cầu của các thuật toán

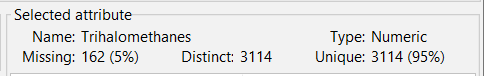
### 1.1. Làm sạch dữ liệu

**1.1.1. Xử lý dữ liệu bị thiếu**

* **Mục tiêu**: Điền hoặc loại bỏ các giá trị thiếu để đảm bảo dữ liệu đầy đủ cho ID3 và C4.5.
* **Phân tích bài toán**:
* Giá trị thiếu xuất hiện trong Sulfate, Trihalomethanes, và pH





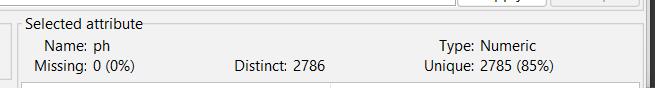


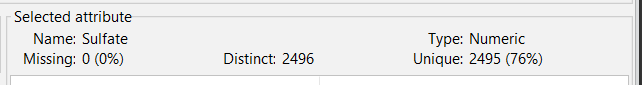
* Xử lý giá trị bị thiếu:
* Bộ lọc Weka: weka.filters.unsupervised.attribute.ReplaceMissingValues

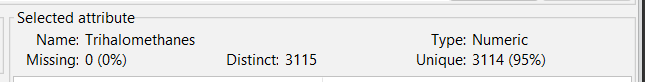
**Các bước:**

1. Trong tab "Preprocess", nhấp vào "Choose" bên cạnh "Filter".
2. Điều hướng đến weka > filters > unsupervised > attribute > ReplaceMissingValues.
3. Nhấp vào "Apply".

* Kết quả sau xử lý
* Các giá trị thiếu trong cột ph, Sulfate, và Trihalomethanes đã được thay thế bằng giá trị trung bình (mean) tương ứng của từng cột. Tập dữ liệu không còn giá trị thiếu.







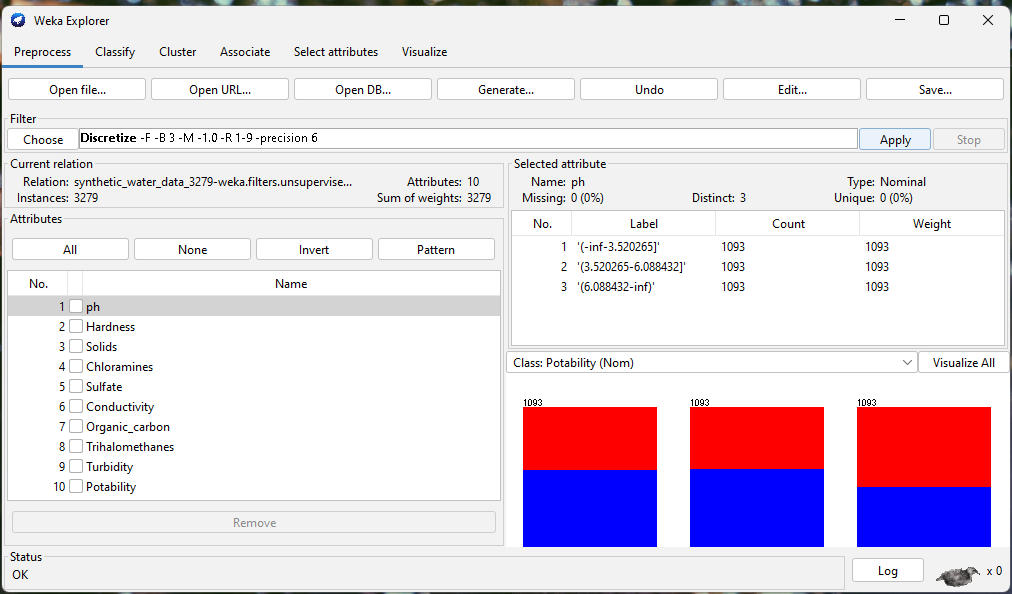
**1.1.2. Rời rạc hóa dữ liệu**

* **Mục tiêu**: Vì dữ liệu nằm trong các khoảng liên tục, do đó ta cần phải tiến hành rời rạc hóa dữ liệu để các thuật toán có thể thuận lợi cho quá trình khai phá
* **Nhận xét:** Bước này là bắt buộc đối với thuật toán ID3 (vì ID3 chỉ xử lý thuộc tính danh nghĩa). Tuy nhiên, thuật toán C4.5 có khả năng xử lý trực tiếp thuộc tính liên tục, do đó trong báo cáo này, khi chạy C4.5 trong Weka, bước rời rạc hóa này sẽ bỏ qua để thuật toán tự tìm điểm chia tốt nhất, tránh làm mất thông tin. File sau khi rời rạc hóa sẽ được xuất ra dành riêng cho thuật toán ID3
* **Xử lý**:

Bộ lọc Weka: weka.filters.unsupervised.attribute.Discretize

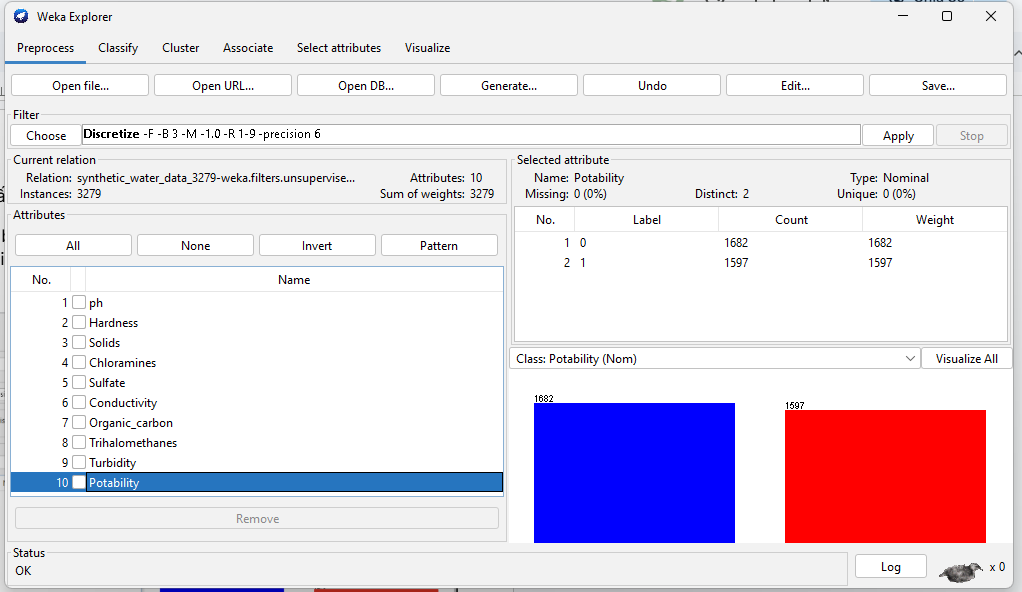
**Các bước:**

1. Trong tab "Preprocess", nhấp vào "Choose" > weka > filters > unsupervised > attribute > Discretize.
2. Chọn Thuộc Tính: chọn các thuộc tính cần rời rạc hóa hoặc để "1-9" để rời rạc hóa tất cả.
3. Số Lượng Khoảng: Thuộc tính bins kiểm soát số lượng danh mục được tạo: đặt là 3 vì các dữ liệu thưởng ở 3 mức: Thấp - Trung Bình - Cao
4. Phương Pháp Binning: Weka cung cấp các phương pháp binning khác nhau. EqualWidth tạo ra các khoảng có độ rộng bằng nhau, trong khi EqualFrequency (hoặc "EqualDepth") cố gắng đưa số lượng instance gần bằng nhau vào mỗi khoảng.
5. Nhấp vào "Apply".



**1.1.3. Xử lý dữ liệu mất cân bằng**

* **Mục tiêu**: Cân bằng tỷ lệ giữa các lớp để tránh tình trạng mô hình bị thiên về lớp chiếm đa số, đặc biệt quan trọng đối với các bài toán phân loại.



Như trong file dữ liệu trên, dữ liệu thiên về “0” tức nước “Không uống được” nhiều hơn một chút, do đó ta cần phải bổ sung hoặc giảm thiểu bản ghi có đầu ra là 0 hoặc 1

* Xử lý mất cân bằng bản ghi

SMOTE: Sử dụng bộ lọc weka.filters.supervised.instance.SMOTE.

**Áp Dụng Bộ Lọc SMOTE**

* Trong tab "Preprocess", nhấp vào "Choose" bên cạnh "Filter".
* Điều hướng đến weka > filters > supervised > instance > SMOTE.
* Nhấp vào bộ lọc SMOTE trong hộp

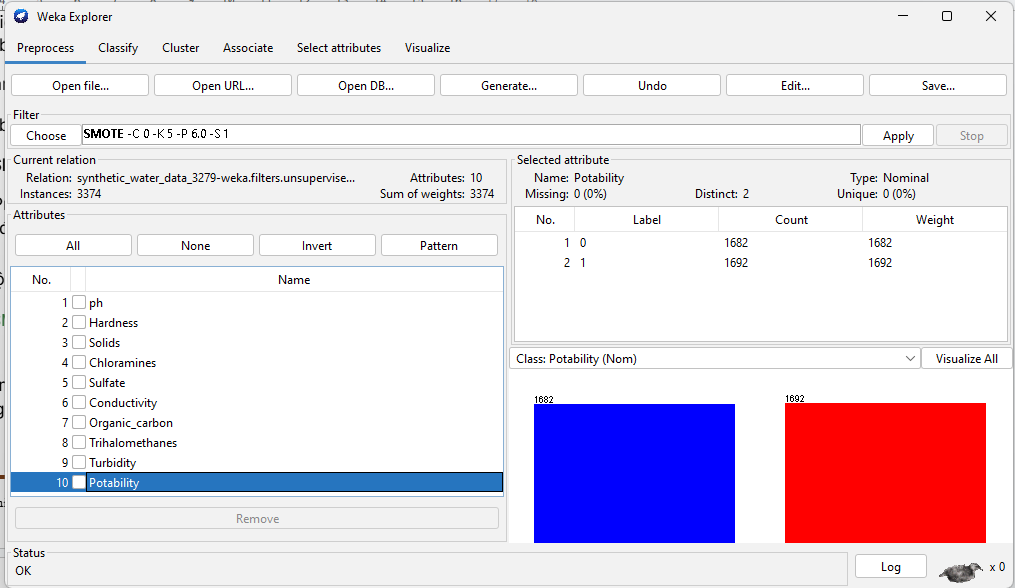
Nhấp vào bộ lọc SMOTE trong hộp để mở cửa sổ thuộc tính. Đây là bước quan trọng.

**percentage:** Tham số này xác định mức độ oversampling: set 6% để có thêm dữ liệu bản ghi cho quá trình khai phá cho đầu ra 1

**k\_NearestNeighbors:** Tham số này xác định số lượng láng giềng gần nhất được sử dụng để tạo ra các mẫu tổng hợp: set ở giá trị mặc định là 5.

Kết quả thu được là dữ liệu đã được cân bằng ở các bản ghi và sinh thêm các bản ghi mới

-> Có tổng cộng 3374 bản ghi mới, trong đó có 1682 bản ghi cho nhãn “0” và 1692 bản ghi cho nhãn “1”



### 1.2. Tích hợp dữ liệu

**Mục tiêu**: Kết hợp dữ liệu từ nhiều nguồn (nếu có) để tạo ra một tập dữ liệu thống nhất, đảm bảo tính đầy đủ và đồng bộ cho quá trình xây dựng mô hình ID3 và C4.5.

**Phân tích bài toán**:

* Tích hợp dữ liệu nhằm:
* Đảm bảo tất cả mẫu có cùng tập hợp đặc trưng (pH, Hardness, Chloramines, Sulfate, Conductivity, Organic\_carbon, Trihalomethanes, Turbidity, Potability).
* Đồng bộ định dạng và đơn vị đo
* Loại bỏ trùng lặp (nếu có).
* Bởi vì tập dữ liệu được lấy ra từ một nguồn duy nhất, do đó không cần quá trình tích hợp dữ liệu

### 1.3. Biến đổi dữ liệu (chuẩn hóa dữ liệu)

Là quá trình biến đổi hay kết hợp dữ liệu vào những dạng thích hợp cho quá trình khai phá dữ liệu, bao gồm:

* Làm trơn dữ liệu
* Kết hợp dữ liệu
* Tổng quát hóa
* Chuẩn hóa
* Xây dựng thuộc tính/đặc tính
* Thu giảm dữ liệu

⇨ Chuyển dữ liệu của các thuộc tính về dạng nominal, bước rời rạc hóa đã thực hiện ở mục 1.1.2 là bước biến đổi quan trọng nhất cho ID3.

## **2. Biến đổi dữ liệu**

**Mục tiêu:** Cân bằng dữ liệu và chuẩn bị dữ liệu phù hợp cho cả ID3 và C4.5.

**Phân tích bài toán:**

* **Cân bằng dữ liệu:**
* Dữ liệu ban đầu chỉ hơi mất cân bằng (51.3% - 48.7%), nhưng ta vẫn đem cân bằng lại dữ liệu
* Sử dụng SMOTE trong Weka để tăng số mẫu của lớp thiểu số (Potability = 1):
* percentage = 6%, classValue = 0.
* Kết quả: Tổng số mẫu tăng lên 3374, với 1682 lớp 0 và 1692 lớp 1 (49.85% - 50.14%).
* **Rời rạc hóa dữ liệu (cho ID3):**
* ID3 yêu cầu dữ liệu dạng danh nghĩa. Các thuộc tính liên tục (pH, Hardness, v.v.) cần được rời rạc hóa
* Lưu dữ liệu rời rạc hóa dưới dạng .csv để sử dụng cho ID3 trong Java.

## **3. Phân tách dữ liệu**

**Mục tiêu:** Chia dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá hiệu suất mô hình.

**Phân tích bài toán:**

* **C4.5 (Weka):**
* Sử dụng 10-fold cross-validation để đánh giá mô hình, đảm bảo kết quả ổn định hơn so với chia tỷ lệ phần trăm (percentage split).
* **ID3 (Java):**
* Chia dữ liệu thành:
* Tập huấn luyện: 80%
* Tập kiểm tra: 20%

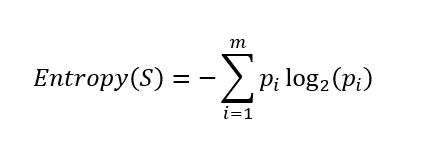
# **III. Thuật toán ID3 và thuật toán C4.5**

## **1. Thuật toán ID3**

### 1.1. Lý thuyết

ID3 (Iterative Dichotomiser 3) là một thuật toán cây quyết định được phát triển bởi Ross Quinlan vào năm 1986, dựa trên lý thuyết thông tin (Information Theory) để xây dựng cây quyết định phân loại. Thuật toán này sử dụng khái niệm **Information Gain** để lựa chọn thuộc tính tốt nhất tại mỗi bước xây dựng cây, nhằm tối ưu hóa việc phân chia dữ liệu thành các lớp mục tiêu.

* **Cơ sở toán học:**
* **Entropy:** Đo lường mức độ hỗn loạn (hay sự không chắc chắn) trong tập dữ liệu (S):

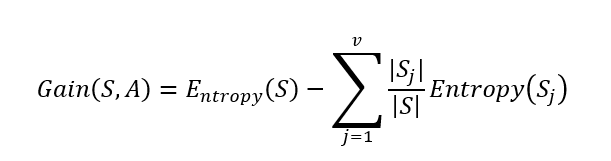
  
  
Trong đó:

S: Tập dữ liệu hiện tại.

m: Số lượng lớp phân loại (trong bài toán này là 2: uống được/không uống được).

pi​: Tỷ lệ (xác suất) các mẫu thuộc lớp thứ i trong tập S.

* **Information Gain:** Đo lường mức độ giảm Entropy khi chia dữ liệu dựa trên một thuộc tính (A):

  
Trong đó:

Entropy(S): Entropy của tập dữ liệu S trước khi chia.

A: Thuộc tính đang được xem xét để chia.

v: Số lượng giá trị khác nhau của thuộc tính A.

Sj​: Tập con của S chứa các mẫu có giá trị thứ j của thuộc tính A.

∣Sj​∣: Số lượng mẫu trong tập con Sj​.

∣S∣: Tổng số lượng mẫu trong tập S.

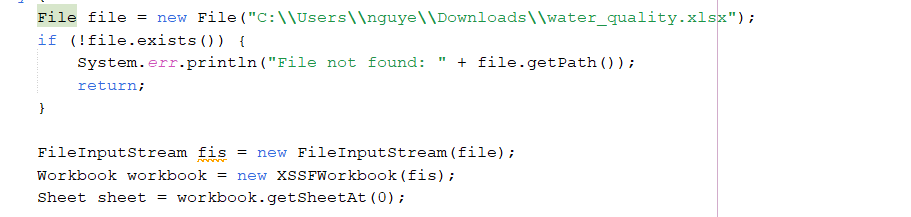
Entropy(Sj​): Entropy của tập con Sj​ sau khi chia theo giá trị thứ j thuộc tính A.

* **Thủ tục giải thuật ID3:**
* B1: Tạo “nút\_gốc” cho cây quyết định
* B2: Kiểm tra trường hợp đặc biệt
  + B2.1: If tất cả mẫu huấn luyện của D đều có trị của C là P, return cây có một nút duy nhất là nút\_gốc với nhãn P
  + B2.2: If tất cả mẫu huấn luyện của D đều có trị của C là N, return cây có một nút duy nhất là nút\_gốc với nhãn N
  + B2.3: If A là rỗng return cây có nút duy nhất là nút\_gốc với nhãn là trị phổ biến nhất của C trong tập mẫu
* B3: Else begin
  + B3.1: Gọi X là thuộc tính của A phân lớp D tốt nhất //Xác định X bằng cách tính Gain information
  + B3.2: Gán nhãn nút\_gốc với tên thuộc tính X
  + B3.3: For each giá trị v của X
    - B3.3.1: Thêm một nhánh cây mới dưới nút\_gốc ứng với X=v
    - B3.3.2: Xác định tập con Dv ứng với X=v
    - B3.3.3: If Dv là rỗng: Thêm dưới nhánh mới này một nút lá có nhãn là trị phổ biến nhất của thuộc tính quyết định trong D
    - B3.3.4: Else: Thêm cây con vào dưới nhánh mới này bằng cách gọi đệ quy ID3(Dv, C, A-{X})
  + End
* Return nút\_gốc
* **Đặc điểm của ID3:**
* **Xử lý dữ liệu danh nghĩa:** ID3 chỉ làm việc với các thuộc tính dạng danh nghĩa (nominal/categorical). Nếu dữ liệu có thuộc tính liên tục (như pH, Hardness), cần rời rạc hóa trước (ví dụ: chia pH thành các khoảng [<6.5, 6.5-8.5, >8.5]).
* **Không xử lý giá trị thiếu:** ID3 không có cơ chế tự động xử lý giá trị thiếu, do đó cần tiền xử lý dữ liệu để điền giá trị thiếu trước khi áp dụng thuật toán.
* **Không hỗ trợ cắt tỉa:** ID3 xây dựng cây đầy đủ, dễ dẫn đến hiện tượng **overfitting** .
* **Thiên về thuộc tính có nhiều giá trị:** Vì Information Gain có xu hướng ưu tiên các thuộc tính có nhiều giá trị (do Entropy giảm mạnh hơn khi chia nhỏ), điều này có thể làm cây quyết định trở nên không tối ưu.
* **Ưu điểm:**
* Dễ triển khai và diễn giải (các quy tắc từ cây quyết định có dạng "Nếu... thì...").
* Phù hợp với các tập dữ liệu nhỏ và dữ liệu danh nghĩa.
* Tốc độ xây dựng cây nhanh.
* **Nhược điểm:**
* Chỉ xử lý được dữ liệu danh nghĩa, yêu cầu rời rạc hóa dữ liệu liên tục.
* Không xử lý được giá trị thiếu trực tiếp.
* Dễ bị overfitting do không có cơ chế cắt tỉa.
* Thiên về thuộc tính có nhiều giá trị, có thể dẫn đến cây không cân bằng.

### 1.2. Quy trình thực hiện

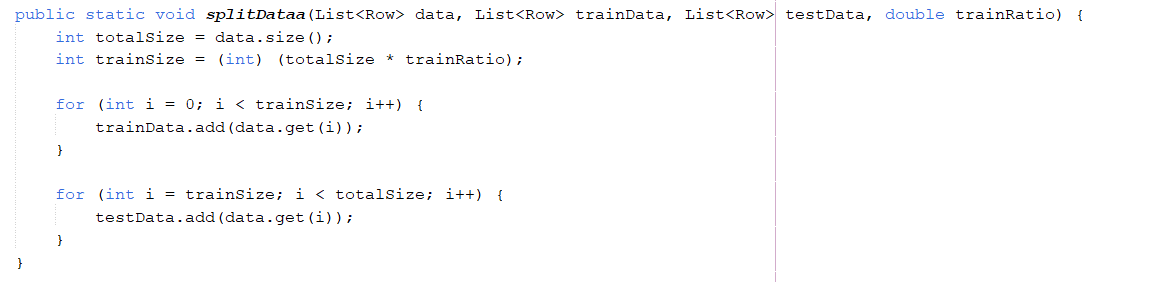
Quy trình triển khai thuật toán ID3 bằng Java để xây dựng cây quyết định từ tập dữ liệu huấn luyện (đã tiền xử lý bằng weka) bao gồm các bước chính sau:

1. **Đọc dữ liệu:** Đọc dữ liệu với bằng java sử dụng thư viện apache-poi.

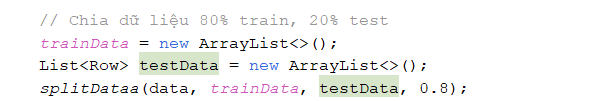


1. **Chia tập dữ liệu** thành trainData(80%) và testData(20%)

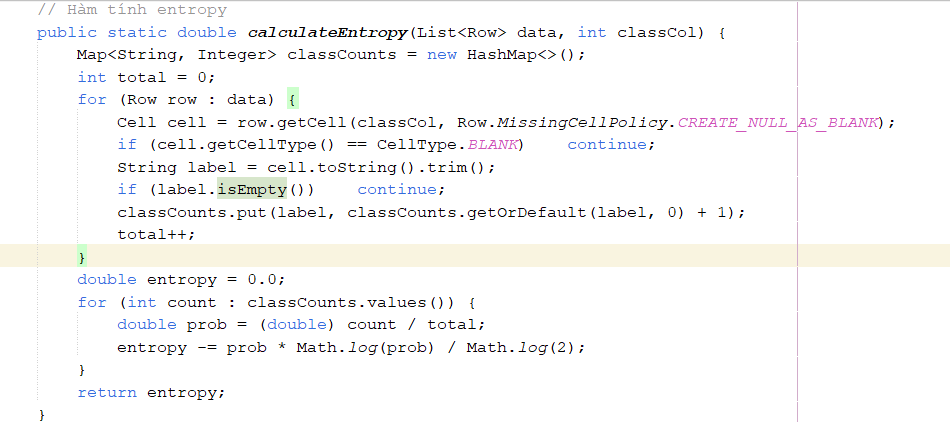
* Hàm chia dữ liệu:



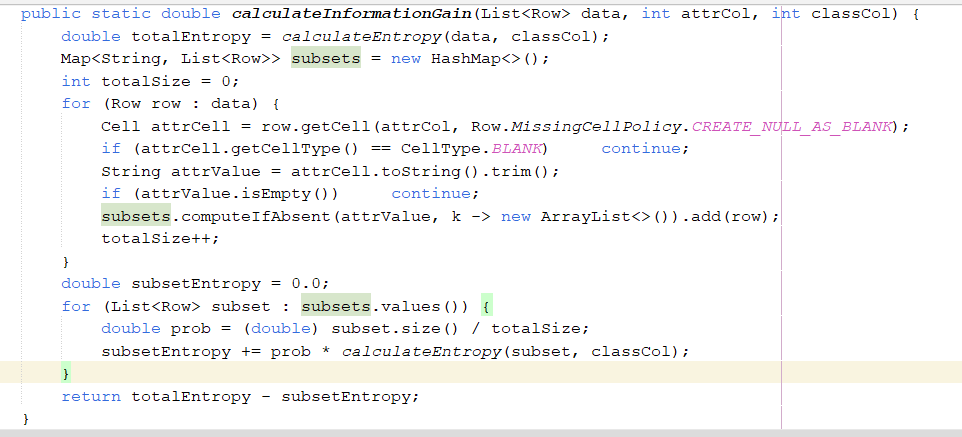
* Chia dữ liệu:



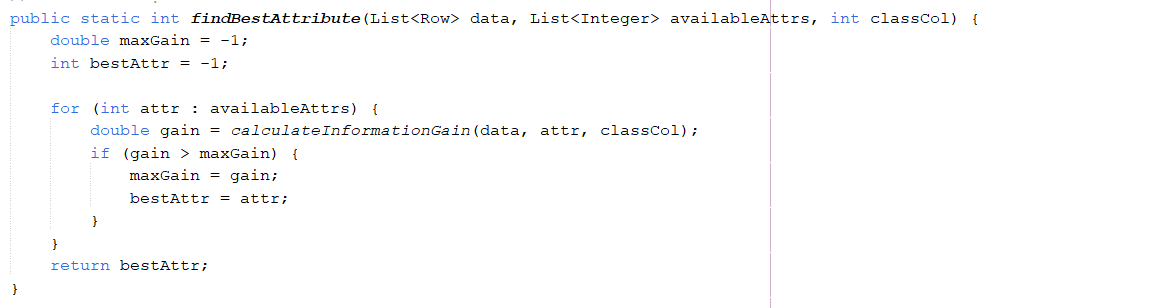
1. **Tính Entropy:** Xây dựng hàm tính Entropy của một tập dữ liệu (hoặc tập con) dựa trên phân phối của các lớp mục tiêu (Potability = 0 và 1).



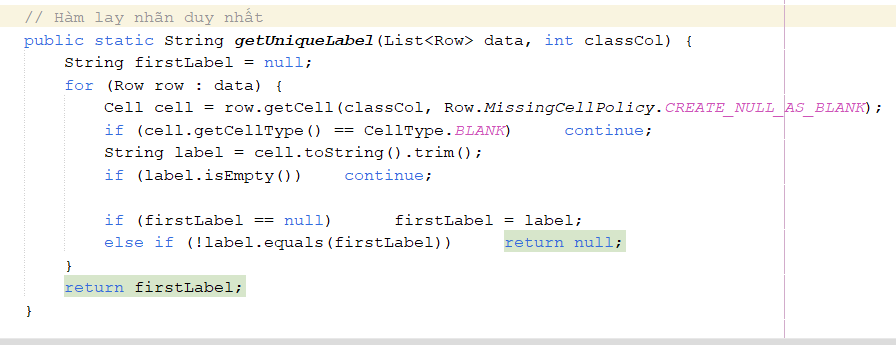
1. **Tính Information Gain:** Xây dựng hàm tính Information Gain cho một thuộc tính cụ thể.



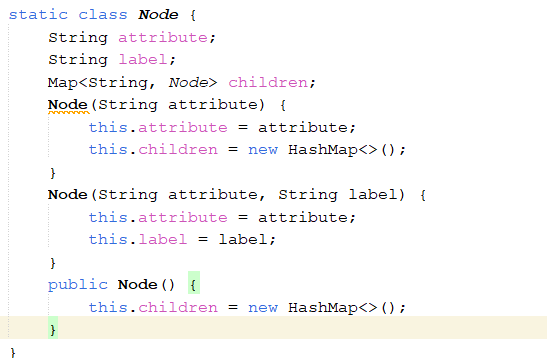
1. **Chọn thuộc tính tốt nhất:** Duyệt qua từng thuộc tính chọn thuộc tính có information gain lớn nhất.



1. **Xây dựng hàm lấy nhãn duy nhất (nếu có)**

****

1. **Định nghĩa lớp Node cho đối tượng là 1 Node ở trong cây**

****

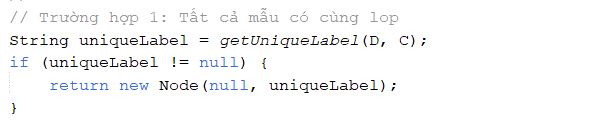
1. **Xây dựng hàm ID3 để xây cây quyết định:**

Bước 1: Tạo node gốc

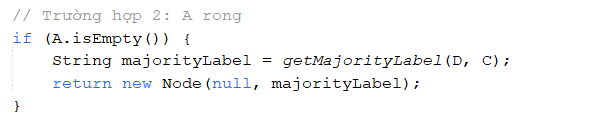


Bước 2: Kiểm tra các trường hợp đặc biệt

* Bước 2.1: Trường hợp 1: tất cả các mẫu huấn luyện cùng nhãn

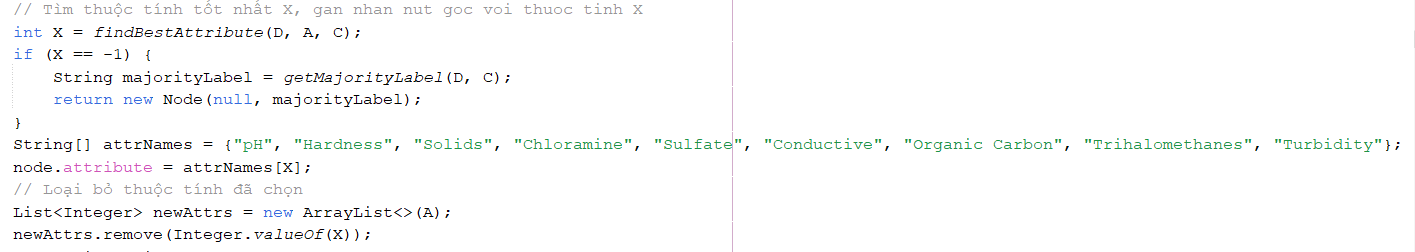


* Bước 2.2: Trường hợp 2: Tập thuộc tính rỗng

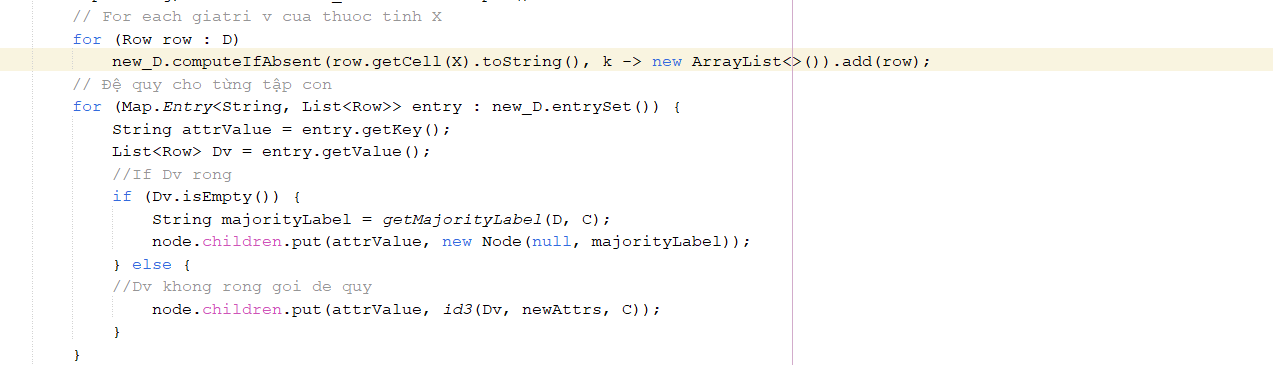


Bước 3:

* Bước 3.1 và 3.2: Xác định thuộc tính X trong tập thuộc tính A phân lớp dữ liệu D tốt nhất: X là thuộc tính có Gain information lớn nhất, gán nhãn nút gốc với tên thuộc tính X và loại X ra khỏi tập thuộc tính



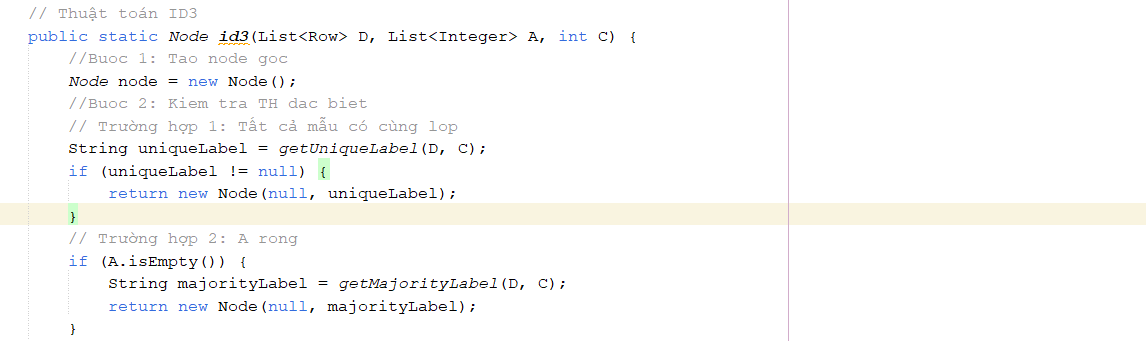
* Bước 3.3: For each với từng giá trị v của X
* B3.3.1: Thêm một nhánh cây mới dưới nút\_gốc ứng với X=v
* B3.3.2: Xác định tập con Dv ứng với X=v
* B3.3.3: If Dv là rỗng: Thêm dưới nhánh mới này một nút lá có nhãn là trị phổ biến nhất của thuộc tính quyết định trong D
* B3.3.4: Else: Thêm cây con vào dưới nhánh mới này bằng cách gọi đệ quy ID3(Dv, C, A-{X})

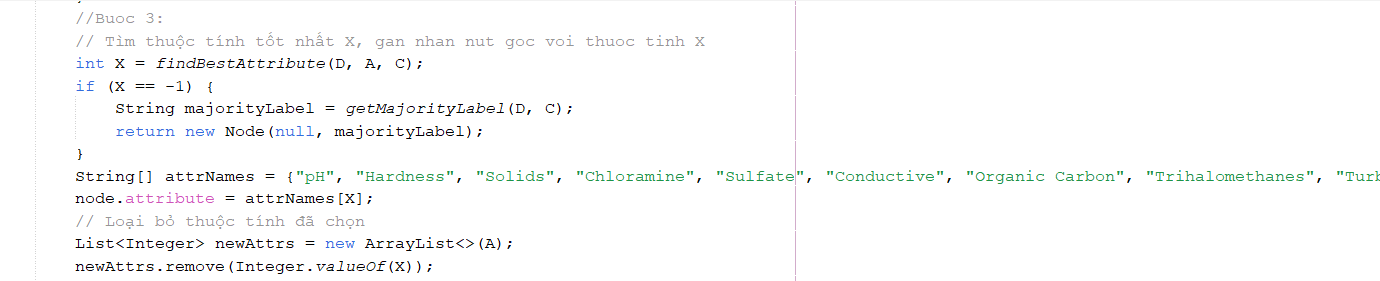


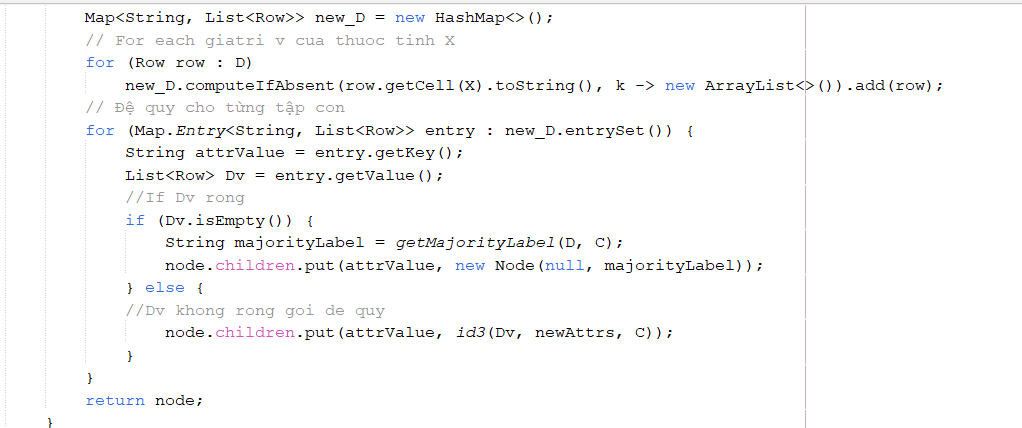
* + End
* Return nút gốc



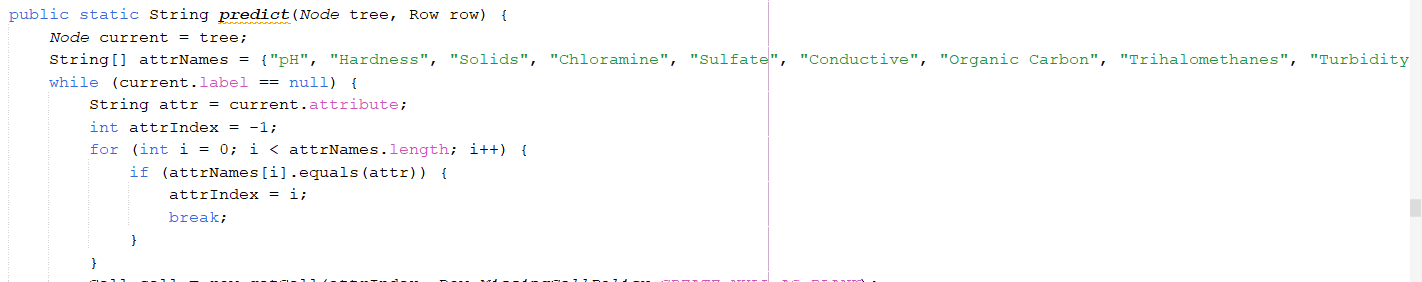
* Code hoàn chỉnh hàm ID3(D,A,C);

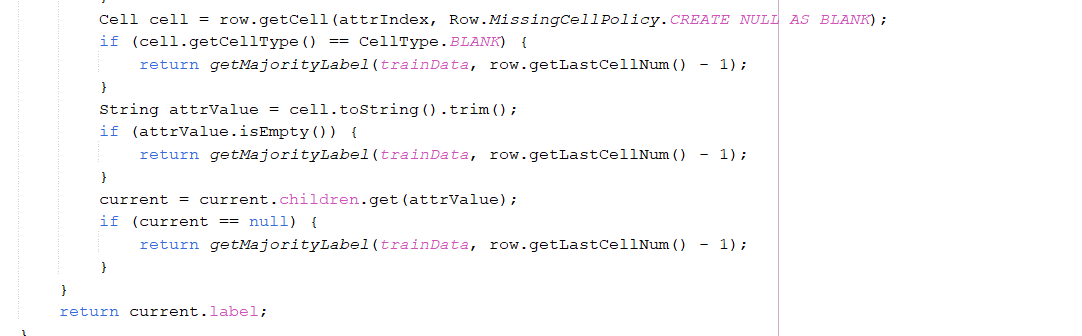






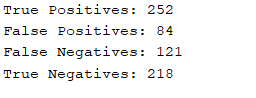
1. **Xây dựng hàm dự đoán predict**

****

****

### 1.3. Kết quả thu được

Đánh giá hiệu năng trên tập kiểm tra:

****

* Accuracy = 0.6963, tỷ lệ dự đoán chính xác khá thấp chỉ 69.63%.
* Precision = 0.75, cho biết trong số tất cả các trường hợp mà mô hình dự đoán là "positive" có 75% là đúng.
* Recall = 0.6756, cho thấy mô hình đã xác định đúng 67.56% trong tổng số các trường hợp "positive" thực tế.
* F1-score = 0.7109, giá trị này cho thấy một sự cân bằng tương đối tốt giữa Precision và Recall.

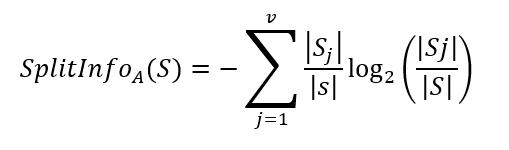
## **2. Thuật toán C4.5**

### 2.1. Lý thuyết

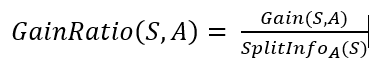
C4.5 là thuật toán xây dựng cây quyết định được phát triển bởi Ross Quinlan như một sự cải tiến của ID3. Nó khắc phục được nhiều nhược điểm của ID3 và trở thành một trong những thuật toán phân loại phổ biến nhất. Các cải tiến chính bao gồm:

1. **Xử lý thuộc tính liên tục (Continuous Attributes):** C4.5 có thể xử lý trực tiếp các thuộc tính số (như pH, Hardness) mà không cần rời rạc hóa trước. Tại mỗi nút, C4.5 xem xét tất cả các ngưỡng chia (split point) khả dĩ cho thuộc tính liên tục (thường là điểm giữa của hai giá trị liên tiếp đã sắp xếp) và chọn ngưỡng cho Gain Ratio tốt nhất để tạo ra một nhánh nhị phân (ví dụ: pH <= 7.2 và pH > 7.2).
2. **Xử lý giá trị thiếu (Missing Values):** Khi gặp giá trị thiếu của một thuộc tính trong quá trình huấn luyện hoặc dự đoán, C4.5 phân phối mẫu đó xuống các nhánh con theo tỷ lệ các mẫu có giá trị đã biết đi xuống các nhánh đó (fractional instances).
3. **Sử dụng Gain Ratio:** Để khắc phục thiên vị của Information Gain đối với các thuộc tính có nhiều giá trị, C4.5 sử dụng **Gain Ratio**. Nó điều chỉnh Information Gain bằng cách chia cho **Split Information** (còn gọi là Potential Information), là giá trị đo lường mức độ phân mảnh dữ liệu của thuộc tính đó.

* **Split Information:** Đo lường thông tin cần thiết để xác định giá trị của thuộc tính A cho một mẫu trong tập S.



* Trong đó: v là số lượng giá trị của thuộc tính A, và Sj​ là tập con các mẫu có giá trị thứ j của thuộc tính A.
* **Gain Ratio:** Là tỷ lệ giữa Information Gain và Split Information

​

* C4.5 chọn thuộc tính có Gain Ratio cao nhất. Tuy nhiên, để tránh trường hợp SplitInfo quá nhỏ (khi thuộc tính chia dữ liệu thành nhiều tập con rất nhỏ), C4.5 thường chỉ xem xét các thuộc tính có Information Gain cao hơn mức trung bình trước khi tính Gain Ratio.

1. **Cắt tỉa cây (Pruning):** Đây là cải tiến quan trọng nhất để chống lại overfitting. Sau khi xây dựng cây quyết định đầy đủ, C4.5 thực hiện **cắt tỉa sau (post-pruning)**. Nó duyệt từ dưới lên (từ lá về gốc), xem xét việc thay thế một cây con (subtree) bằng một nút lá duy nhất hoặc thay thế một nút trong bằng nhánh con phổ biến nhất của nó

### 2.2. Quy trình thực hiện

**Chuẩn bị dữ liệu:**

* Mở Weka Explorer.
* Trong tab Preprocess, nhấn "Open file..." và tải file dữ liệu đã được tiền xử lý
* Kiểm tra lại các thuộc tính và đảm bảo thuộc tính Potability được chọn là lớp (chọn thuộc tính và nhấn nút "Attributes -> Use as class").

**Chọn thuật toán và cấu hình:**

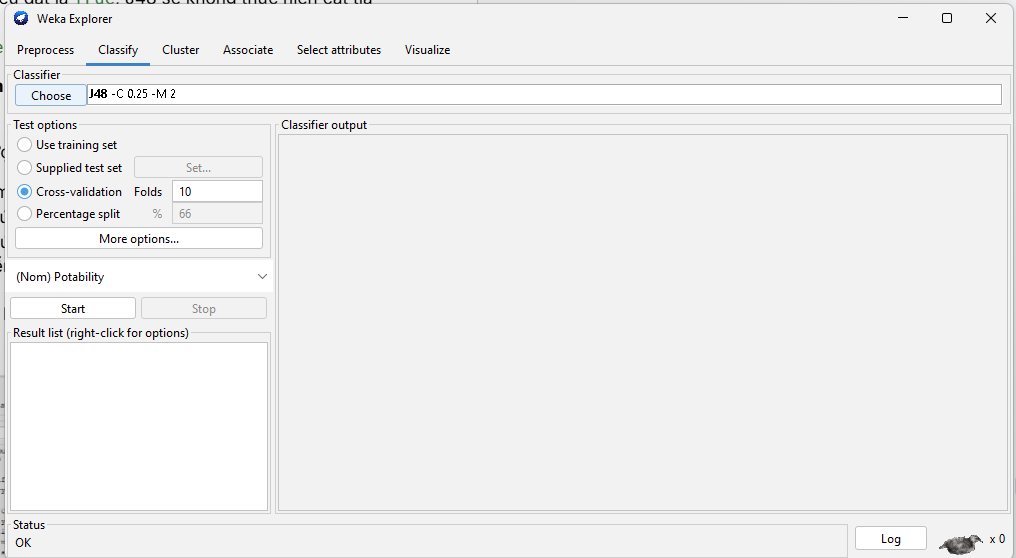
* Chuyển sang tab Classify.
* Nhấn nút "Choose" -> trees -> J48. Thuật toán J48 sẽ được chọn.
* Nhấp vào tên thuật toán "J48" để xem và thay đổi các tham số. Các tham số quan trọng bao gồm:
* confidenceFactor: Tham số cho cơ chế cắt tỉa (giá trị nhỏ hơn -> cắt tỉa nhiều hơn, mặc định 0.25).
* minNumObj: Số lượng mẫu tối thiểu tại một nút lá (mặc định 2).
* unpruned: Nếu đặt là True, J48 sẽ không thực hiện cắt tỉa (tương tự ID3 hơn nhưng với các cải tiến khác của C4.5). Mặc định là False (thực hiện cắt tỉa).

**Chọn phương pháp đánh giá:**

* Trong khung Test options, chọn Cross-validation.
* Đảm bảo Folds được đặt là 10.

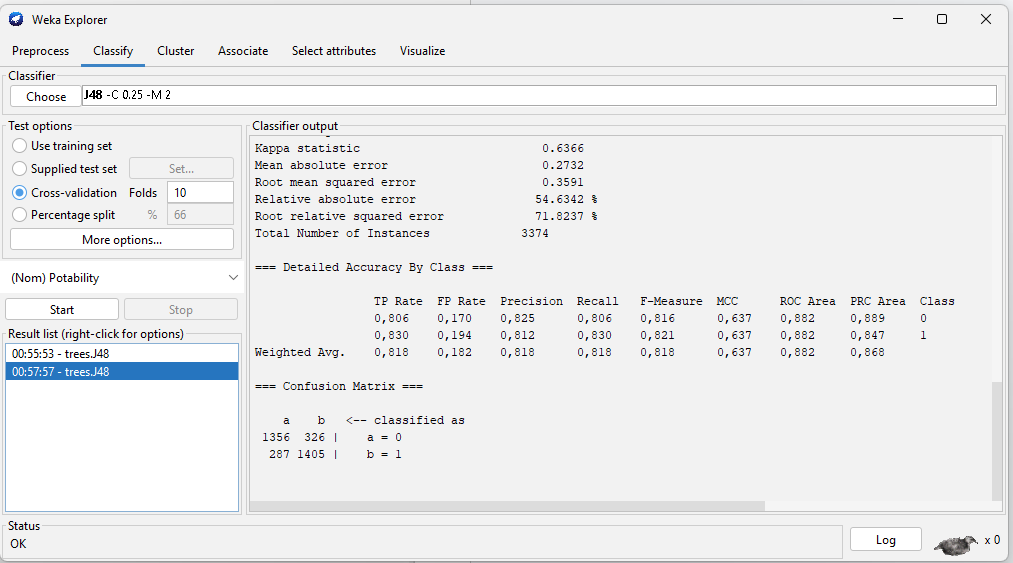
**Chọn thuộc tính lớp:** Đảm bảo danh sách thả xuống ở trên cùng (bên cạnh "Test options") hiển thị đúng thuộc tính lớp là Potability.

**Chạy thuật toán:** Nhấn nút Start. Weka sẽ thực hiện huấn luyện và đánh giá mô hình J48 bằng kiểm định chéo 10 lần.

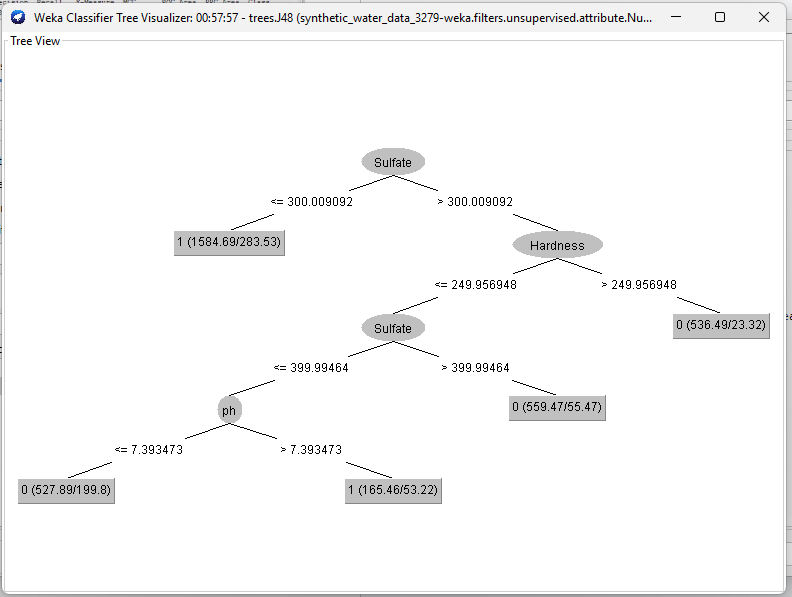


**Xem kết quả:** Kết quả sẽ hiển thị trong khung Classifier output bên phải.

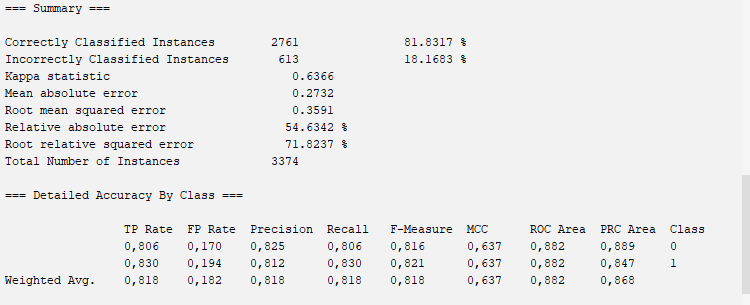
### 2.3. Kết quả thu được

****

**Cấu trúc cây quyết định:**

****

**Đánh giá hiệu năng trên tập kiểm tra:**

****

* **Về độ chính xác:** Đạt 81.8317%, tương đối cao
* **Kappa statistic:** 0.6366 - Đây là một giá trị Kappa cao, cho thấy mức độ đồng thuận đáng kể (substantial agreement), vượt xa sự trùng hợp ngẫu nhiên.

**Hiệu suất theo lớp (Detailed Accuracy By Class):**

* **Rất cân bằng và tốt:** Cả hai lớp đều có Recall và Precision trên 0.8 (Class 0: Recall=0.806, Precision=0.825; Class 1: Recall=0.830, Precision=0.812). F-Measure cũng rất cao cho cả hai lớp (~0.82).

-> Điều này cho thấy mô hình hoạt động tốt trên cả việc nhận diện lớp 0 và lớp 1.

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

* Số lượng dự đoán đúng (True Positives và True Negatives) cao (1356 và 1405).
* Số lượng dự đoán sai (False Positives và False Negatives) tương đối thấp (326 và 287).

**Nhận xét:**

* Cấu trúc đầu ra của cây quyết định tương đối đơn giản
* Các thuộc tính quan trọng nhất được mô hình lựa chọn để phân loại là Sulfate, Hardness, và ph.
* Mô hình đạt **hiệu suất rất tốt** trên bộ dữ liệu được sử dụng trong thử nghiệm này (Accuracy > 81%, Kappa > 0.63).
* Hiệu suất **cân bằng tốt** giữa hai lớp, với Recall và Precision đều trên 80% cho cả lớp 0 và lớp 1.

# **SO SÁNH KẾT QUẢ CỦA 2 THUẬT TOÁN**

* **Về Hiệu Suất Phân Loại:**
  + **C4.5 (J48):** Cho thấy hiệu suất vượt trội với độ chính xác (Accuracy) đạt **81.83%** và chỉ số Kappa là **0.6366**, thể hiện mức độ đồng thuận đáng kể và khả năng phân loại tốt. Các chỉ số Precision, Recall, và F1-Score cho cả hai lớp đều rất cân bằng và cao, cho thấy mô hình hoạt động hiệu quả trong việc nhận diện cả hai loại mẫu.
  + **ID3 (Java):** Hiệu suất thấp hơn đáng kể, với Accuracy chỉ đạt **69.63%**. Mặc dù Precision khá cao (0.75), nhưng Recall lại thấp hơn (0.6756), dẫn đến F1-Score là 0.7109. Điều này cho thấy mô hình ID3 tự triển khai có thể bỏ sót nhiều trường hợp nước uống được (False Negatives) hoặc gặp khó khăn trong việc cân bằng giữa độ chính xác và độ bao phủ.
* **Lý Giải Sự Khác Biệt:** Sự chênh lệch về hiệu năng này chủ yếu xuất phát từ những cải tiến cốt lõi của C4.5 so với ID3:  
  + **Xử lý thuộc tính liên tục:** C4.5 có khả năng tự tìm ngưỡng chia tối ưu cho các thuộc tính liên tục (như pH, Hardness, Sulfate...) dựa trên Gain Ratio, giữ được nhiều thông tin hơn. Ngược lại, ID3 yêu cầu phải rời rạc hóa dữ liệu liên tục trước (trong báo cáo này là chia thành 3 khoảng bằng Weka), quá trình này có thể làm mất thông tin và ngưỡng chia có thể không phải là tối ưu nhất.
  + **Sử dụng Gain Ratio:** ID3 sử dụng Information Gain, vốn có xu hướng thiên vị các thuộc tính có nhiều giá trị sau khi rời rạc hóa. C4.5 sử dụng Gain Ratio, giúp điều chỉnh sự thiên vị này, dẫn đến việc lựa chọn thuộc tính phân chia hợp lý hơn.
  + **Xử lý giá trị thiếu:** C4.5 có cơ chế nội tại để xử lý giá trị thiếu, trong khi với ID3, giá trị thiếu phải được xử lý hoàn toàn ở bước tiền xử lý. Cách xử lý của C4.5 thường linh hoạt và hiệu quả hơn.
  + **Cắt tỉa cây (Pruning):** Thuật toán C4.5 trong Weka thực hiện cắt tỉa sau (post-pruning) dựa trên confidenceFactor, giúp loại bỏ các nhánh ít tin cậy, giảm độ phức tạp của cây và chống lại hiện tượng quá khớp *(overfitting)*. ID3 không có cơ chế cắt tỉa, thường tạo ra cây phức tạp, bám sát dữ liệu huấn luyện nhưng lại hoạt động kém trên dữ liệu mới (dữ liệu kiểm tra), điều này giải thích phần nào cho Accuracy thấp hơn trên tập test của ID3.
  + **Triển khai:** Thuật toán J48 trong Weka là một triển khai chuẩn, đã được tối ưu và kiểm chứng rộng rãi. Việc tự triển khai ID3 bằng Java, dù tuân thủ lý thuyết, có thể chưa đạt mức tối ưu hoặc thiếu các kỹ thuật tinh chỉnh nhỏ có trong các thư viện chuẩn.
* **Cấu Trúc Cây Quyết Định:** Cây ID3 không cắt tỉa sẽ phức tạp hơn cây C4.5 đã được cắt tỉa. Kết quả C4.5 cho thấy một cây tương đối đơn giản và dễ diễn giải hơn.

**Kết luận so sánh:** Trên tập dữ liệu phân loại chất lượng nước này, thuật toán C4.5 (thông qua J48 của Weka) đã chứng tỏ hiệu quả vượt trội so với thuật toán ID3 tự triển khai. Các cải tiến của C4.5 trong việc xử lý dữ liệu liên tục, sử dụng Gain Ratio, xử lý giá trị thiếu và đặc biệt là cơ chế cắt tỉa chống overfitting đã đóng góp trực tiếp vào độ chính xác và độ tin cậy cao hơn của mô hình. Do đó, C4.5 là lựa chọn phù hợp hơn cho bài toán và tập dữ liệu đầu vào này.

# **TỔNG KẾT**

**1. Kết quả đạt được**

- Áp dụng thành công quy trình CRISP-DM: Đã thực hiện đầy đủ các bước của quy trình chuẩn trong khai phá dữ liệu, từ việc hiểu rõ bài toán nghiệp vụ, thu thập và tìm hiểu dữ liệu, tiền xử lý dữ liệu, xây dựng mô hình, đánh giá và chuẩn bị cho triển khai.

- Tiền xử lý dữ liệu hiệu quả: Đã xác định và xử lý thành công các vấn đề trong dữ liệu thô, bao gồm việc xử lý giá trị thiếu bằng phương pháp thay thế giá trị trung bình (sử dụng Weka), rời rạc hóa các thuộc tính liên tục để phù hợp với yêu cầu của thuật toán ID3, và cân bằng dữ liệu bằng kỹ thuật SMOTE để khắc phục sự mất cân bằng nhẹ giữa hai lớp mục tiêu.

- Triển khai thuật toán ID3 bằng Java: Đã xây dựng và triển khai thành công thuật toán ID3 từ đầu bằng ngôn ngữ Java, bao gồm các hàm tính toán Entropy, Information Gain, lựa chọn thuộc tính tốt nhất và xây dựng cấu trúc cây quyết định theo phương pháp đệ quy.

- Sử dụng thuật toán C4.5 qua Weka: Đã sử dụng công cụ Weka để áp dụng thuật toán J48, tận dụng khả năng xử lý thuộc tính liên tục, xử lý giá trị thiếu và cơ chế cắt tỉa của thuật toán này.

- Xây dựng và đánh giá mô hình phân loại: Đã xây dựng được hai mô hình cây quyết định (ID3 và C4.5) để phân loại khả năng uống được của nước. Các mô hình đã được đánh giá hiệu năng trên tập dữ liệu kiểm tra (ID3) hoặc thông qua kiểm định chéo 10 lần (C4.5) bằng các độ đo phổ biến như Accuracy, Precision, Recall, F1-Score và ma trận nhầm lẫn.

- So sánh hiệu năng thuật toán: Kết quả đánh giá cho thấy mô hình C4.5 (J48) với Accuracy ~81.8% và Kappa ~0.6366 tỏ ra vượt trội hơn hẳn so với mô hình ID3 tự triển khai (Accuracy ~69.6%). Điều này khẳng định ưu điểm của C4.5 trong việc xử lý thuộc tính liên tục và khả năng chống overfitting nhờ cơ chế cắt tỉa

**2. Nhược điểm**

Chất lượng dữ liệu: Mặc dù đã tiền xử lý, chất lượng của dữ liệu gốc (ví dụ: phương pháp đo, sai số) vẫn có thể ảnh hưởng đến kết quả. Việc xử lý giá trị thiếu bằng cách thay thế giá trị trung bình/trung vị có thể không phải là tối ưu nhất.

Phương pháp cân bằng dữ liệu: SMOTE giúp cân bằng lớp nhưng cũng có thể tạo ra nhiễu nếu các lớp không tách biệt rõ ràng.

Rời rạc hóa: Việc chọn số lượng khoảng (bins) và phương pháp rời rạc hóa cho ID3 có thể ảnh hưởng đáng kể đến kết quả.

Giới hạn thuật toán: Chỉ sử dụng ID3 và C4.5. Các thuật toán khác (như Random Forest, Gradient Boosting, SVM, Neural Networks) có thể cho hiệu năng tốt hơn.

Triển khai ID3: Việc triển khai ID3 thủ công bằng Java có thể tiềm ẩn lỗi và có thể không tối ưu bằng các thư viện có sẵn. Không có cơ chế cắt tỉa tích hợp sẵn.

**3. Hướng phát triển**

Thử nghiệm thuật toán khác: Áp dụng và so sánh với các thuật toán phân loại mạnh mẽ hơn như Random Forest (cũng dựa trên cây quyết định nhưng tổng hợp kết quả từ nhiều cây), Gradient Boosting (XGBoost, LightGBM), SVM, hoặc mạng nơ-ron.

Kỹ thuật Feature Engineering: Tạo ra các thuộc tính mới từ các thuộc tính hiện có (ví dụ: tỷ lệ giữa hai chất) hoặc sử dụng các phương pháp lựa chọn thuộc tính nâng cao hơn để tìm ra tập con thuộc tính tối ưu.

Tinh chỉnh tham số (Hyperparameter Tuning): Tối ưu hóa các tham số của thuật toán J48 bằng các kỹ thuật như Grid Search hoặc Random Search trong Weka (thông qua Experimenter hoặc Knowledge Flow).

Đánh giá sâu hơn: Phân tích kỹ hơn các trường hợp dự đoán sai (False Positives, False Negatives) để hiểu rõ hơn điểm yếu của mô hình. Sử dụng các kỹ thuật diễn giải mô hình khác (ví dụ: SHAP, LIME) nếu áp dụng các thuật toán phức tạp hơn.

Dữ liệu lớn hơn/đa dạng hơn: Thu thập thêm dữ liệu từ nhiều nguồn nước khác nhau hoặc trong các điều kiện khác nhau để xây dựng mô hình tổng quát hơn.

# **TÀI LIỆU THAM KHẢO**

1. Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1), 81–106.
2. Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.
3. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques (4th ed.)*. Morgan Kaufmann.