 **PHÂN HIỆU TRƯỜNG ĐẠI HỌC THUỶ LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



**BÀI TẬP LỚN MÔN**

**HỌC MÁY – MACHINE LEARNING**

**Đề tài:** **Dự Đoán Chất Lượng Nước Cho Người Dùng**

**Water Quality Analysis**

Giáo viên hướng dẫn: ThS. Vũ Thị Hạnh

Lớp : S24 – CNTT02

Nhóm thực hiện : Nhóm 12

Thành viên nhóm : Nguyễn Văn Tân

Phan Công Hiếu

Nguyễn Phong Vũ

Khưu Anh Minh

Thành phố Hồ Chí Minh, ngày 24 tháng 12 năm 2023

# **LỜI NÓI ĐẦU**

Hiện nay, vấn đề biến đổi khí hậu đang là một trong những vấn đề nhức nhối toàn cầu. Nguyên nhân chủ yếu dẫn đến biến đổi khí hậu là do hoạt động sản xuất của con người. Hậu quả mang lại mà con người phải trả giá là sức khỏe, môi trường sống bị suy giảm đến mức trầm trọng. Trong đó, hiện trạng nguồn nước bị ô nhiễm nghiêm trọng đã và đang làm ảnh hưởng ít nhiều đến nguồn nước mà chúng ta đang sử dụng. Đó là thiếu nước sạch.

Nước sạch rất quan trọng trong cuộc sống của chúng ta, sử dụng nguồn nước sạch sẽ đảm bảo an toàn cho sức khỏe. Đối với cơ thể con người nước chiếm đến hơn 70% ở lúc sơ sinh và giảm xuống còn khoảng 60% khi trưởng thành, và tận 85% khối lượng bộ não được cấu tạo từ nước. Trong cơ thể chúng ta nước đóng vai trò là dung môi cho những phản ứng hóa học xảy ra trong cơ thể. Nước vận chuyển tất cả những nguyên tố dinh dưỡng đến toàn bộ cơ thể. Giúp điều hòa thân nhiệt bằng tuyến mồ hôi… con người vẫn có thể sống sót nếu nhịn ăn trong mấy tuần, nhưng lại không thể tồn tại được nếu thiếu nước trong khoảng 3-4 ngày.

Do đó, nhóm chúng em đã lựa chọn đề tài “Water Quality” để làm báo cáo đề tài môn “Học Máy” vì nhận thấy: tiếp cận nguồn nước sạch là một trong những quyền cơ bản của con người và là một phần trong chính sách bảo vệ sức khỏe. Đây là vấn đề quan trọng tại từng quốc gia, từng khu vực và từng địa phương. Việc đầu tư vào làm sạch và cung cấp nguồn nước sạch cho người dân có thể mang lại sự phát triển kinh tế vượt bậc vì đối với việc làm giảm những tác động xấu ảnh hưởng đến sức khỏe và chăm sóc sức khỏe sẽ có chi phí cao hơn việc xây dựng nhà máy lọc nước.

Báo cáo gồm 6 phần:

Phần 1: Tổng quan về khai phá dữ liệu

Phần 2: Tiền xử lý dữ liệu

Phần 3: Các thuật toán áp dụng

Phần 4: Đánh giá thuật toán

Phần 5: Kết luận và phát triển

Phần 6: Tài liệu tham khảo

# **LỜI CẢM ƠN**

Đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến tập thể quý Thầy Cô của Phân hiệu trường Đại học Thủy Lợi và quý Thầy, Cô khoa Công nghệ thông tin đã giúp cho nhóm chúng em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Đặc biệt, nhóm chúng em xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Cô giáo, ThS Vũ Thị Hạnh (Giảng viên môn Học máy). Các Thầy Cô đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến quý báu giúp nhóm tác giả hoàn thành tốt báo cáo môn học của mình. Trong thời gian một học kỳ thực hiện đề tài, nhóm chúng em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, nhóm tác giả vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất.

Tuy nhiên, trong quá trình thực hiện không tránh khỏi những thiếu sót. Chính vì vậy, nhóm chúng em rất mong nhận được những sự góp ý từ phía các Thầy Cô nhằm hoàn thiện những kiến thức mà nhóm chúng em đã học tập và là hành trang để nhóm chúng thực hiện tiếp các đề tài khác trong tương lai.

Xin chân thành cảm ơn các quý Thầy Cô!

**Nhóm sinh viên thực hiện**

# **PHẦN I. Tổng Quan Về Khai Phá Dữ Liệu.**

Phát hiện tri thức và khai phá dữ liệu

Phát hiện tri thức (Knowledge Discovery) trong các cơ sở dữ liệu là một qui trình nhận biết các mẫu hoặc các mô hình trong dữ liệu với các tính năng: hợp thức, mới, khả ích và có thể hiểu được. [1]

Khai phá dữ liệu (Data mining) được định nghĩa như sau: “Data mining là một quá trình tìm kiếm, phát hiện các tri thức mới, tiềm ẩn, hữu dụng trong CSDL lớn”.

Khai phá dữ liệu có thể được sử dụng cho các lĩnh vực y tế, phân tích thị trường, xây dựng ... có thể được xem như là kết quả của sự tiến triển tự nhiên của công nghệ thông tin.

Mục tiêu và ý nghĩa

* + 1. Mục Tiêu:

Lý do chọn đề tài về "Dự án Machine Learning có tiêu đề Chất Lượng Nước" là do chất lượng nước đang là một vấn đề quan trọng và ảnh hưởng đến sức khỏe con người và môi trường

Khai phá dữ liệu có thể được sử dụng cho các lĩnh vực y tế, phân tích thị trường, xây dựng ... có thể được xem như là kết quả của sự tiến triển tự nhiên của công nghệ thông tin.

Sử dụng mô hình Maching Learning dự đoán chất lượng nước có biến số đo lường như aluminium, ammonia, arsenic, barium, cadmium, chloramine, chromium, copper, fluoride, bacteria, lead, nitrates, nitrites, mercury, perchlorate, radium, selenium, silver, uranium có an toàn không? " hoặc " Các chất này có an toàn cho sức khỏe không?"

Phân tích được các mối quan hệ giữa các yếu tố khác nhau đối với chất lượng nước để hiểu rõ hơn về nguyên nhân và ảnh hưởng của chúng.

Tạo ra hệ thống cảnh báo để phát triển nó dựa trên Maching Learning để thông báo về các thay đổi đột ngột trong chất lượng nước và đưa ra các biện pháp đối phó.

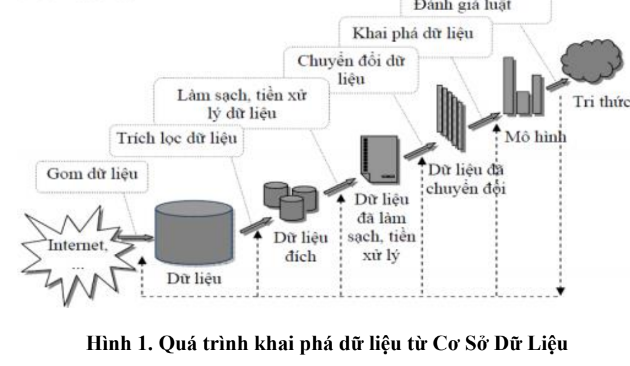
* + 1. Ý nghĩa:

Sự an toàn của các chất này, cần phải xem xét mức độ nồng độ và loại chất trong nước. Một số chất có thể gây hại cho sức khỏe nếu chúng tồn tại ở mức độ cao. Các tiêu chuẩn an toàn được đặt ra để đảm bảo rằng nước uống không chứa các chất độc hại ở mức độ đe dọa đến sức khỏe con người.

Việc kiểm tra chất lượng nước thường xuyên và tuân thủ các tiêu chuẩn an toàn là quan trọng để bảo vệ sức khỏe cộng đồng bằng cách cung cấp thông tin chính xác và kịp thời về chất lượng nước.

Tối ưu hoá quản lý nguồn nước và hiểu rõ sâu sắc về chất lượng nước để hỗ trợ một cách hiểu quả, giúp ngăn chặn sự ô nhiễm và bảo vệ nguồn nước trong tương lai.

Quy trình khám phá tri thức trong CSDL



Mô tả bài toán

* + 1. Tổng quan bài toán

Nước sạch rất quan trọng trong cuộc sống của chúng ta, sử dụng nguồn nước sạch sẽ đảm bảo an toàn cho sức khỏe, tuy nhiên, ngoài nguồn nước sạch được sử dụng cho các bữa ăn, sinh hoạt thường ngày còn có những nguồn nước “bẩn”. Việc phân biệt giữa nguồn nước sạch và không sạch tưởng chừng đơn giản nhưng không.

Một vài quan niệm sai lầm cho rằng: “Nước sạch là nước không mùi, màu hơi trong hoặc trong. Đun sôi lên sẽ thành nước sạch”, quan niệm này hoàn toàn sai lầm. Nước sạch đúng là không mùi, màu hơi trong hoặc trong nhưng không phải nguồn nước nào có đặc điểm như vậy đều là nguồn nước sạch.  Ngoài những đặc tính về mùi và màu sắc, thì nước còn có độ, các khoáng chất, ... Nếu như không dùng những phương pháp hiện đại kiểm chứng thì khi uống phải nguồn nước chứa độ ph cao hay độ khoáng chất quá 1000mg/l sẽ ảnh hưởng không tốt đến sức khỏe.

Để hiểu rõ và xác định cái gì quyết định nguồn nước sạch và nguồn nước không sạch, nhóm đã quyết định chọn đề tài phân tích, khai thác dữ liệu về các đặc tính của nước.



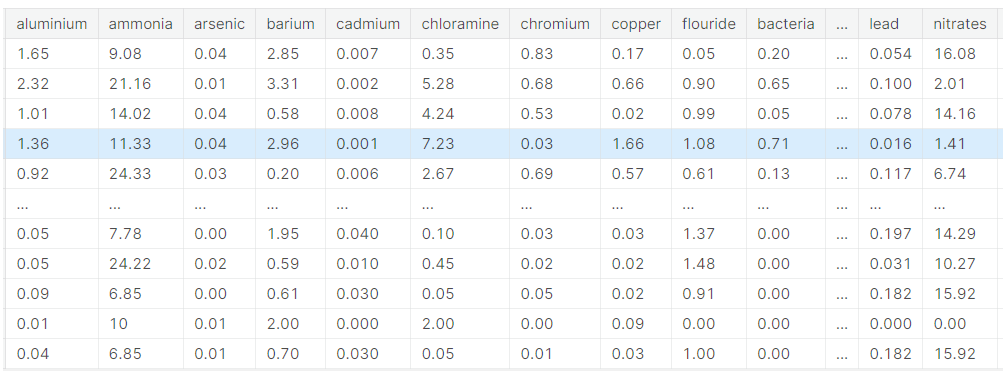
* + 1. Phân tích dữ liệu

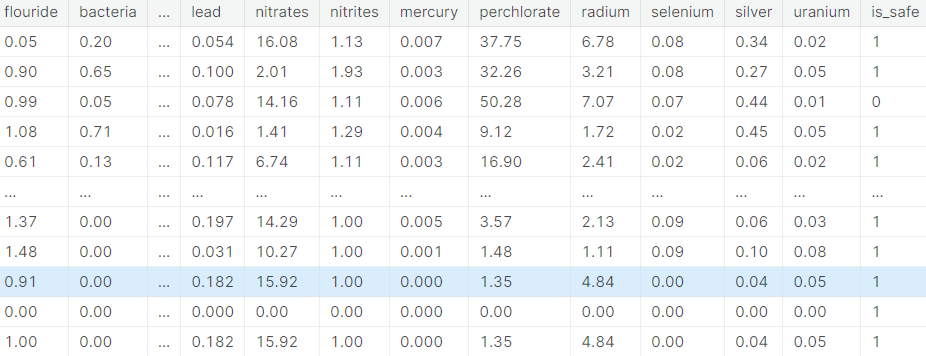
Bộ dữ liệu mô tả về các mẫu giả định tương ứng với 3276 nguồn nước khác nhau. Bộ dữ liệu này nghiên cứu và cho ra kết luận rằng khả năng mẫu nước này có dùng được hay không dùng được của từng mẫu.

Link dataset: <https://www.kaggle.com/datasets/adityakadiwal/water-potability>

Bộ dữ liệu này gồm 10179 và 21 cột thuộc tính.

Mỗi cột đại diện cho một nguồn nước và chúng được dán nhãn là có thể uống được hay không uống được. Sau khi đi kiểm tra thì có nhiều giá trị bị trùng lặp, kiểu thuộc tính thì kiểu object và float. Đối với một số mô hình, dữ liệu là được tạo thành một tập hợp và một tập hợp thử nghiệm không thể gắn nhãn. Các nhãn đã được kiểm tra đảm bảo chúng là những yếu tố. Không có giá trị hoặc ngoại không chính xác nào được quan sát thấy.





Hiểu về thuộc tính và kiểu dữ liệu:

|  |  |  |  |
| --- | --- | --- | --- |
| Stt | Tên thuộc tính | Ý nghĩa | Kiểu dữ liệu |
| 1 | Aluminium(Nhôm) | Aluminium (Nhôm): Kim loại nhẹ, thường xuất hiện trong nước do quá trình đất đá hóa. | Float |
| 2 | Ammonia (Amôniac) | Hợp chất hóa học chứa nitrogen và hydrogen, thường xuất hiện trong nước do quá trình phân giải hữu cơ. | Ban đầu là object sau khi đi tiền xử lý về kiểu Float |
| 3 | Arsenic (Asen) | Nguyên tố kim loại, có thể xuất hiện tự nhiên trong nước hoặc do hoạt động công nghiệp. | Float |
| 4 | Barium (Bari) | Nguyên tố kim loại, thường được sử dụng trong nhiều ứng dụng công nghiệp. | Float |
| 5 | Cadmium (Cadmi) | Kim loại nặng, thường xuất hiện trong nước từ các nguồn công nghiệp. | Float |
| 6 | **chloramine** | Nồng độ của chloramine trong nước. | Float |
| 7 | Chromium (Crôm) | Nguyên tố kim loại, có thể xuất hiện tự nhiên hoặc do hoạt động công nghiệp. | Float |
| 8 | Copper (Đồng) | Kim loại, thường xuất hiện trong nước từ ống nước hoặc các nguồn khác | Float |
| 9 | Fluoride (Fluorua): | Anion có chứa fluor, thường được thêm vào nước để cung cấp fluoride cho sức khỏe răng | Float |
| 10 | Bacteria (Vi khuẩn): | Vi khuẩn có thể xuất hiện trong nước và ảnh hưởng đến sức khỏe. | Float |
| 11 | Lead (Chì) | Kim loại nặng, thường xuất hiện trong nước từ ống nước cũ hoặc do hoạt động công nghiệp. | Float |
| 12 | Nitrates (Nitrat) | Hợp chất chứa nitơ và oxy, thường xuất hiện trong nước do sử dụng phân bón và nước thải. | Float |
| 13 | Nitrites (Nitrit) | Hợp chất chứa nitơ và oxy, thường xuất hiện trong nước như một sản phẩm phụ của quá trình chuyển đổi nitrates. | Float |
| 14 | Mercury (Thủy ngân) | Kim loại nặng, thường xuất hiện trong nước từ hoạt động công nghiệp và môi trường tự nhiên. | Float |
| 15 | Perchlorate (Perclorat) | Hợp chất hóa học chứa perchlorate ion, có thể xuất hiện từ chất béo hóa và các nguồn khác. | Float |
| 16 | Radium (Radium) | Nguyên tố phóng xạ, có thể xuất hiện tự nhiên trong nước | Float |
| 17 | Selenium (Seleni) | Nguyên tố vi lượng, có thể xuất hiện trong nước từ đất đai và nước mưa. | Float |
| 18 | Silver (Bạc) | Kim loại, có thể xuất hiện trong nước từ nguồn nước tự nhiên và hoạt động công nghiệp. | Float |
| 19 | Uranium (U-ran) | Kim loại, có thể xuất hiện trong nước từ nguồn nước tự nhiên và hoạt động công nghiệp. | Float |
| 20 | Is\_safe | Nguyên tố phóng xạ, có thể xuất hiện tự nhiên trong nước. | Ban đầu là object sau khi đi tiền xử lý về kiểu Float |

# **PHẦN II. PHẦN TIỀN XỬ LÝ.**

Có rất nhiều kỹ thuật thao tác dữ liệu được lưu trong dạng ndarray. Nhưng để áp dụng học sâu vào giải quyết các vấn đề thực tế, ta thường phải bắt đầu bằng việc xử lý dữ liệu thô, chứ không có luôn dữ liệu ngăn nắp được chuẩn bị sẵn trong định dạng ndarray. Trong số các công cụ phân tích dữ liệu phổ biến của Python, gói pandas khá được ưa chuộng. Cũng như nhiều gói khác trong hệ sinh thái rộng lớn của Python, ‘pandas’ có thể được sử dụng kết hợp với định dạng ndarray. Vì vậy, chúng ta sẽ đi nhanh qua các bước để tiền xử lý dữ liệu thô bằng pandas rồi đổi chúng sang dạng ndarray.

1. Tìm hiểu khái quát dữ liệu gốc

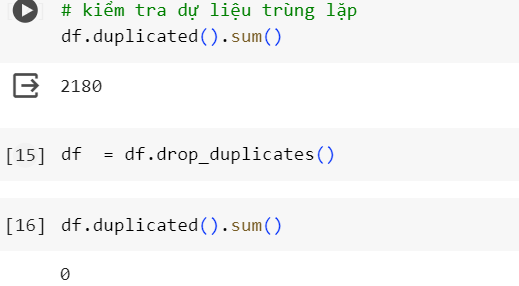
Đối với từng thuộc tính như aluminium, ammonia, arsenic, và các thuộc tính khác, "gốc" có thể liên quan đến nguồn gốc của chúng. Điều này có thể là phương pháp đo lường, thiết bị đo, hay nơi mà dữ liệu được thu thập (ví dụ: từ các trạm đo chất lượng nước).

Và biến mục tiêu là is\_safe có thể là biến nhị phân 0 và 1 thể hiện tính tương đương như có thể giả định rằng giá trị 1 có thể biểu thị nước an toàn, trong khi giá trị 0 có thể biểu thị nước không an toàn.

Làm sạch dữ liệu

Là quá trình nhận dạng dữ liệu đã có để tiến hành xử lý các dữ liệu bị thiếu (missing data) và dữ liệu trùng lặp, dán nhãn có những kiểu dữ liệu ban đầu ở dạng object.

Ở bộ dữ liệu Water Quality, nhóm chúng em đã tiến hành kiểm tra dữ liệu bị thiếu nhưng trong dữ liệu này của em không có dữ liệu thiếu nên kiểm tra dữ liệu bị trùng lặp và thấy được có 2180 dữ liệu bị trùng ngay lặp tức đi xử lý và trả về trạng thái 0.



Và tiếp tục chuyển đổi 2 kiểu dữ liệu object khi đi tiền xử lý phát hiện đó là ammonia và is\_safe cả hai thuộc tính này đều có dữ liệu ở trong NUM!

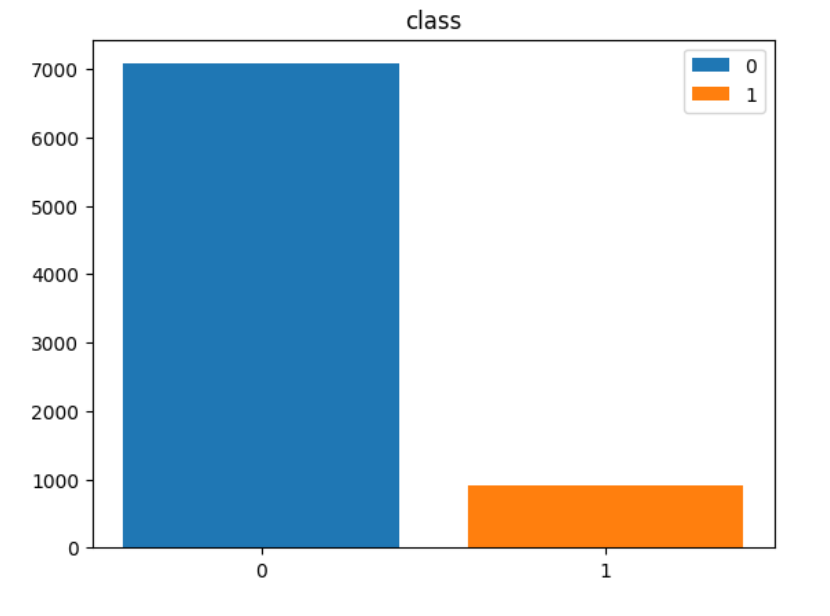
Nên chúng em thứ nhất xử lý ammonia dùng Type để đưa về class ‘str’ sau đó sử dụng hàm to\_numeric(df['ammonia'], errors='coerce'). Tham số errors='coerce' sẽ chuyển đổi các giá trị không thể chuyển đổi NaN nên dùng hàm này thì NaN vẫn giữ nguyên. Sau khi xong đi kiểm tra lại dữ liệu đã chuyển về float chưa.

Thứ hai chúng em đã chuyển is\_safe f['is\_safe'] = df['is\_safe']. astype(str): Dòng này chuyển đổi cột 'is\_safe' của DataFrame df sang kiểu dữ liệu chuỗi (string). Việc này đảm bảo rằng mọi giá trị trong cột 'is\_safe' đều được biểu diễn dưới dạng chuỗi.

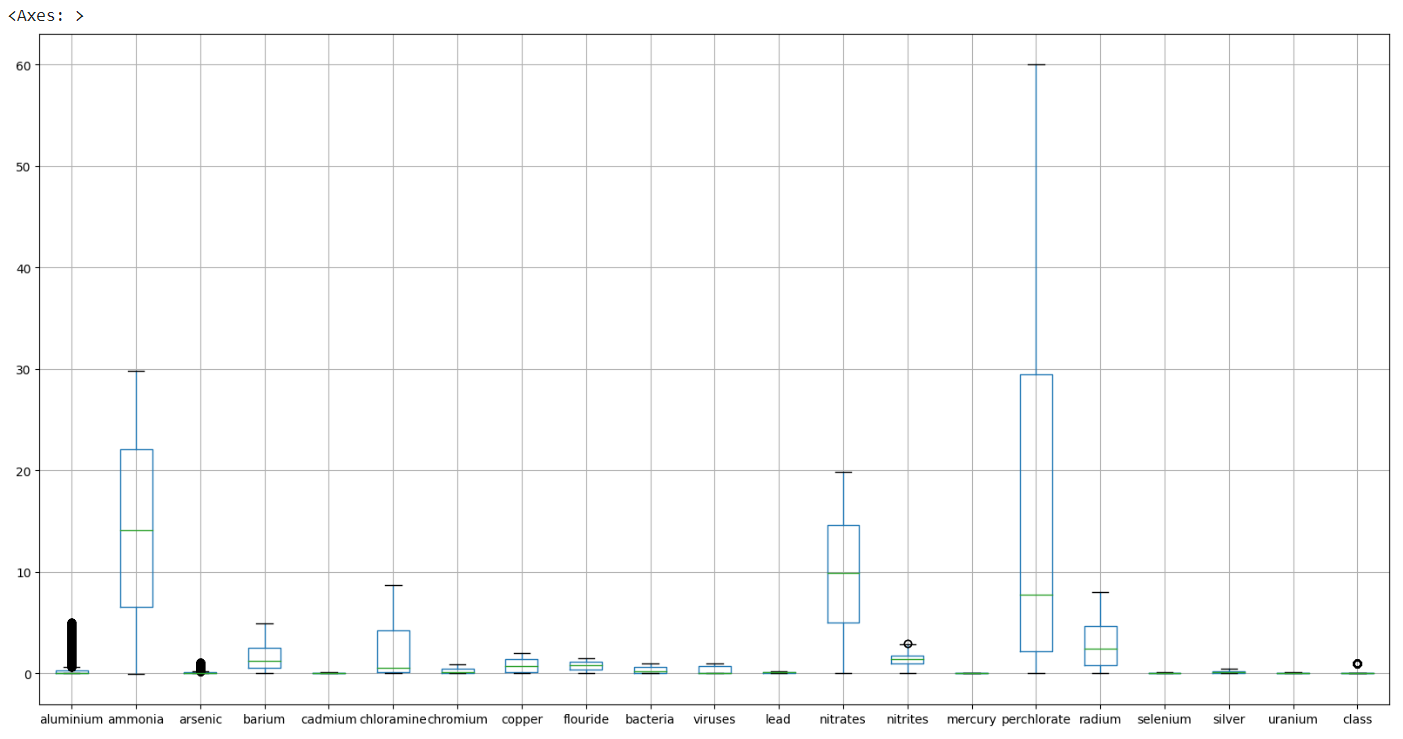
df = df[~df['is\_safe']. str. contains('#NUM!')]: Dòng này lọc các hàng trong DataFrame df trong đó giá trị trong cột 'is\_safe' không chứa chuỗi '#NUM!'. Ký hiệu ~ được sử dụng để đảo ngược điều kiện, nghĩa là chỉ giữ lại các hàng không chứa chuỗi '#NUM!' trong cột 'is\_safe'.

Tóm lại, hai dòng mã trên cùng nhau đảm bảo rằng cột 'is\_safe' đã được chuyển đổi sang kiểu dữ liệu chuỗi và sau đó lọc bỏ các hàng có giá trị '#NUM!' trong cột 'is\_safe'. Điều này có thể hữu ích khi bạn muốn loại bỏ những giá trị không mong muốn hoặc không hợp lý trong dữ liệu

Và cuối cùng biến mục tiêu chuyển về một class có giá trị [0, 1] và được đưa về kiểu float trong đó dữ liệu 0 chiếm 7076 và 1 chiếm được 910 cho thấy được nó đang không cân bằng về dữ liệu. Dữ liệu sau khi đã xử lý:

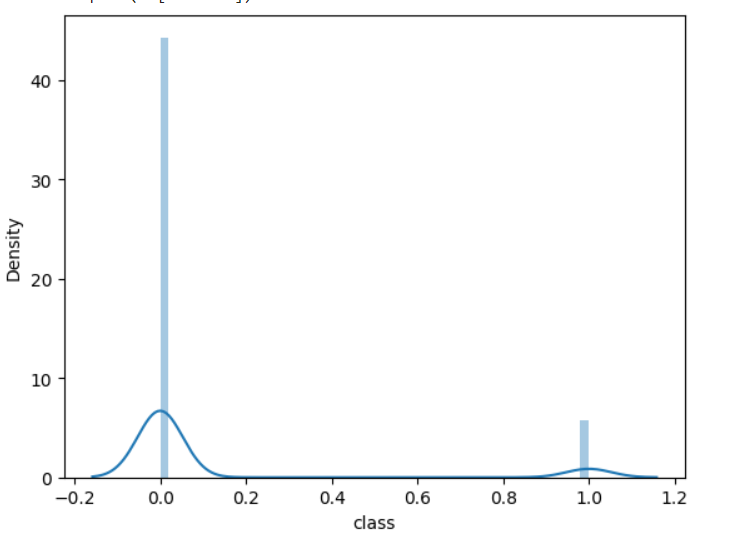


Sau đó chúng em đi vẽ các biểu đồ hộp (boxplot) để thể hiện phân phối của các thuộc tính trong dữ liệu nước. Dưới đây là giải thích chi tiết:



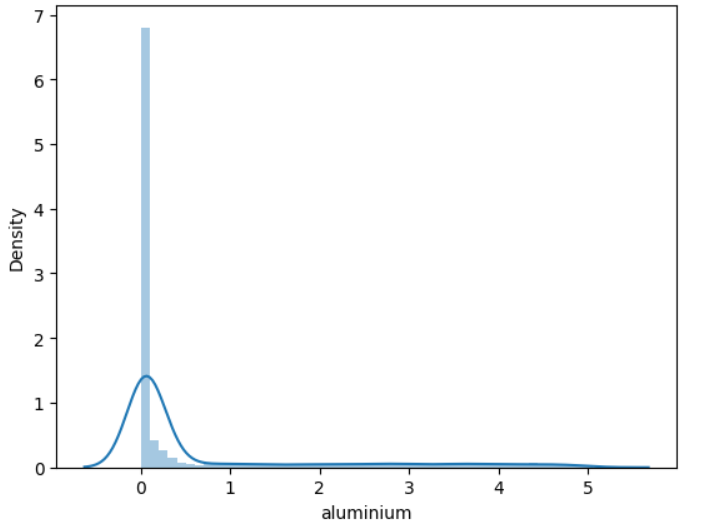
Nhìn vào biểu chúng em thấy rằng perchlorate đang mô tả được nguồn dữ liệu cao nhất khi chúng em describe thấy được 25% đang chiếm khoảng 2, 50% chiếm khoảng 8, 75% chiếm khoảng 29 và max chiếm đến 60. Thuộc tính perchlorate không có outliers.

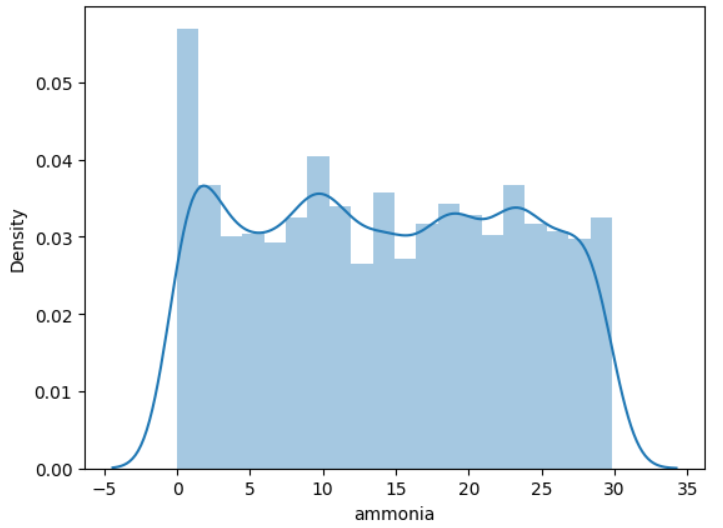
Theo số liệu thu được thì số lượng nguồn nước sạch - uống được (Potability) là 1891, nguồn nước không sạch - không uống được (Not\_Potability) là 1223.

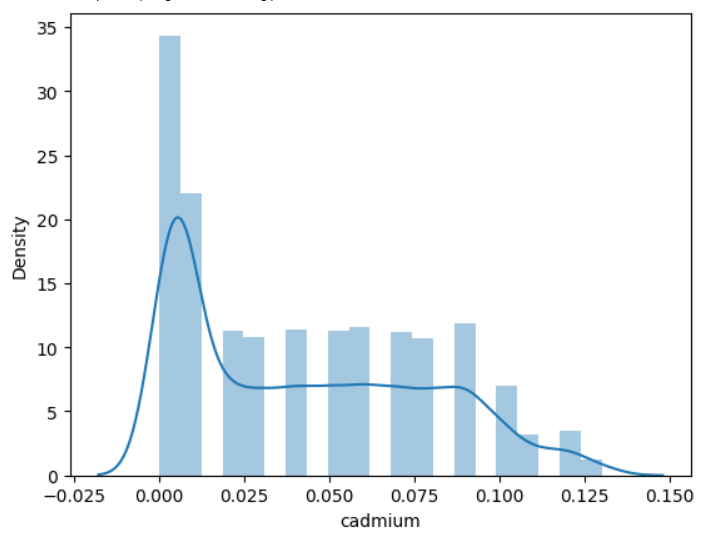


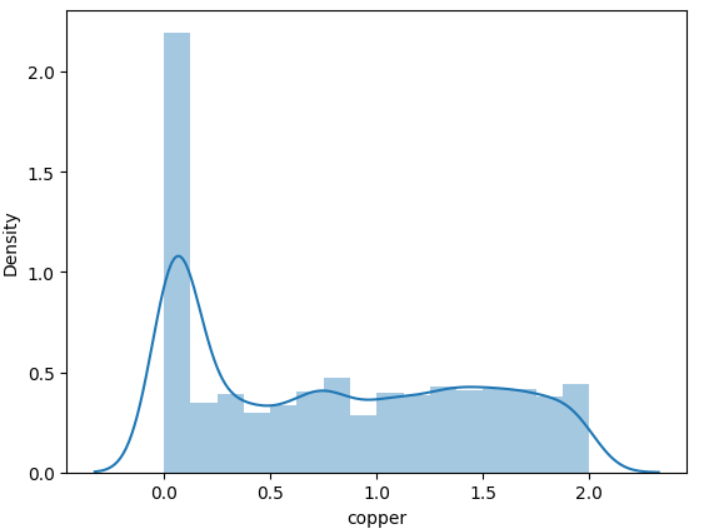
Nhận xét: nhìn vào biểu đồ trên, chúng ta nhận thấy tập dữ liệu chưa cân bằng.

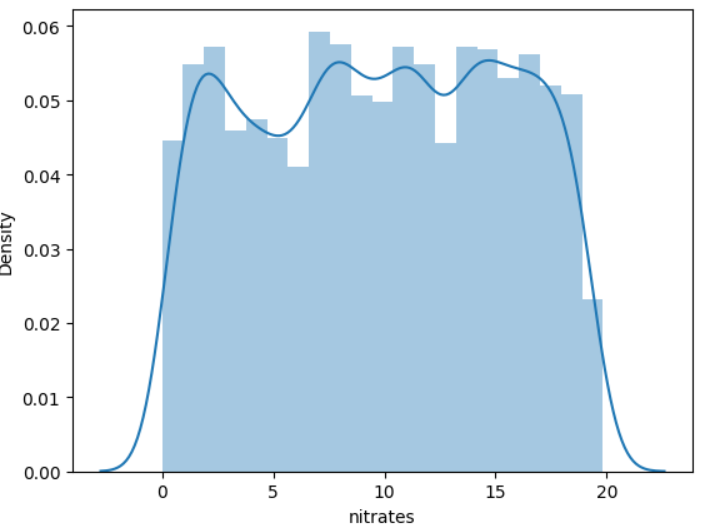
Dưới đây là những biểu đồ thống kê các thuộc tính có trong tập dữ liệu Water Quality.

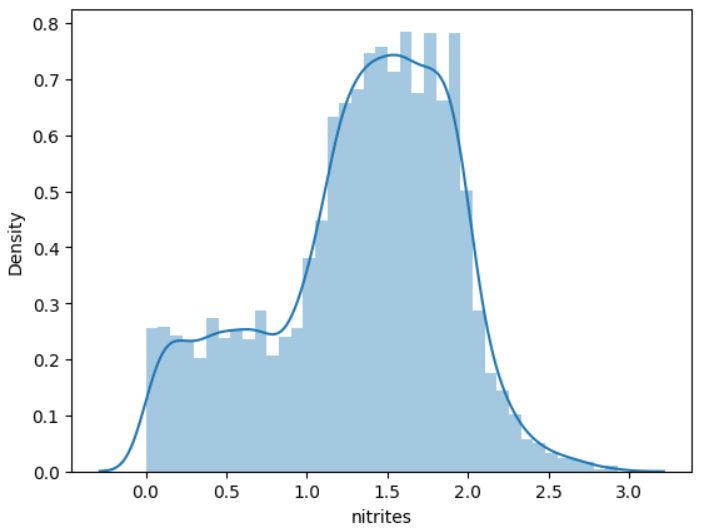


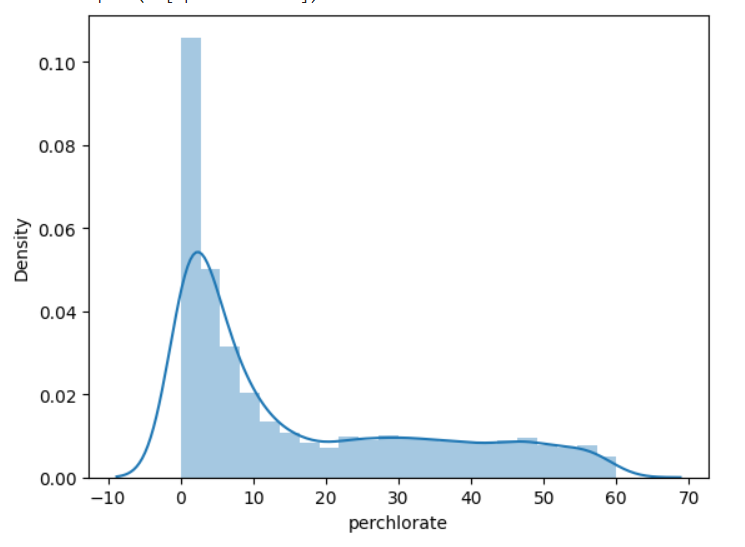


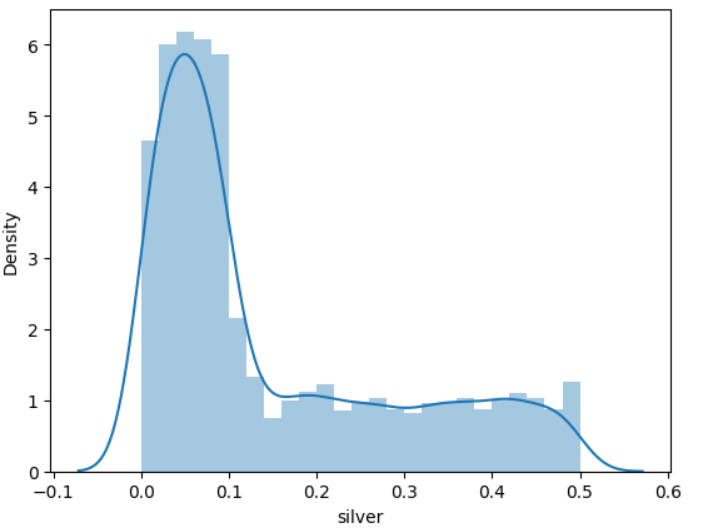


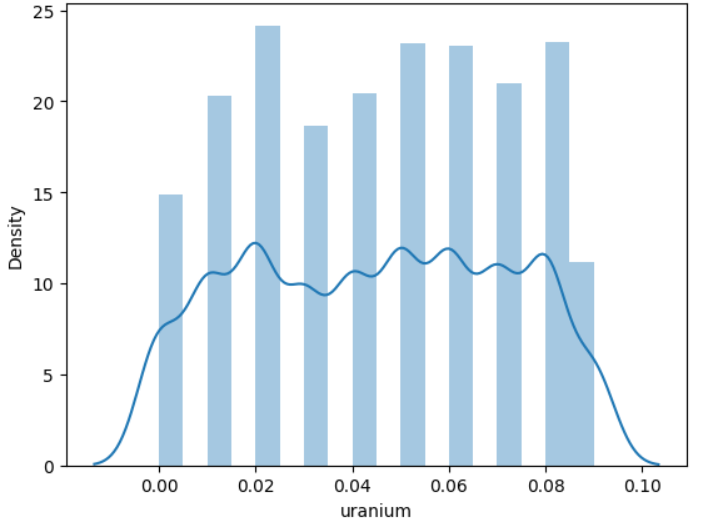












Phương pháp chuẩn hóa (scaling) dữ liệu thường được sử dụng trong quá trinh tiền lý xử lý trong quá trình xây đựng và đào tạo mô hình học máy. Nên bài tập lớn này của chúng em đã sử dụng MinMaxScaler.

Phương pháp chuẩn hóa này dùng để chuyển đổi dữ liệu sao cho nó nằm trong khoảng cụ thể [0, 1] hoặc [-1, 1]. Điều này giúp làm cho các biến có thang đo khác nhau có thể được so sánh và so sánh dễ dàng hơn.

Min-MaxScaler không bị ảnh hưởng nhiều bởi các outliers, vì nó chỉ quan tâm đến khoảng giá trị tuyệt đối của dữ liệu và không phản ứng mạnh giữa các giá trị ngoại lai.

Công thức chuyển đổi:



Công thức này khi chuyển đổi nhưng nó vẫn giữ được phân phối tuyến tính của dữ liệu.

Do dự án này chúng em có sử dụng thuật toán SVM nên Min-MaxScaler là một sự lựa chọn phù hợp để đảm bảo đầu vào của mô hình nằm trong phạm vi mong muốn. Và lưu ý rằng phương pháp này sẽ không áp dụng nếu như dữ liệu outliers mạnh hoặc khi phân phối biến nó quá chênh lệch.

# **PHẦN III. TỔNG QUAN VỀ CÁC THUẬT TOÁN SỦ DỤNG**

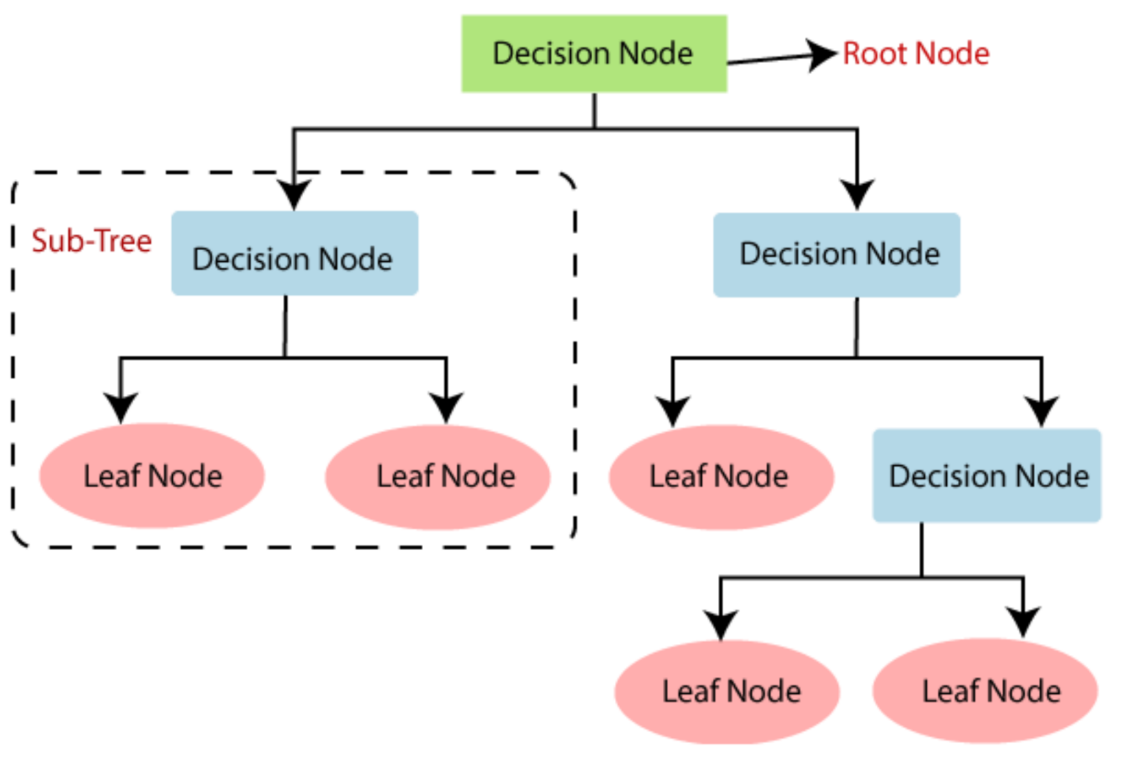
1. Thuật toán cây quyết định – Decision Tree

* + 1. Giới thiệu

Thuật toán cây quyết định là một phướng pháp học có giám sát được sử dụng trong thống kê, khai thác dữ liệu và học máy. Có thể được dùng để giải quyết bài toán hồi quy (Regression) và phân loại (Classification)

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Cây quyết định phân loại các cá thể bằng cách sắp xếp chúng theo cây từ gốc đến một số nút lá, điều này cung cấp sự phân loại của cá thể. Một cá thể được phân loại bằng cách bắt đầu từ nút gốc của cây, kiểm tra thuộc tính được chỉ định bởi nút này, sau đó di chuyển xuống nhánh cây tương ứng với giá trị của thuộc tính như trong hình trên. Quá trình này sau đó được lặp lại cho cây con bắt nguồn từ nút mới. Là mô hình đưa ra quyết định dựa trên các câu hỏi.

Về cấu trúc Decision Trees gồm 3 phần chính: 1 node gốc (root node), những node lá (leaf nodes) và các nhánh của nó (branches). Node gốc là điểm bắt đầu của cây quyết định và cả hai node gốc và node chứa câu hỏi hoặc tiêu chí để được trả lời. Nhánh biểu diễn các kết quả của kiểm tra trên nút.



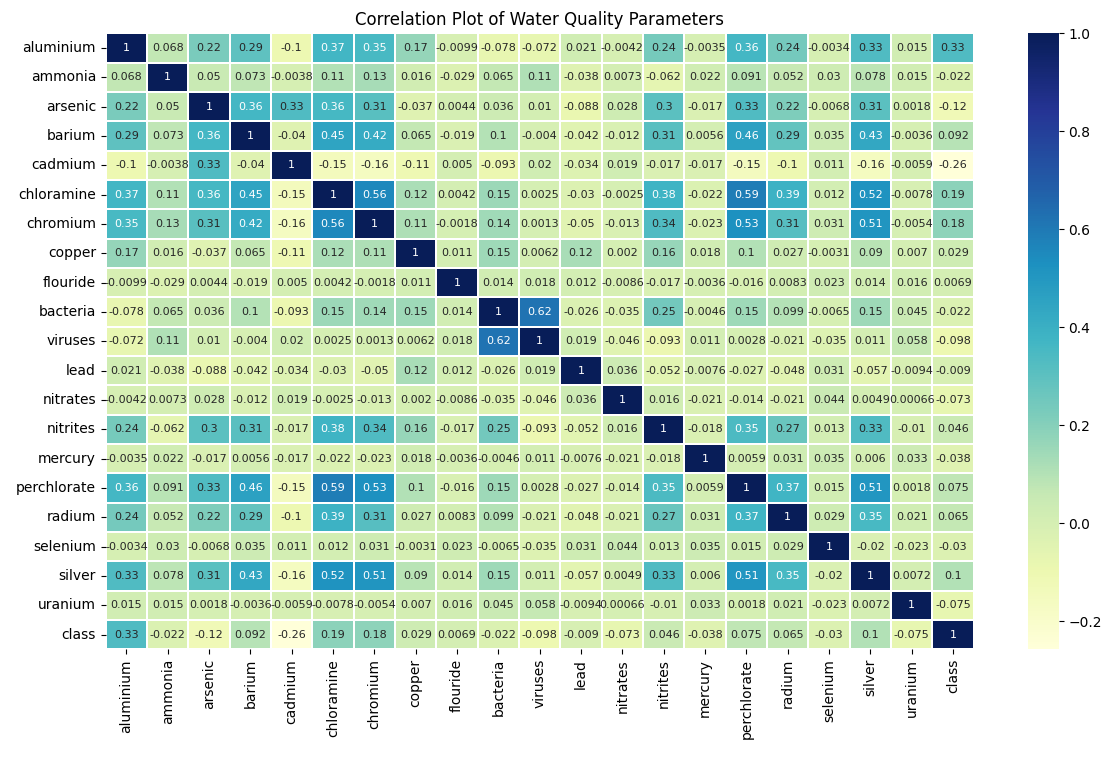
Một thuật toán học máy thường sẽ có:

* Huấn luyện: Từ dữ liệu huấn luyện thuật toán sẽ học và tạo ra mô hình.
* Dự đoán: Dùng mô hình học được từ bước trên dự đoán ra các giá trị mới.

Thuật toán cho cây quyết định:

* Cây quyết định được xây dựng từ tập dữ liệu huấn luyện bằng cách phân chia và chinh phục đệ quy từ trên xuống.
* Khi bắt đầu, tất cả các ví dụ đào tạo đều ở gốc.
* Các thuộc tính được phân loại (nếu có giá trị liên tục, chúng dược phân chia rời rạc trước).
* Các ví dụ được phân vùng đệ quy dựa trên các tuộc tính đã chọn.
* Các thuộc tính kiểm tra được chọn trên cơ sở đo lường thống kê hoặc kinh nghiệm (thu thập thông tin - information gain)
* Dừng phân vùng khi tất cả các mẫu cho một nút nhất định thuộc cùng một lớp hoặc không còn thuộc tính nào để phân vùng hoặc không còn mẫu nào
* Để lựa chọn tuộc tính phân vùng tốt cho mô hình cần sử dụng các kỹ thuật tính toán lựa chọn thuộc tính. VD như Information Gain, Gini Index.
  + 1. Ưu điểm & nhược điểm của thuật toán
* Ưu điểm:
* Tính phổ biến: là một thuật toán đơn giản và được dùng trong cả bài toán phân lớp ᴠà hồi quу nên cây quуết định là một trong những thuật toán máу học phổ biến nhất hiện nay.
* Dễ hiểu: biểu diễn của cây quyết định giúp chúng ta dễ hiểu hơn, nó tạo ra bộ luật với mỗi nhánh lá là một luật của cây. Bản chất phân cấp của cây quyết định cũng giúp bạn dễ dàng thấy thuộc tính nào là quan trọng nhất.
* Ít hoặc không cần chuẩn bị dữ liệu: cây quyết định có thể xử lý các giá trị bị thiếu, không cần chuẩn hóa hoặc tạo biến giả
* Tính linh hoạt: trong các mối quan hệ cơ bản giữa các thuộc tính, nếu hai biến có tương quan cao, thuật toán sẽ chỉ chọn một trong các đặc điểm để tách.
* Xử lý tốt dữ liệu lớn trong thời gian ngắn: thuật toán thể làm việc với cả dữ liệu số và dữ liệu phân loại và có khả năng làm việc với bộ dữ liệu lớn.
* Nhược điểm:
* **Dễ bị overfitting**: Cây quyết định phức tạp có xu hướng quá mức và không tổng quát hóa tốt cho dữ liệu mới.
* **Tốn kém hơn:** Do cây quyết định có cách tiếp cận tìm kiếm tham lam trong quá trình xây dựng, vì vậy chúng có thể tốn kém hơn so với các thuật toán khác.
* **Ít chính xác với dữ liệu nhỏ:** Cây quyết định dễ mắc lỗi trong các bài toán phân loại với nhiều lớp và số lượng ví dụ huấn luyện tương đối nhỏ.
  + 1. Triển khai

Thông qua biểu đồ heatmap dưới đây, ta thực hiện xem xét mối tương quan giữa các biến.

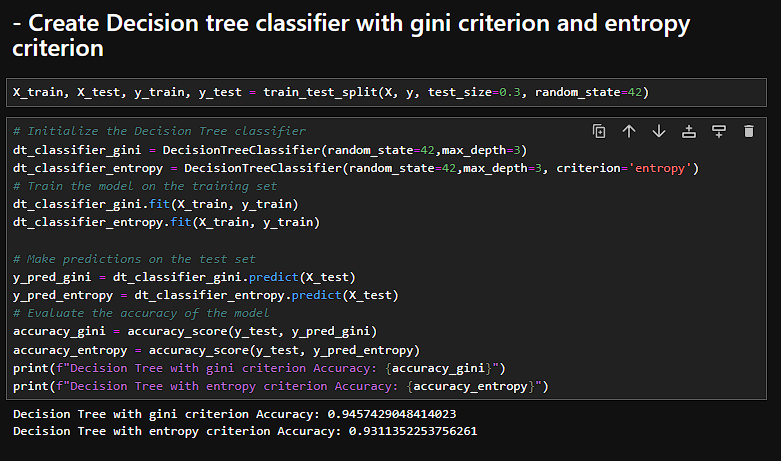


* + 1. Xử lý dữ liệu

Khởi tạo mô hình dự đoán với thuật toán Cây quyết định với kỹ thuật lựa chọn thuộc tính Information gain tiêu chuẩn entropy và kỹ thuật Gini impurity tiêu chuẩn gini.

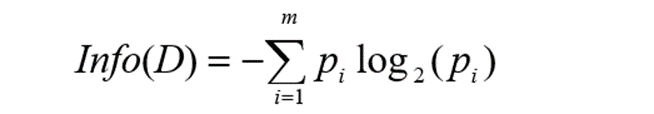
**Bước 1:** Huấn luyện với tập dữ liệu huấn luyện và dự đoán với tập dữ liệu kiểm thử từ đó cho ra được độ chính xác.

* Chia tập dữ liệu thành 70% training và 30% test.
* Dữ liệu ban đầu: 3276 hàng và 10 cột.
* Dữ liệu gốc sau khi qua xử lý: 3114 hàng và 10 cột.
* Chọn thuộc tính có mức thông tin cao

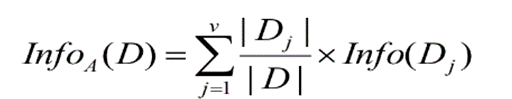


Gọi pi là xác suất để một bộ tùy ý trong D thuộc lớp Ci, được ước tính bởi |Ci,D|/|D|

Thông tin mong đợi (entropy) cần thiết để phân loại 1 bộ trong D:



Thông tin cần thiết (sau khi sử dụng A để chia D thành v phân vùng) để phân loại D:



Thông tin thu được bằng cách phân nhánh trên thuộc tính A



· Đặt A là thuộc tính có giá trị liên tục

· Xác định điểm phân tách tốt nhất cho A

· Sắp xếp các giá trị của thuộc tính A theo thứ tự tăng dần

· Điểm giữa mỗi cặp giá trị liền kề có thể coi là điểm phân tách (split point)

n (ai+ai+1)/2 là điểm giữa cặp (ai, ai+1)

· Điểm có thông tin mong đợi nhỏ nhất được gọi là điểm phân tách cho A

· Split – Phân tách:

D1 là tập hợp các bộ giá trị trong D thoả mãn A ≤ split-point, và D2 là tập hợp các bộ giá trị trong D thoả mãn A > split-point

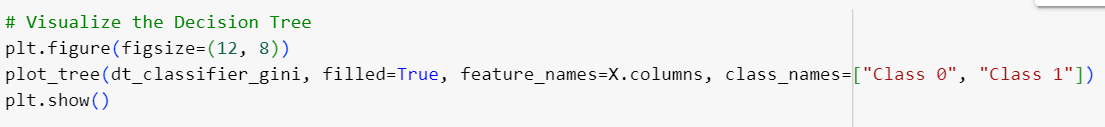
* **Thực hiện:**

Khi đã có dữ liệu train và dữ liệu test, ta tiến hành xây dựng cây quyết định bằng đoạn mã bên dưới, với độ sâu tối đa của cây = 3 cho ra được

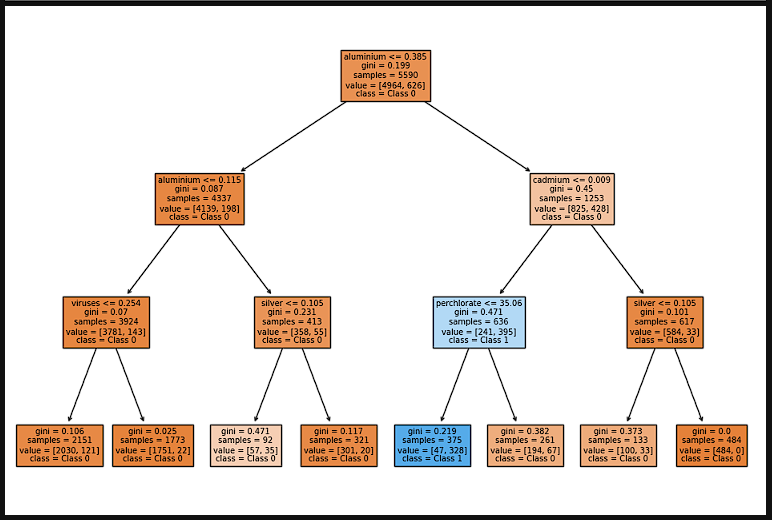
Độ chính xác của mô hình cây quyết định (Information gain) = 94,6%

Độ chính xác của mô hình cây quyết định (Gini impurity) = 93%

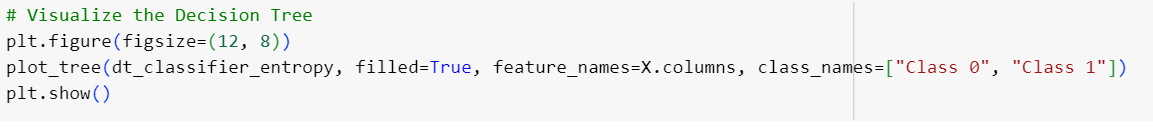
Sau khi chạy đoạn code này:



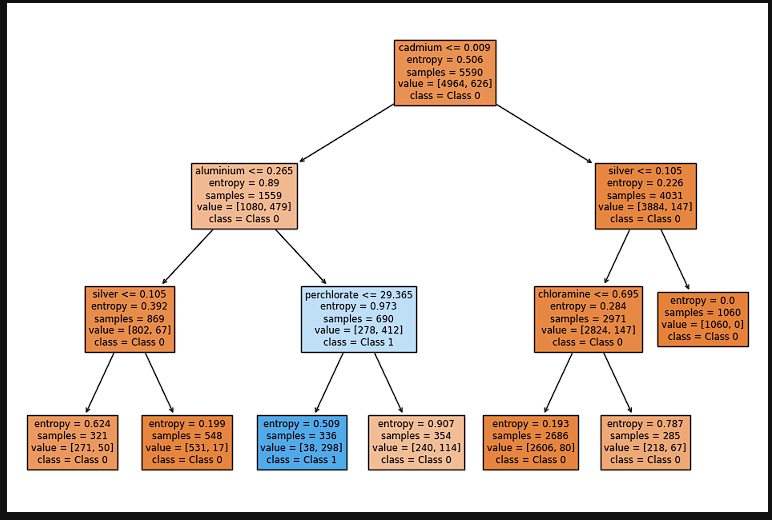
Hình ảnh cây quyết định (Information gain)



Sau khi chạy đoạn code này:

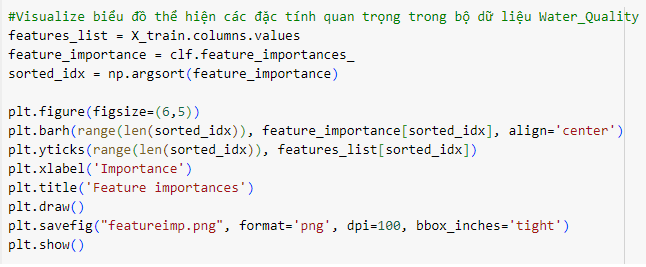


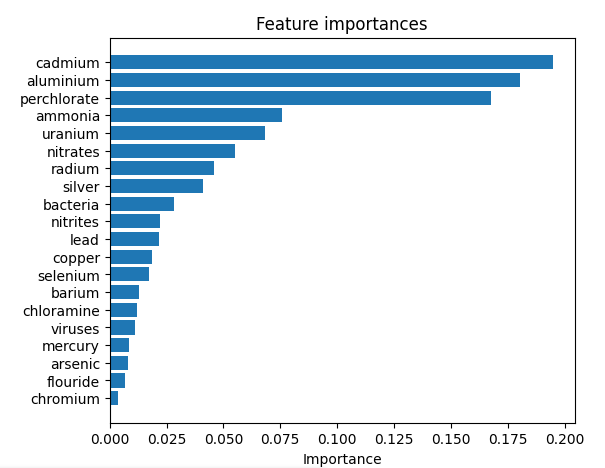
Hình ảnh cây quyết định (Gini impurity)



Nhận xét: Nhìn vào cây quyết định của bộ dữ liệu, ta thấy nút gốc của cây là “Cadmium”.

Ta tiến hành trích xuất các đặc trưng quan trọng để phân lớp thông qua đoạn mã sau:



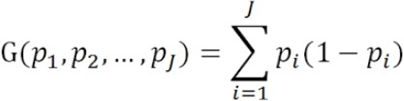


Tiếp theo sử dụng bộ phân loại cây quyết định và [Graphviz](http://www.graphviz.org/) để visualize mô hình cây quyết định và xác định nút gốc của cây

Cây phân loại dự đoán phản hồi định tính, ngược lại với cây hồi quy dự đoán phản ứng định lượng. Cây phân loại dự đoán quan sát thuộc về lớp xảy ra phổ biến nhất từ dữ liệu đào tạo cho từng khu vực. Để phát triển cây phân loại, thuật toán sử dụng phép tách nhị phân đệ quy.

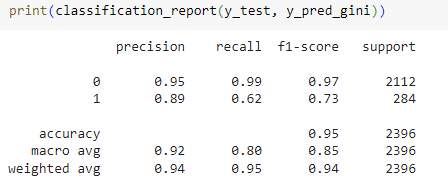
Chỉ số Gini là thước đo sự bất bình đẳng của một hệ thống có các giá trị từ 0 đến 1 (trong khi 0 là bình đẳng hoàn hảo và 1 là bất bình đẳng tuyệt đối). Nói cách khác, chỉ số Gini là một thước đo xác định độ tinh khiết của nút (giá trị 0 là tinh khiết và giá trị 1 không thuần khiết).

Chỉ số Gini được xác định bởi:



Trong đó: J là các lớp, pi là phần nhỏ của các mục được gắn nhãn với lớp i trong tập hợp.

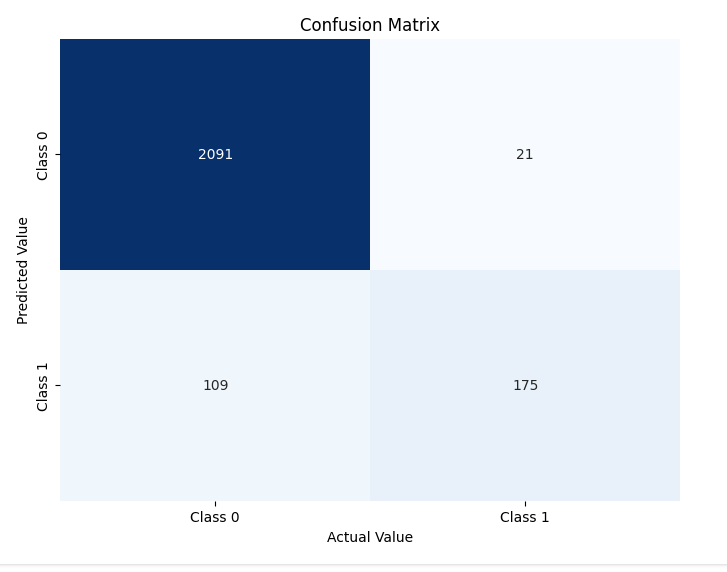
Kết quả:



Nhìn vào số liệu bên trên, ta thấy mô hình Decision Tree có độ phân loại trung bình khá trên tập dữ Water Quality với tổng điểm trung bình tầm khoảng từ 93%.

Dựa vào Confusion Matrix, ta rút được nhận xét:

* Số lượng dự đoán chính xác chất lượng nước đạt tiêu chuẩn (uống được) là: 175
* Số lượng dự đoán sai lệch chất lượng nước đạt tiêu chuẩn (uống được) là: 109
* Số lượng dự đoán chính xác chất lượng nước không đạt tiêu chuẩn (không uống được) là: 2091
* Số lượng dự đoán sai lệch chất lượng nước không đạt tiêu chuẩn (không uống được) là: 21
* Vậy tỉ lệ dự đoán số lượng chính xác của mô hình Decision Tree thuộc trung bình khá.



* 1. Thuật Toán Navi Bayes
     1. Giới thiệu:

Thuật toán Naive Bayes là một phương pháp học máy dựa trên nguyên tắc của định lý Bayes với giả định rằng các đặc trưng đầu vào độc lập với nhau khi đã biết lớp của dữ liệu. Mặc dù giả định này thường không đúng trong thực tế, thuật toán Naive Bayes thường làm việc hiệu quả và nhanh chóng trong nhiều ứng dụng thực tế.

Dưới đây là một số đặc điểm quan trọng của thuật toán Naive Bayes:

* Dựa trên định lý Bayes để tính xác suất của một sự kiện dựa trên xác suất của các sự kiện khác.
* Cụ thể, công thức Bayes cho phân loại nhóm (class) d dựa trên đặc trưng x là:



* Trong đó:
* P(classd​∣x) là xác suất của classd khi biết đặc trưng x.
* P(x∣classd​) là xác suất của đặc trưng x khi biết classd
* P(classd​) là xác suất của lớp classd
* P(x) là xác suất của đặc trưng x.

Có ba loại chính của thuật toán Naive Bayes:

* Gaussian Naive Bayes: Sử dụng khi đặc trưng đầu vào là liên tục và có phân phối Gaussian.
* Multinomial Naive Bayes: Phù hợp cho dữ liệu rời rạc (đa biến).
* Bernoulli Naive Bayes: Sử dụng cho dữ liệu nhị phân (0/1).

Thuật toán Naive Bayes là một trong những thuật toán đơn giản và hiệu quả, đặc biệt là đối với các bài toán phân loại với số lượng lớn các đặc trưng. Tuy nhiên, đối với dữ liệu mà giả định ngây thơ không được đáp ứng, có thể có sự giảm hiệu suất.

* + 1. Ưu điểm & nhược điểm của thuật toán
* **Ưu điểm:**

Naive Bayes là một thuật toán đơn giản, dễ triển khai và hiệu quả với dữ liệu lớn.

Hoạt động tốt khi dữ liệu phù hợp với giả định ngây thơ, đặc biệt là trong các tình huống với số lượng lớn các đặc trưng.

Multinomial Naive Bayes thích hợp cho dữ liệu rời rạc, như làm việc với văn bản và từ vựng.

Thường được sử dụng trong xử lý ngôn ngữ tự nhiên và lọc thư rác dựa trên văn bản.

Dễ mở rộng cho các bài toán với nhiều lớp và đặc trưng.

* **Nhược điểm:**

Giả định về sự độc lập giữa các đặc trưng không phản ánh đúng mối quan hệ thực tế trong một số tình huống.

Khó xử lý các đặc trưng liên tục và không phân phối đều.

Không xử lý tốt các tương quan giữa các đặc trưng, vì giả định về sự độc lập.

Nhạy cảm với dữ liệu nhiễu, có thể dẫn đến kết quả không chính xác.

Nếu dữ liệu có sự mất cân bằng giữa các lớp, có thể dẫn đến kết quả phân loại không chính xác.

Dễ bị ảnh hưởng khi có dữ liệu thiếu (missing data) hoặc các giá trị không xuất hiện trong quá trình huấn luyện.

* + 1. Ứng dụng

Naive Bayes thường được sử dụng trong các ứng dụng phân loại văn bản, lọc thư rác, phân loại tin tức, và các bài toán phân loại khác.

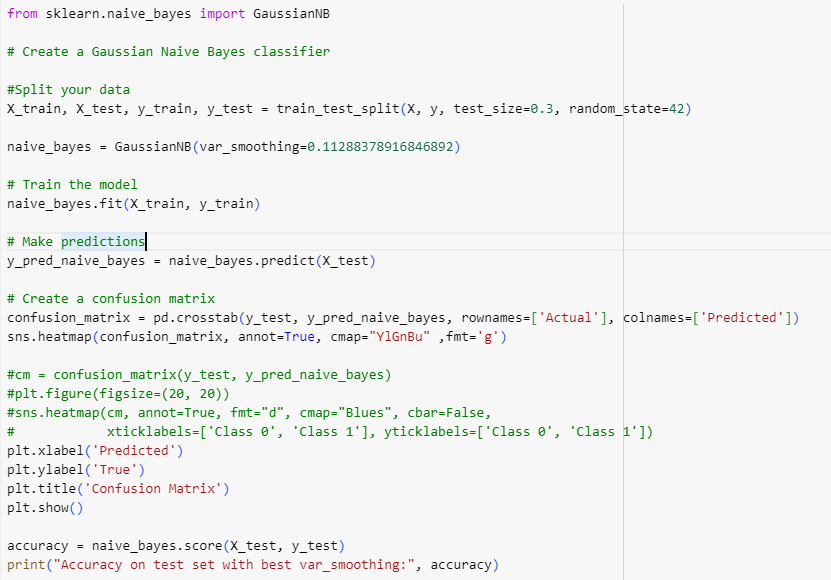
Naive Bayes có khả năng chống lại nhiễu tốt, tức là nó có thể hoạt động tốt trên dữ liệu mà có một số đặc trưng không quan trọng hoặc không liên quan.

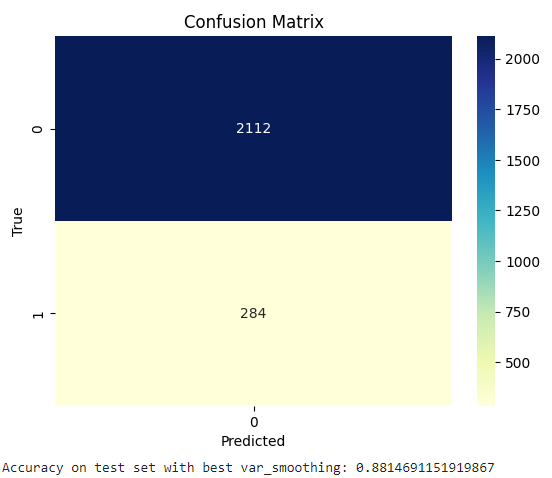
Naive Bayes thường hoạt động tốt trên dữ liệu lớn, và nó có thể được huấn luyện nhanh chóng.

Không Gặp Vấn Đề Overfitting Nếu Dữ Liệu Ít: Do giả định đơn giản và khả năng chống chịu nhiễu, Naive Bayes thường không gặp vấn đề overfitting nếu dữ liệu là ít. Tính Hiệu Quả: Mặc dù đơn giản, Naive Bayes thường rất hiệu quả trong nhiều tình huống thực tế, đặc biệt là khi dữ liệu lớn. Hiểu rõ về những yếu tố này sẽ giúp bạn áp dụng thuật toán Naive Bayes hiệu quả trong các bài toán phân loại của mình.

* + 1. Xử lý dữ liệu

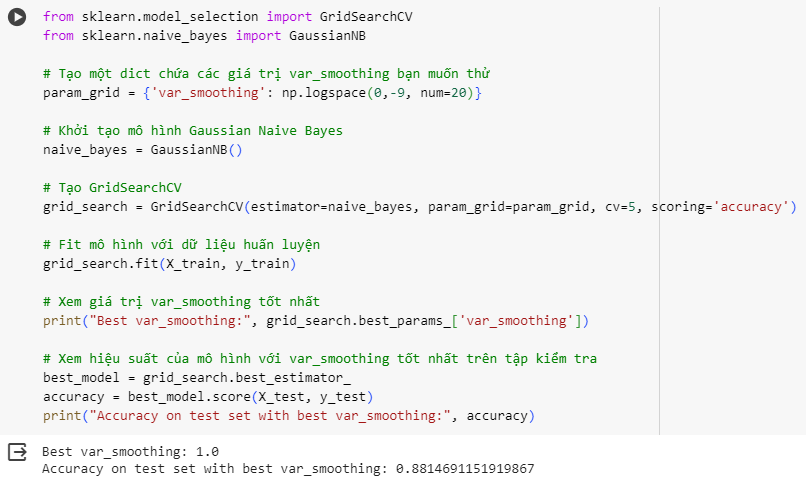
Dùng thư viên GaussianNB





Dựa vào Confusion Matrix, ta rút được nhận xét:

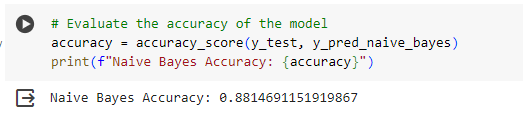
* Số lượng dự đoán chính xác chất lượng nước đạt tiêu chuẩn (uống được) là: 0
* Số lượng dự đoán sai lệch chất lượng nước đạt tiêu chuẩn (uống được) là: 284
* Số lượng dự đoán chính xác chất lượng nước không đạt tiêu chuẩn (không uống được) là: 2112
* Số lượng dự đoán sai lệch chất lượng nước không đạt tiêu chuẩn (không uống được) là: 0
* Vậy tỉ lệ dự đoán số lượng chính xác của mô hình Decision Tree thuộc trung bình khá 88.1%



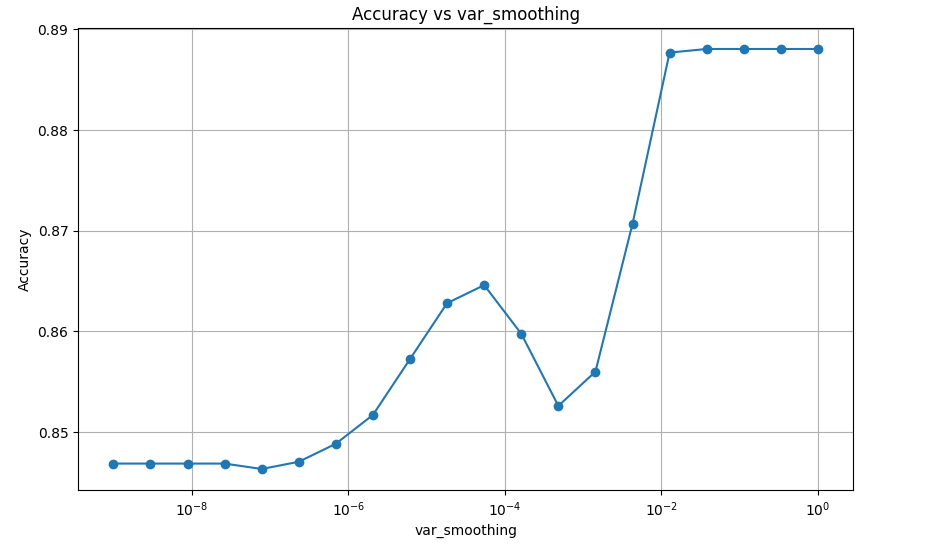
Sau khi thử lại tạo ra giá trị var-smoothing thì có độ best là 1.0. Cái 1.0 đó là best var smoothing đang tìm để accuracy tốt nhất.

Nhưng mà bài nhóm em var smoothing = 1 thì accuracy cao nhưng nhìn vào confusion matrix nó không tốt nên để var moothing default.

Và test lại

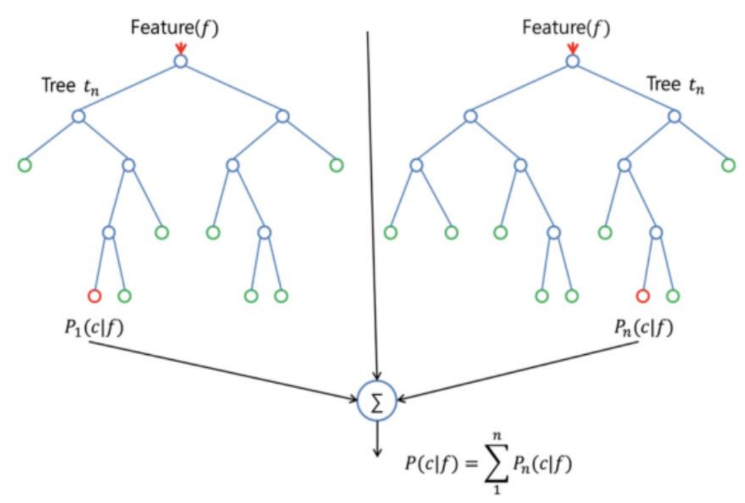


Vẽ mô hình



* 1. Thuật toán Random Forest
     1. Giới thiệu

Rừng ngẫu nhiên (Random Forest) là một thuật toán học máy linh hoạt và dễ sử dụng, luôn tạo ra kết quả xuất sắc, ngay cả khi không cần điều chỉnh hyper. Nó cũng là một trong những thuật toán được sử dụng rộng rãi nhất, do tính đơn giản và thực tế là nó có thể được sử dụng cho các nhiệm vụ phân loại và hồi quy. Nó tạo ra một khu rừng một cách ngẫu nhiên. “Khu rừng” tạo ra là sự kết hợp (quần thể) của các cây quyết định, trong hầu hết các trường hợp được đào tạo bằng phương pháp đóng bao. Ý tưởng chính của phương pháp đóng gói là sự kết hợp của các mô hình học tập làm tăng kết quả chung.



* + 1. Ưu điểm & nhược điểm
* **Ưu điểm:**

Một ưu điểm chính của thuật toán rừng ngẫu nhiên là nó có thể được sử dụng cho cả nhiệm vụ phân loại và hồi quy, đại diện cho hầu hết các hệ thống học máy hiện nay.

Đây là thuật toán dễ tiếp cận, số siêu tham số không lớn và dễ hiểu.

RandomForest có khả năng xử lý tốt trên dữ liệu lớn và có thể xử lý hàng nghìn biến độc lập mà không cần quá nhiều tinh chỉnh.

RandomForest thích hợp cho cả các bài toán phân loại lẫn dự đoán, giúp giảm sự phụ thuộc vào loại mô hình.

RandomForest cung cấp thông tin về độ quan trọng của từng biến, giúp phân tích đặc trưng và hiểu rõ hơn về quy luật trong dữ liệu

RandomForest có khả năng chống overfitting tốt hơn so với một số mô hình phức tạp khác.

RandomForest có khả năng tốt khi làm việc với dữ liệu có giá trị bị thiếu một cách hiệu quả

RandomForest không đòi hỏi nhiều cấu hình và thường độc lập với việc chuẩn hóa dữ liệu.

* **Nhược điểm:**

Số lượng cây lớn làm cho thuật toán chậm và kém hiệu quả đối với các dự đoán ngoài thời gian thực. Là một công cụ mô hình dự đoán và không có mô tả.

Khi có một lượng lớn các biến quan trọng hơn, RandomForest có thể trở nên ít hiệu quả.

Mô hình của RandomForest có thể trở nên lớn khi số cây (n\_estimators) tăng, điều này có thể làm tăng thời gian huấn luyện và không gian lưu trữ.

RandomForest có thể khó giải thích so với một số mô hình đơn giản hơn, nhất là khi có nhiều cây trong mô hình.

Khi số cây tăng lên, thời gian huấn luyện và khả năng tiêu tốn bộ nhớ cũng tăng lên.

RandomForest có thể bị ảnh hưởng bởi dữ liệu nhiễu và biến động, đặc biệt là khi số cây ít.

* + 1. Ứng dụng

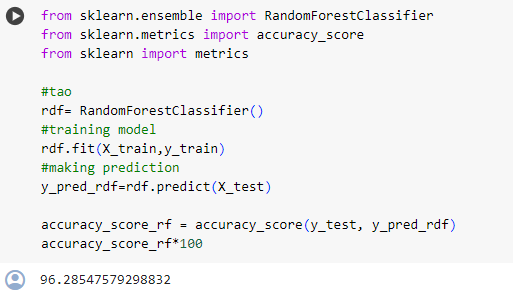
Thuật toán Random Forest được sử dụng trong nhiều lĩnh vực khác nhau, chẳng hạn như ngân hàng, thị trường tài chính, y học, thương mại điện tử.

Ví dụ trong lĩnh vực ngân hàng, nó được sử dụng để phát hiện những khách hàng sẽ sử dụng dịch vụ ngân hàng thường xuyên hơn những người khác và trả nợ đúng hạn. Trong lĩnh vực này, nó cũng được sử dụng để phát hiện hành vi gian lận của những khách hàng muốn gây hại cho ngân hàng.

Trong lĩnh vực tài chính, nó được sử dụng để xác định hiệu suất trong tương lai của một cổ phiếu. Trong chăm sóc sức khỏe, nó được sử dụng để xác định sự kết hợp chính xác của các thành phần trong y học, và nó cũng được sử dụng để phân tích bệnh sử của bệnh nhân để xác định bệnh. Cuối cùng, trong thương mại điện tử, Forest Random được sử dụng để xác định liệu khách hàng có thích sản phẩm hay không.

* + 1. Thực hiện

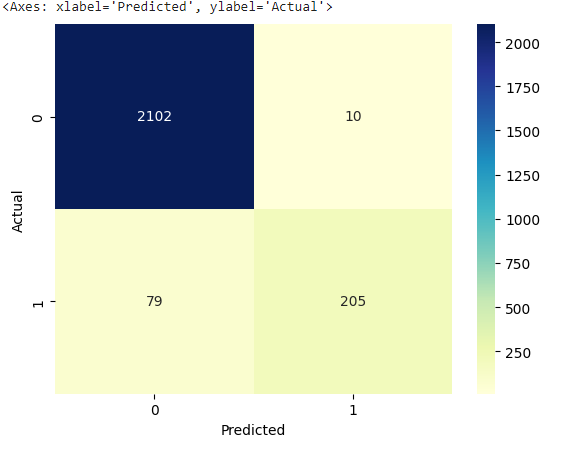
Ta tiến hành xây dựng model Random Forest bằng thư viện Scikit learn để dự đoán và ước tính kết quả:



*Nhận xét:*

Nhìn vào số liệu bên trên, ta thấy mô hình RandomForestClassifier có độ phân loại khá tốt trên tập dữ liệu Water Quality với tổng điểm trung bình: 96.28 %

Còn đối với mô hình Confusion Matrix



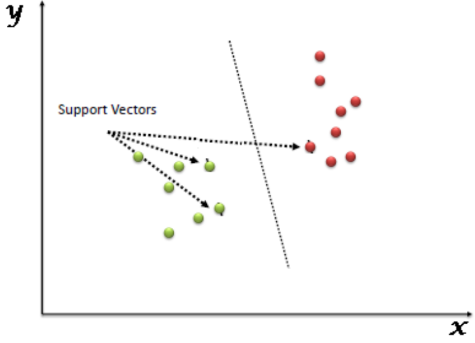
Dựa vào Confusion Matrix, ta có một số nhận xét:

* Số lượng dự đoán chính xác nước đạt tiêu chuẩn (uống được) là 205
* Số lượng dự đoán sai lệch nước đạt tiêu chuẩn (uống được) là 10
* Số lượng dự đoán chính xác nước không đạt tiêu chuẩn (không uống được) là 2102
* Số lượng dự đoán sai lệch nước không đạt tiêu chuẩn (không uống được) là 79
  1. Thuật toán Support Vector Machines
     1. Giới thiệu

Support Vector Machines là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại.

Trong thuật toán này, chúng ta vẽ đồi thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết.

Sau đó chúng ta thực hiện tìm "đường bay" (hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.



* + 1. Ưu điểm & nhược điểm
* **Ưu điểm:**

 Xử lý trên không gian số chiều cao, ví dụ: áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi có nhiều chiều dữ liệu

Tiết kiệm bộ nhớ, chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.

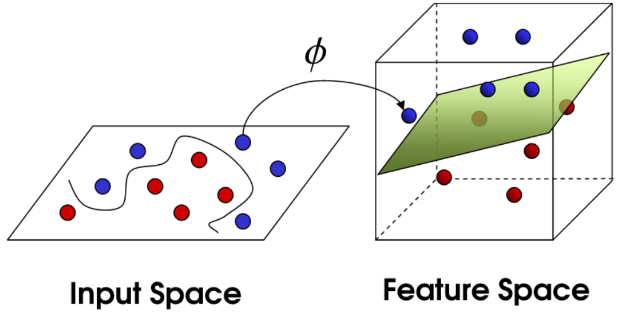
Tính linh hoạt – phân lớp thường là phi tuyến.

* **Nhược điểm:**

Bài toán số chiều cao: Trong trường hợp số lượng thuộc tính (p) của tập dữ liệu lớn hơn rất nhiều so với số lượng dữ liệu (n) thì SVM cho kết quả khá tồi.

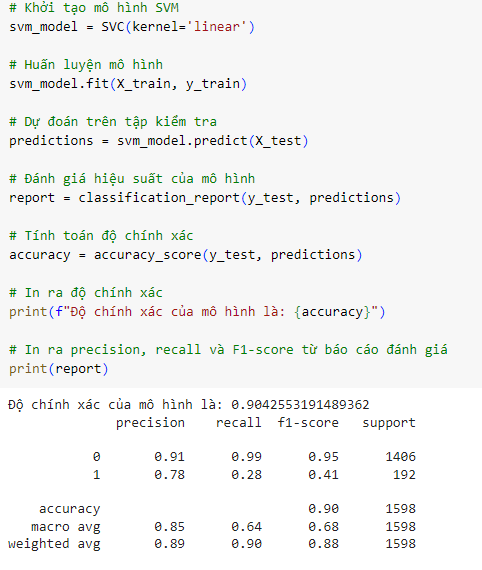
Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào.

*Kết luận:* SVM là một phương pháp hiệu quả cho bài toán phân lớp dữ liệu. Nó là một công cụ đắc lực cho các bài toán về xử lý ảnh, phân loại văn bản, phân tích quan điểm. Một yếu tố làm nên hiệu quả của SVM đó là việc sử dụng Kernel function khiến cho các phương pháp chuyển không gian trở nên linh hoạt hơn.

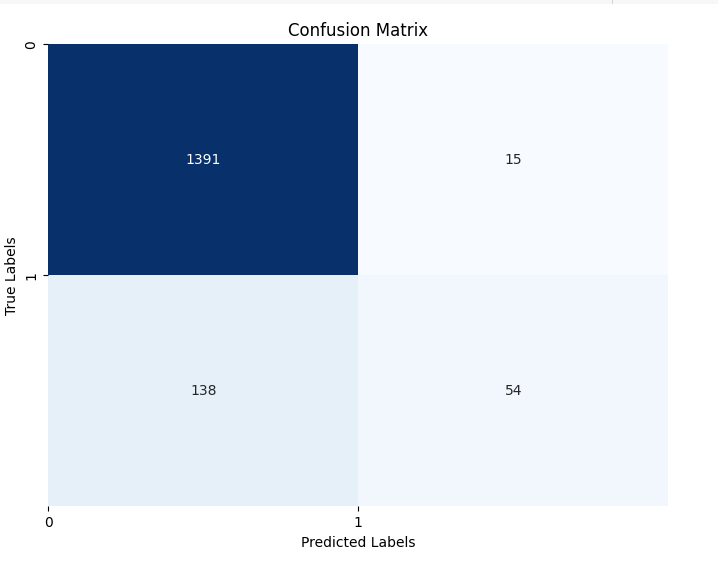
**

* + 1. Thực hiện

Ta tiến hành xây dựng model Support Vector Machines bằng thư viện Scikit learn để dự đoán và ước tính kết quả:



Nhìn vào số liệu bên trên, ta thấy mô hình Support Vector Machines có độ phân loại tốt trên tập dữ liệu Water Quality với tổng điểm trung bình 90%.



Dựa vào Confusion Matrix, ta có một số nhận xét:

* Số lượng dự đoán chính xác nước đạt tiêu chuẩn (uống được) là 54.
* Số lượng dự đoán sai lệch nước đạt tiêu chuẩn (uống được) là 138.
* Số lượng dự đoán chính xác nước không đạt tiêu chuẩn (không uống được) là 1391.
* Số lượng dự đoán sai lệch nước không đạt tiêu chuẩn (không uống được) là 15.

# **PHẦN IV. ĐÁNH GIÁ THUẬT TOÁN**

1. Đánh giá

Nhóm chúng em đã lựa chọn phương pháp Xác thực chéo (Cross-validation) để ước tính độ chính xác và lựa chọn mô hình ROC để lựa chọn mô hình của trình phân loại

**Xác thực chéo:**

* Phân vùng dữ liệu ngẫu nhiên thành k tập con loại trừ lẫn nhau, mỗi tập có kích thước xấp xỉ bằng nhau
* Ở lần lặp thứ i, sử dụng Di làm bộ kiểm tra và các bộ khác làm bộ huấn luyện
* Bỏ lại 1 lần: k lần gấp, k = số bộ dữ liệu, đối với dữ liệu có kích thước nhỏ
* Xác nhận chéo phân tầng: Các nếp gấp được phân tầng để phân lớp, mỗi nếp gấp là khoảng giống như trong dữ liệu ban đầu

**Mô hình ROC** (Receiver Operating Characteristics): để so sánh trực quan các mô hình phân loại

* Bắt nguồn từ lý thuyết phát hiện tín hiệu
* Cho thấy sự cân bằng giữa tỷ lệ dương tính thực sự và tỷ lệ dương tính giả
* Diện tích dưới đường cong ROC là thước đo độ chính xác của mô hình
* Xếp hạng các bộ giá trị thử nghiệm theo thứ tự giảm dần: bộ giá trị có nhiều khả năng thuộc về lớp tích cực nhất xuất hiện ở đầu danh sách
* Càng gần đường chéo (tức là vùng càng gần 0,5), mô hình càng kém chính xác
* Trục dọc biểu thị tỷ lệ dương thực sự
* Đại diện trục ngang biểu thị tỷ lệ dương tính giả
* Biểu đồ cũng cho thấy một đường chéo
* Một mô hình với độ chính xác hoàn hảo sẽ có diện tích là 1,0

2. Kết luận

Kết luận 4 độ chính xác thì ra được Thuật toán RandomForgrest là tốt nhất

# 

# **PHẦN V. TỔNG QUAN VỀ CÁC THUẬT TOÁN SỦ DỤNG**

1.Kết luận

Phân lớp và phân cụm dữ liệu là lĩnh vực khá quan trọng trong khai phá dữ liệu nhằm đưa ra những dự đoán, xu hướng trong tương lai, nó được ứng dụng trong nhiều ngành như y tế, thương mại, ngân hàng, giáo dục, nông nghiệp ... Trong quá trình tìm hiểu và hoàn thành bài tập lớn với đề tài “Water Quality”. Nhóm em đã đạt được một số kết quả như sau:

* Tìm hiểu tổng quan về khai phá dữ liệu, bài toán phân lớp, qua các mô hình hình phân lớp như Decision Tree, Random Forest, Support Vector Machine. Hỗ trợ phân loại nước sạch hay không sạch.
* Thu thập dữ liệu trên Kaggle, tiền xử lý dữ liệu bằng thư viện pandas trong python. Xây dựng nên mô hình phân lớp và phân cụm trên Google Collaboration ...
* Đánh giá các mô hình phân lớp và phân cụm khác nhau.
* Tìm hiểu về ngôn ngữ Python các các thư viện hỗ trợ như Pandas, Matplotlib, Sci-learn.

***Hạn chế:***

* Thời gian triển khai ngắn, nên chúng em còn nhiều sơ sót và kiến thức của chúng em về mặt toán học và khả năng lập trình còn hạn chế.

2.Hướng phát triển

Xây dựng, cải tiến mô hình chẩn đoán bệnh với phương pháp học máy khác như CNN,...Tích hợp các mô hình học máy hiện đại, nhằm dự đoán qua hình ảnh, qua video realtime, để chế tạo máy phân loại, giúp người nông dân có thể loại bỏ các nguồn nước “bẩn” trong quá trình sinh hoạt.

Trong quá trình hoàn thành bài tập lớn, chúng em đã cố gắng tìm hiểu và tham khảo các tài liệu liên quan. Tuy nhiên, thời gian có hạn nên chúng em sẽ không tránh khỏi những thiếu sót, rất mong nhận được sự đóng góp ý kiến của quý thầy cô và các bạn để báo cáo và kỹ năng của chúng em ngày được hoàn thiện hơn và có thế áp dụng trong thực tiễn.

1. Tài Liệu Tham Khảo

[1] Th.S Vũ Thị Hạnh, (2022), Slide Bài giảng Học Máy.

[2] <https://phamdinhkhanh.github.io/deepai-book/ch_appendix/appendix_pandas.html>

[3] <http://uet.vnu.edu.vn/~thuyhq/Student_Thesis/K55_Pham_Huy_Thanh_Thesis.pdf>

[4] <https://www.youtube.com/watch?v=JpcbV5Kl4nw>

[5] <https://www.youtube.com/watch?v=V-RpBGmqHM8>

[6] <https://www.youtube.com/watch?v=lX-3nGHDhQg&t=1739s>

[7] <https://www.youtube.com/watch?v=vOyrfgQDJ4A&t=1040s>

\*Source code nhóm 5:

<https://github.com/nguyenhoangthanhly/Nhom5_WaterQuality>