**Report:** **Water Quality**

**Thành viên nhóm 8:**

1. Nguyễn Anh Đắc 19133020
2. Nguyễn Thanh Tân Kỷ 19133031
3. Lại Hữu Trác 19133059
4. Đào Thị Cẩm Tiên 19133055
5. Tóm tắt (Abstract)

Tiếp cận nước uống an toàn là điều cần thiết đối với sức khỏe, một quyền cơ bản của con người và là một thành phần của chính sách hiệu quả để bảo vệ sức khỏe.

Đây là vấn đề quan trọng như một vấn đề sức khỏe và phát triển ở cấp quốc gia, khu vực và địa phương.

Ở một số vùng, người ta đã chỉ ra rằng các khoản đầu tư vào cấp nước và vệ sinh có thể mang lại lợi ích kinh tế ròng, vì việc giảm các tác động xấu đến sức khỏe và chi phí chăm sóc sức khoẻ lớn hơn chi phí thực hiện các can thiệp.

Vì vậy, nhóm em đã áp dụng các thuật toán Machine Learning như: Random Forest, KNN, Logistic Regression vào để dự đoán xem nước uống được và nước không uống được.

* 1. Câu hỏi đặt ra cho bài toán

Dự đoán chất lượng nước (1: an toàn, 0: không an toàn)

* 1. Kết quả

Thuật toán Random forest có kết quả tốt nhất

1. Giới thiệu (Introdution)

Cải thiện việc cung cấp nước và vấn đề vệ sinh cũng như quản lý tài nguyên nước tốt hơn, có thể thúc đẩy tăng trưởng kinh tế của các quốc gia và có thể đóng góp rất nhiều vào việc xóa đói giảm nghèo.

Nước bị ô nhiễm và điều kiện vệ sinh kém có liên quan đến việc lây truyền các bệnh như tả, tiêu chảy, kiết lỵ, viêm gan A, thương hàn và bại liệt.

Vì vậy, nhóm em đã tìm kiếm và sử dụng bộ dữ liệu Water Quality này để hiểu những yếu tố cấu thành nên nước an toàn, nước uống được và áp dụng thuật toán Machine Learning vào để dự đoán xem nước uống được và nước không uống được

* 1. Input

Tập dữ liệu: <https://www.kaggle.com/datasets/adityakadiwal/water-potability>

Gồm các biến: pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic\_carbon, Trihalomethanes, Turbidity.

* 1. Output

**Output:** Potability (1: an toàn, 0: không an toàn)

Chia tập train, tập test:

Trên tập train sẽ thử các thuật toán:

* KNN
* Random Forest
* Logistic Regression

Đánh giá mô hình trên tập test.

1. Dữ liệu (Data)
   1. Giới thiệu về tập dữ liệu

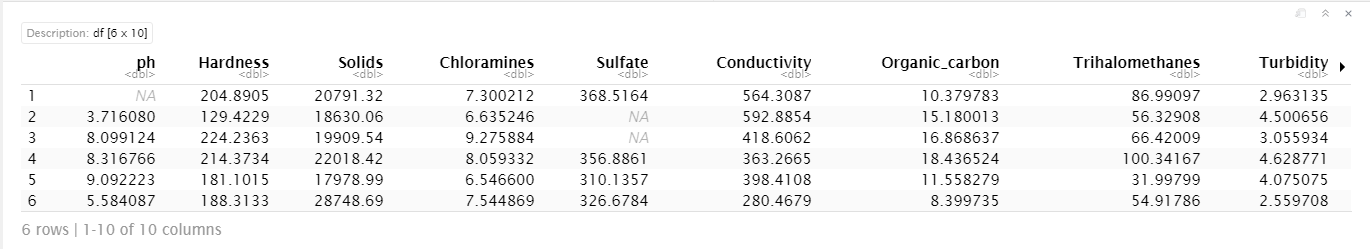
Tập dữ liệu được lấy từ Kaggle.

Link tập dữ liệu: <https://www.kaggle.com/datasets/adityakadiwal/water-potability>

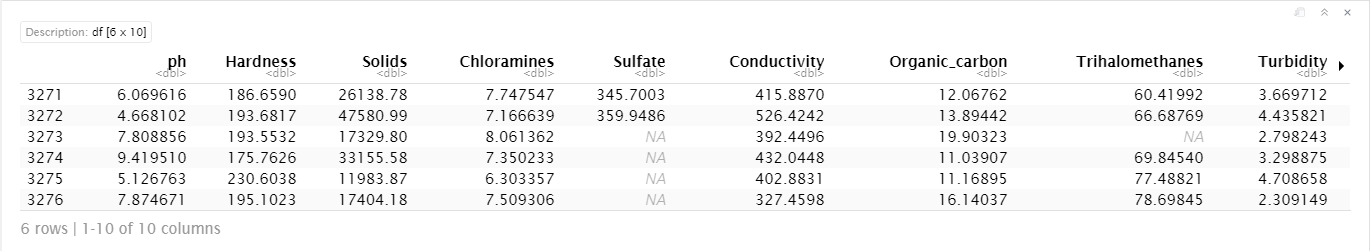
Dữ liệu chứa các chỉ số chất lượng nước, gồm 10 cột và 3276 dòng

Các cột trong tập dữ liệu:

* **pH value:** Độ pH của nước. WHO đã khuyến cáo giới hạn pH tối đa cho phép từ 6,5 đến 8,5
* **Hardness(mg/l):** Độ cứng của nước. Độ cứng của nước được định nghĩa đơn giản nhất là loại nước có tổng lượng muối Ca và Mg được hòa tan trong nước vượt qua mức cho phép
* **Solids (Total dissolved solids - TDS):** Tổng chất rắn hoà tan trong nước. giới hạn mong muốn cho TDS là 500 mg / l và giới hạn tối đa là 1000 mg / l được quy định cho mục đích uống
* **Chloramines:** Mức clo lên đến 4 miligam mỗi lít (mg / L hoặc 4 phần triệu (ppm)) được coi là an toàn trong nước uống
* **Sulfate:** lượng sulfate trong nước
* **Conductivity:** tính dẫn điện.
* **Organic\_carbon:** lượng cacbon hữu cơ trong nước
* **Trihalomethanes:** lượng THM trong nước. Đây là sản phẩm phụ của quá trình khử trùng nước bằng Chlorine.
* **Turbidity:** Độ đục của nước
* **Potability:** Cho biết liệu nước có an toàn cho con người hay không, trong đó 1 nghĩa là Uống được và 0 nghĩa là Không uống được
* 6 dòng đầu của tập dữ liệu

****

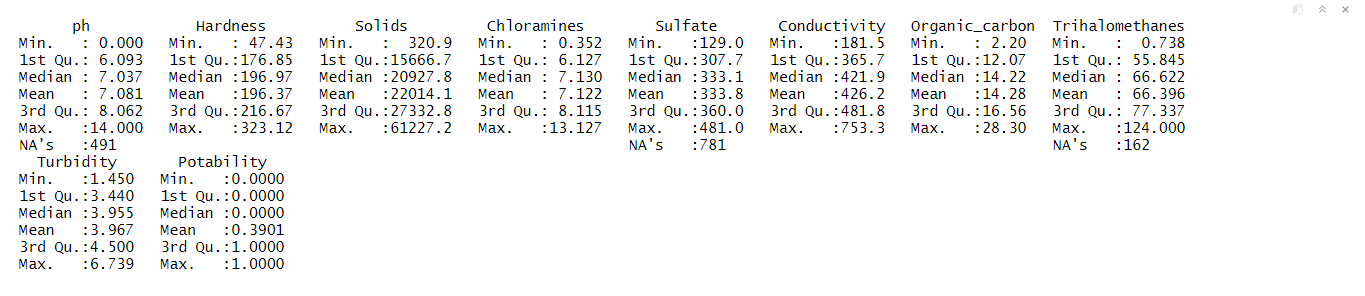
* 6 dòng cuối của tập dữ liệu

****

* Cấu trúc dữ liệu của các biến

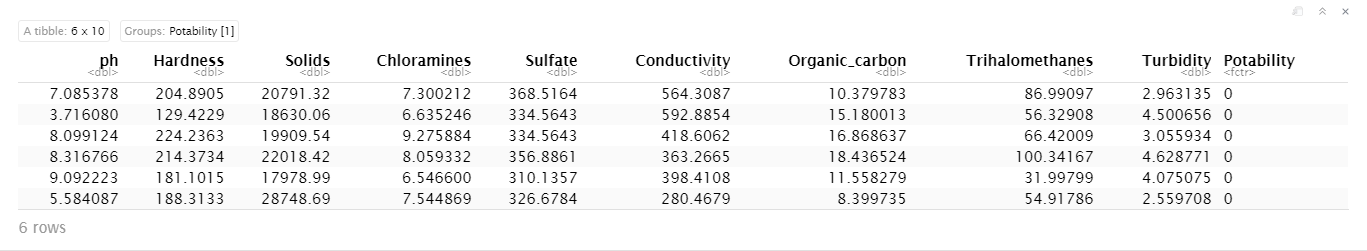
****

* Tóm tắt thống kê tập dữ liệu

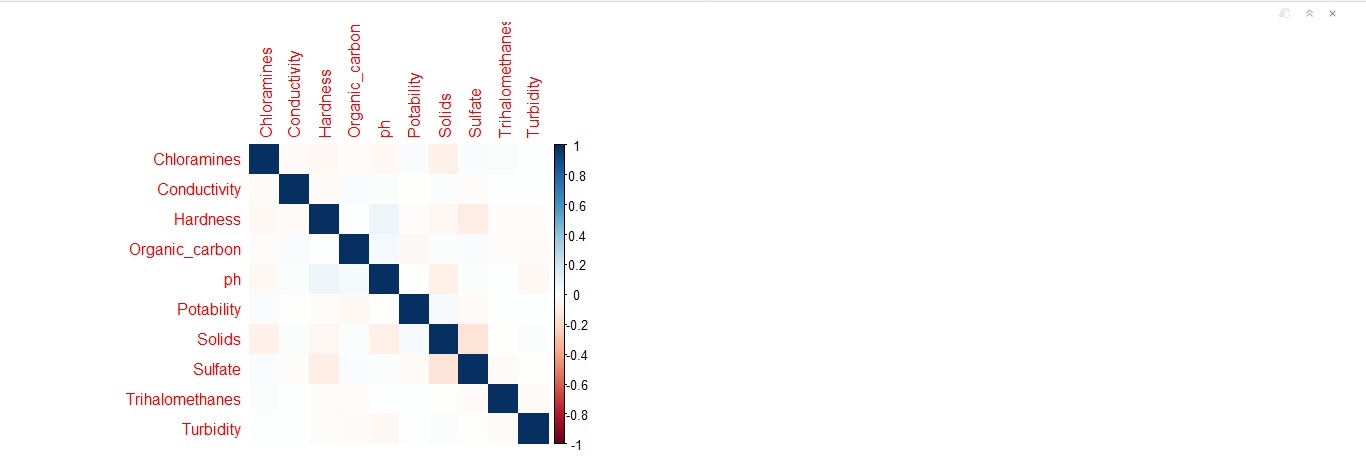
****

* 1. Tiền xử lý dữ liệu
* Xử lý các giá trị NA bằng cách thay chúng bằng giá trị mean





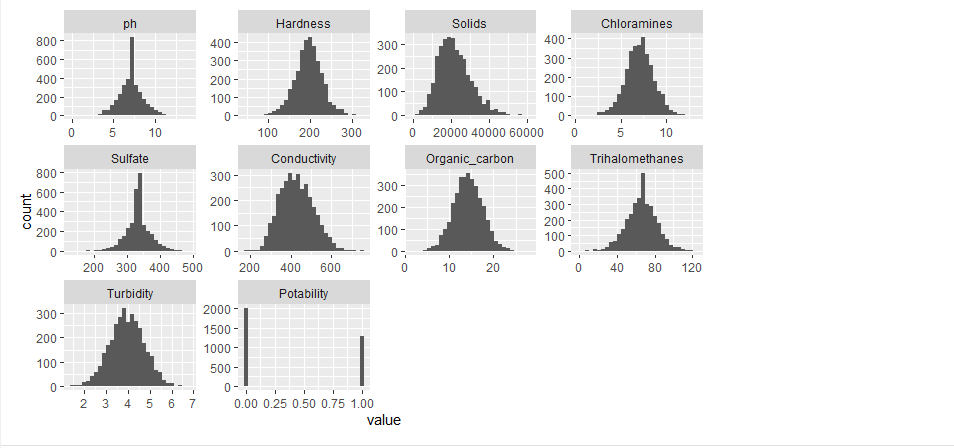
* 1. EDA
* Biểu đồ thể hiện sự tương quan giữa các biến trong tập dữ liệu:



Dựa vào biểu đồ corrplot ở trên ta có thể thấy sự tương quan giữa các biển với nhau:

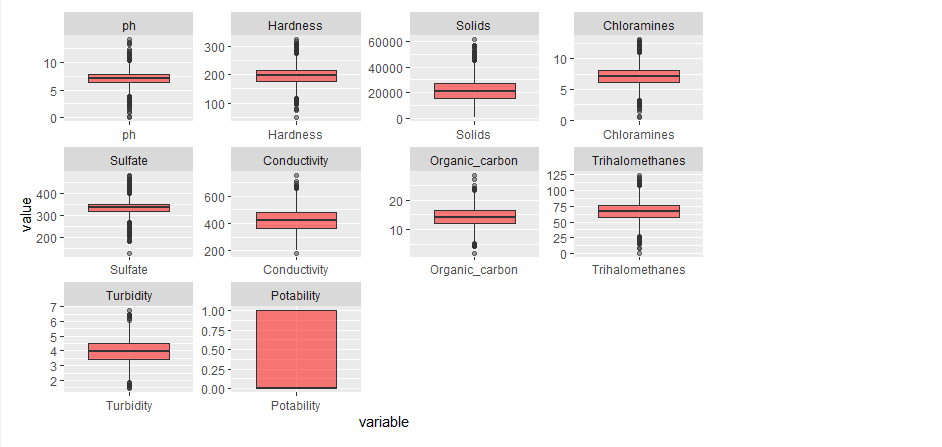
Mức độ tương quan giữa các biến cũng không quá rõ rệt, tiêu biểu ở 1 số biển như:

* Hardness và ph có mức độ tương quan khá tích cực khoảng 0.2
* Solids và sulfate có mực độ tương quan tiêu cực khoảng -0.2
* Biểu đồ thể hiện sự phân bố của các biến trong tập dữ liệu



Với biểu đồ histogram ở trên chúng ta có thể thấy sự phân bố số liệu của các biển có trong tập dữ liệu:

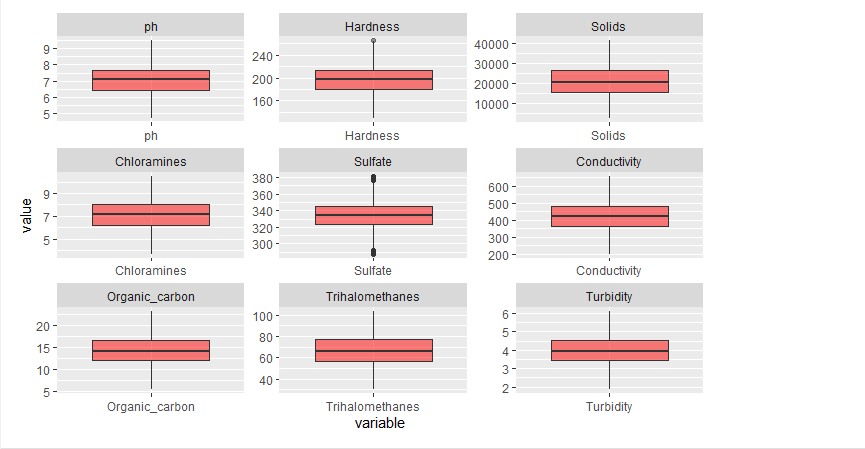
* Với biểu đồ của 3 biến ph, sulfate, Trihalomethanes chúng ta có thể thấy có sự phân bố không đồng đều, đây là hậu quả của việc quá nhiều giá trị NA và sau khi ta thấy giấ trị mean vào.



- Với biểu đồ boxplot thì chúng ta có thể thấy sự xuất hiện của các outlier của các biển và nó khá nhiều. Nó có thể ảnh hưởng đến kết quả training của 1 số model nên nhóm sẽ quyết định xử lý các outlier này.

* 1. Xử lý các giá trị outlier

Biểu đô boxplot ở trên là biểu đồ sau khi nhóm đã thực hiện xử lý các outliers. Nhưng vẫn còn sót lại 1 số outliers ở biến sulfate. Có thể là do ở biến này các outliers xuất hiện quá nhiều nên thành ra không thể xóa hết được, hoặc là mình sử dụng phép tính chưa tối ưu, nhưng chắc sẽ không ảnh hưởng quá nhiều đến kết quả training model.



* 1. Chia tập dữ liệu

Chia tập train, tập test: Training (75%), Testing (25%)

Phương pháp k-fold Cross-Validation: Nhóm em sẽ chia dữ liệu tập training thành 5 phần, chọn một phần trong 5 phần đó làm tập test, và thực hiện train các phần còn lại, lặp lại việc chọn tập test như vậy cho đến khi đủ 5 lần.

Trên tập train sẽ thử các thuật toán:

* KNN
* Random Forest
* Logistic Regression

Đánh giá mô hình trên tập test.

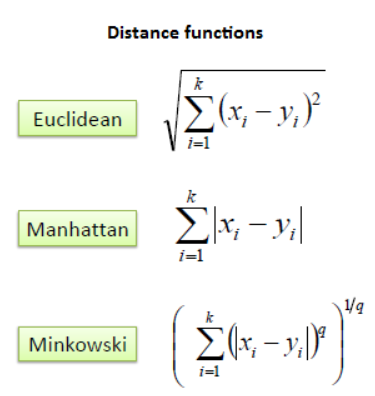
1. Phương pháp (Method)
   1. KNN

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.

Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

* Các bước trong mô hình KNN:
* Ta có D là tập các điểm dữ liệu đã được gắn nhãn và A là dữ liệu chưa được phân loại.
* Đo khoảng cách (Euclidian, Manhattan, Minkowski, Minkowski hoặc Trọng số) từ dữ liệu mới A đến tất cả các dữ liệu khác đã được phân loại trong D.

Công thức tính đo khoảng cách:



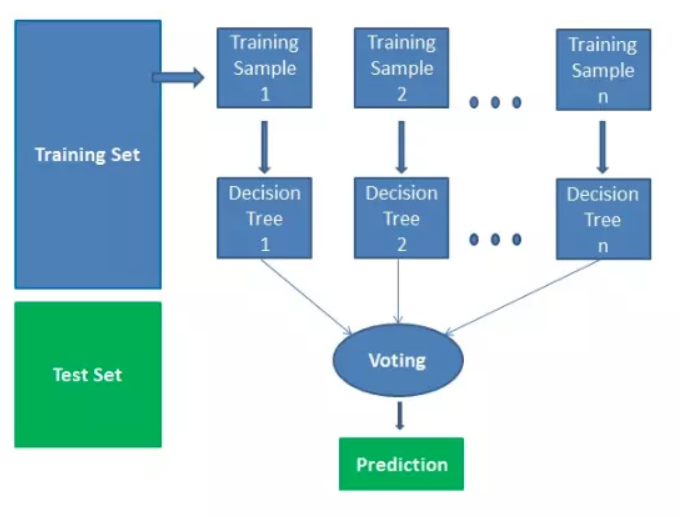
* Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất.
* Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.
* Lấy đúng lớp (lớp xuất hiện nhiều lần nhất).
* Lớp của dữ liệu mới là lớp mà bạn đã nhận được ở bước
  1. Random Forest

Random forest là thuật toán supervised learning, có thể giải quyết cả bài toán regression và classification.

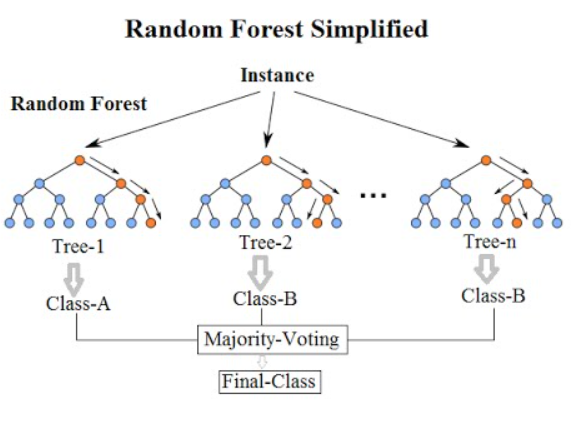
Random là ngẫu nhiên, Forest là rừng, nên ở thuật toán Random Forest mình sẽ xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên mỗi cây quyết định sẽ khác nhau (có yếu tố random). Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định.

* Hoạt động:

1. Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.
2. Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi quyết định cây.
3. Hãy bỏ phiếu cho mỗi kết quả dự đoán.
4. Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng.



Đơn vị của RF là thuật toán cây quyết định, với số lượng hàng trăm. Mỗi cây quyết định được tạo ra một cách ngẫu nhiên từ việc : Tái chọn mẫu (bootstrap, random sampling) và chỉ dùng một phần nhỏ tập biến ngẫu nhiên (random features) từ toàn bộ các biến trong dữ liệu. Ở trạng thái sau cùng, mô hình RF thường hoạt động rất chính xác, nhưng đổi lại, ta không thể nào hiểu được cơ chế hoạt động bên trong mô hình vì cấu trúc quá phức tạp. RF do đó là một trong số những mô hình hộp đen (black box).



* **Confusion matrix**

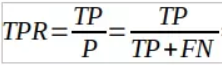
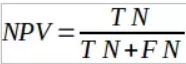
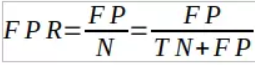
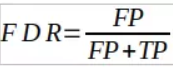
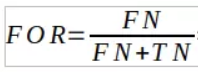
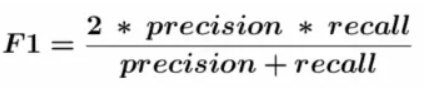
Confusion Matrix ma trận nhầm lẫn hay ma trận lỗi là một bố cục bảng cụ thể cho phép hình dung hiệu suất của một thuật toán.

Ma trận nhầm lẫn là một trong những kỹ thuật đo lường hiệu suất phổ biến nhất và được sử dụng rộng rãi cho các mô hình phân loại.

Từ ma trận cơ bản này, ta sẽ có một số thuật ngữ sau:

* Condition positive (P): Tổng số ca dương tính thực tế.
* Condition Negative (N: Tổng số ca âm tính thực tế.
* True positive (TP): Số các ca dự đoán dương tính đúng hay dương tính thật.
* True negative (TN): Số các ca dự đoán âm tính đúng hay âm tính thật.
* False positive (FP): Số các ca dự đoán dương tính sai hay dương tính giả.
* False negative (FN):: Số các ca dự đoán âm tính sai hay âm tính giả.

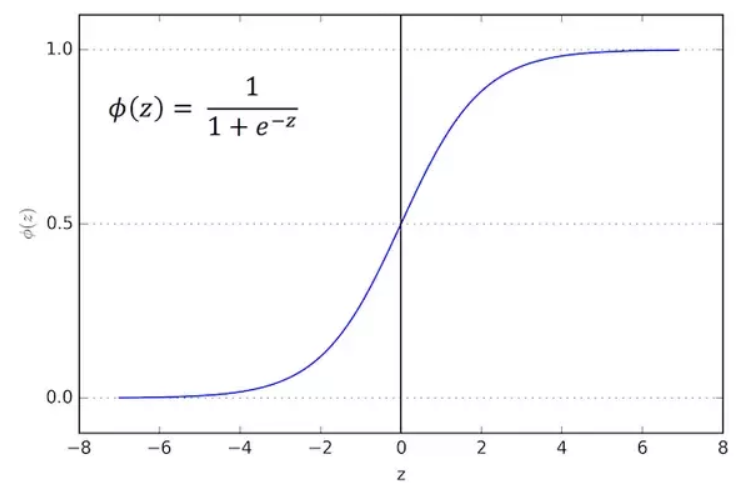
Với các thuật ngữ trên, ta có các chỉ số đánh giá sau:

* **Độ chính xác** – **Accuracy**:  
  
* Sensitivity, **Recall**, Hit Rate, Or True Positive Rate (TPR): **Độ nhạy - Tỷ lệ dương tính thực**:  
  
* **Precision** Or Positive Predictive Value (PPV): **Tỉ lệ dương tính đoán đúng**  
  
* **NegativePredictive Value**  (NPV): **Tỉ lê âm tính đoán đúng**  
  
* **Miss Rate** Or False Negative Rate (FNR): **Tỉ lệ dương tính giả**  
  
* **Fall-Out**  Or  False Positive Rate (FPR): Tỉ lệ âm tính giả  
  
* False Discovery Rate(FDR): **Tỉ lệ đoán dương tính sai**  
  
* False Omission Rate (FOR): **Tỉ lệ đoán âm tính sai**  
  
* **F1 score - Điểm F1**: Điểm F1 là một trung bình hài hòa **Precision** và **Recall**.  
  
  1. Logistic Regression

**Logistic regression** được áp dụng trong bài toán phân loại nhị phân (Binary classification) tức ta sẽ có hai output, hoặc có thể gọi là hai nhãn (ví dụ như 0 và 1 hoặc cat và non-cat).

* Hàm sigmoid:

Hàm luôn có giá trị trong đoạn [0, 1], liên tục mà lại dễ sử dụng thì đó là hàm sigmoid.



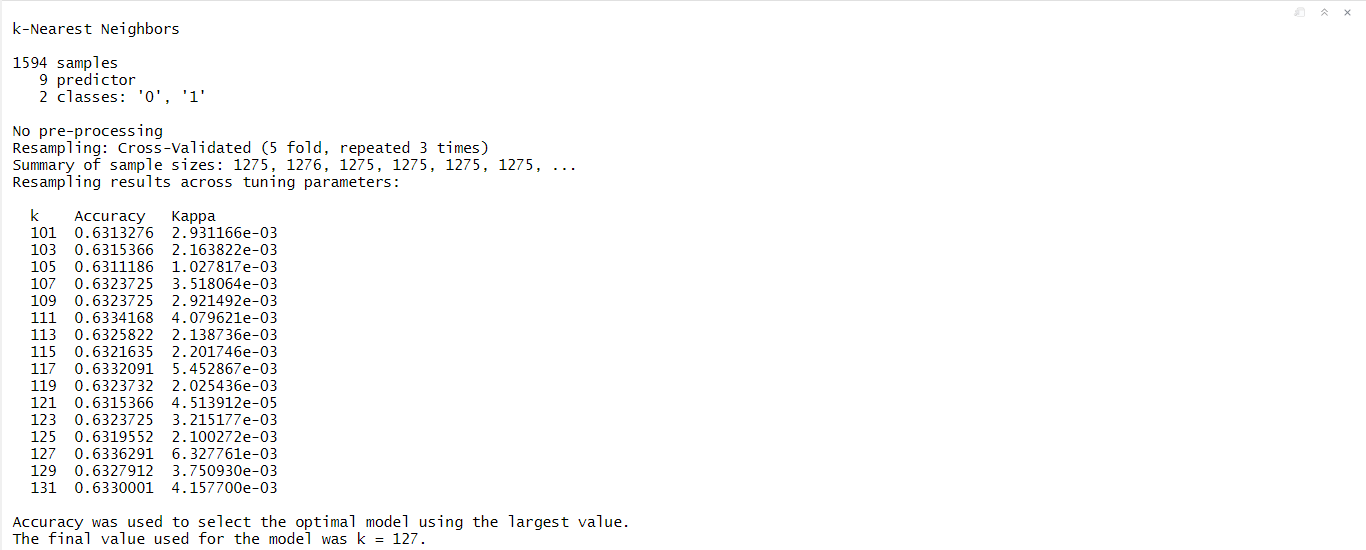
Nhận xét:

* Hàm liên tục và luôn đưa ra giá trị trong khoảng (0, 1)
* Có đạo hàm tại mọi điểm nên có thể dùng gradient descent

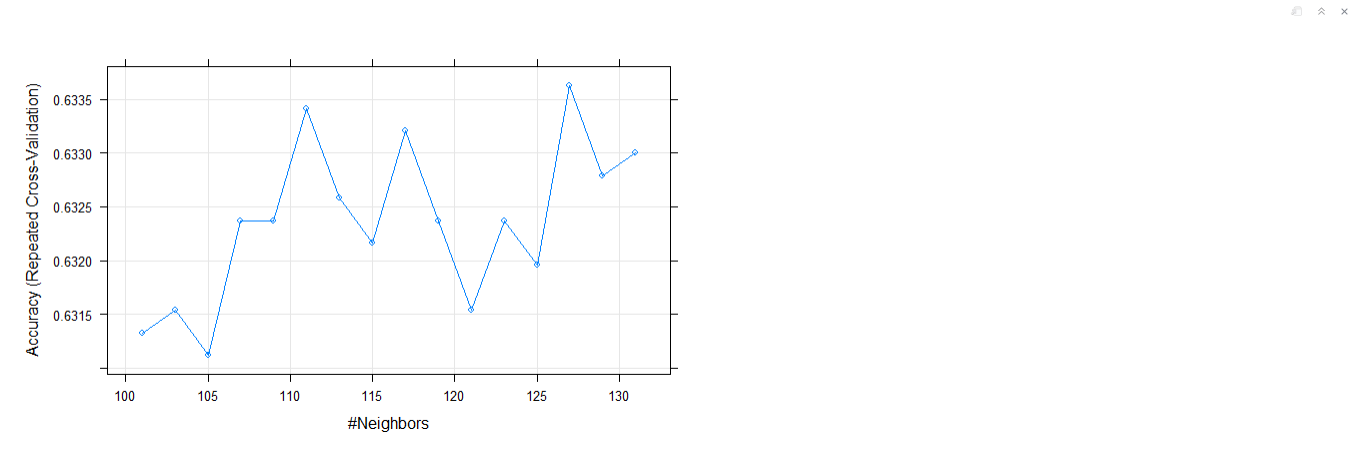
Về cơ bản thì chúng ta sẽ có các bước sau

* Thiết lập model
* Thiết lập hàm mất mát Loss Function
* Tìm tham số bằng việc tối ưu loss function
* Dự đoán dữ liệu mới dựa vào loss function mới tìm được

1. **Thực nghiệm, kết quả, và thảo luận (experiments, results, and discussions)**
   1. KNN



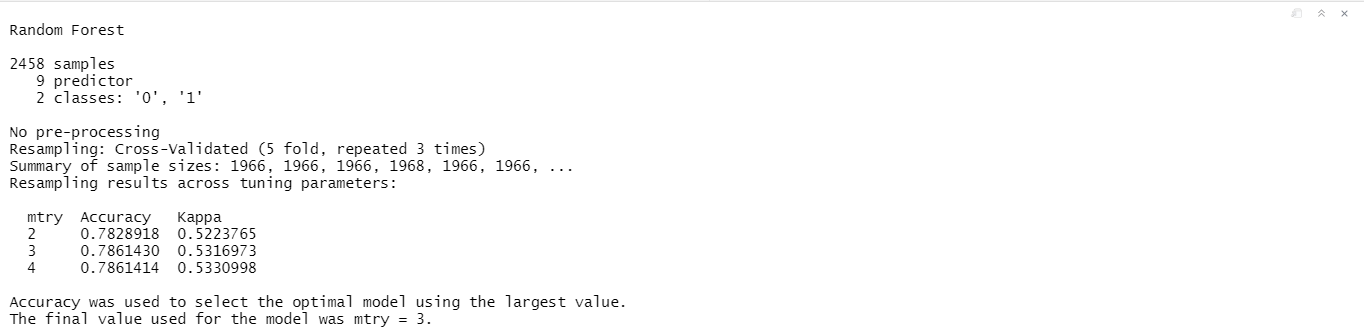
Sau áp dụng thuật toán lên tập training với phương pháp k-fold cross-validate và lặp lại 3 lần, ta được kết quả như hình trên.



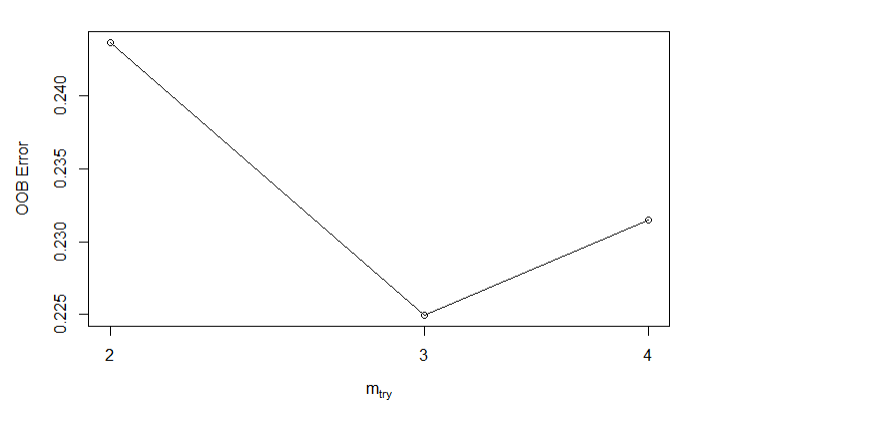
Và kết quả dự đoán chính xác nhất với mô hình KNN là khoảng 63% với k = 127.

**Nhận xét:** Với mô hình KNN cho độ chính xác accurancy là khoảng 63%, giá trị thực sự chưa tốt, chúng ta sẽ tiếp tục thử nghiệm với mô hình khác.

* 1. Random Forest



Chúng ta sẽ thay đổi các tham số ntree và mtry với ntree = [50, 100, 250, 500], mtry= [2,3,4], sau khi train thì với ntree = 500 và mtry = 3 thì mô hình cho kết quả chính xác cao nhất.



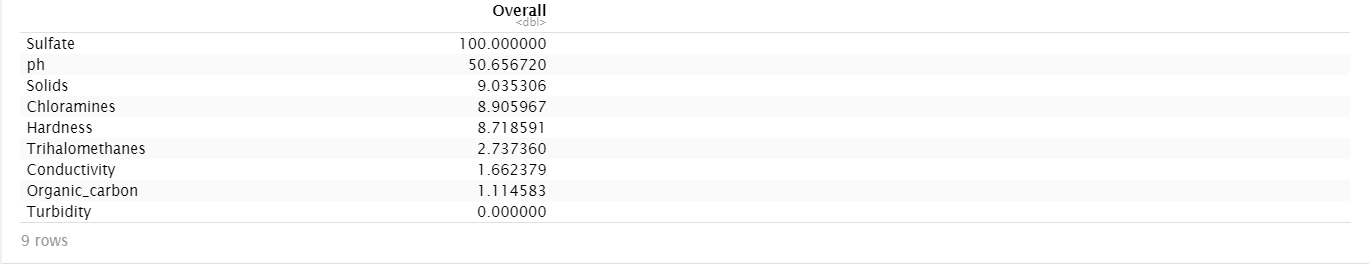
Nhìn vào biểu đồ thì chúng ta thấy tham số mtry tốt nhất của mô hình là 3. Và chúng ta sẽ chọn nó để train mô hình.

Nhận xét: Với mô hình Random Forest cho độ chính xác accurancy là khoảng 79%, giá trị chưa thực sự quá tốt, chúng ta sẽ tiếp tục thử nghiệm với mô hình khác.

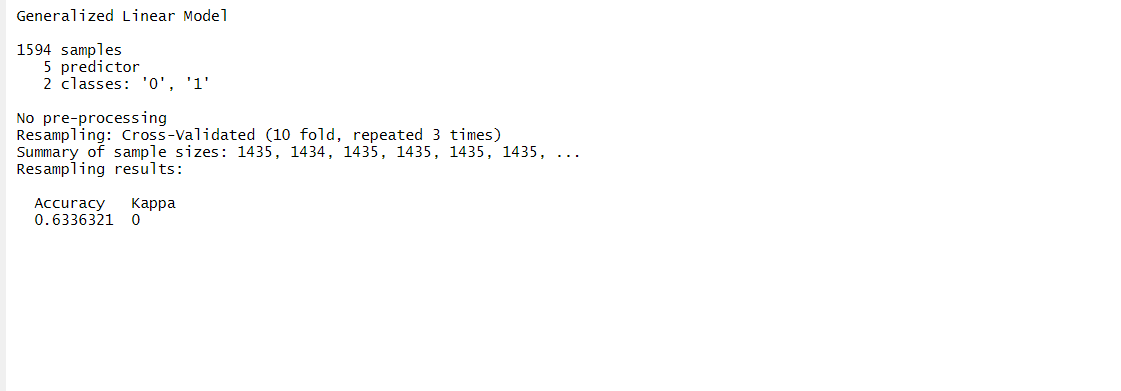
* 1. Logistic Regression

Dựa vào độ quan trọng của các biến dữ đoán biết ở trên thì chúng ta sẽ lược bỏ một số biến ít quan trọng giảm độ phức tạp của mô hình. Cụ thể ở đây chúng ta sẽ chọn 5 biến để dự đoán :"pH", "Solids","Sulfate","Chloramines","Hardness".

Vì các biến này đều là biến số và giá trị trung bình của một số biến có chênh lêch khá lớn nên chúng ta sẽ thực hiện scale các biến này để giảm sự chênh lệch, và tránh việc mô hình phụ thuộc quá vào 1 số biến khi giá trị nó quá lớn.



Sau khi áp dụng mô hình Logistic Regression, ta được kết quả sau:



**Nhận xét:** Với mô hình Logistic Regression cho độ chính xác accurancy là 63.3%, giá trị này khá thấp.

* 1. So sánh các mô hình và chọn mô hình thực hiện trên tập testing
* Sau khi thực hiện train 3 mô hình ở trên và chúng ta thu được kết quả như sau:
* Mô hình KNN với accurancy là 63.3%
* Mô hình Random Forest với accurancy là 79.8%
* Mô hình Logistic Regression với accurancy là 63.3%
* Nhận xét:
* Độ chính xác vẫn chưa quá tốt, như chúng ta thấy với mô hình Random Forest cho được độ chính xác cao nhất khoảng 80%.
* Do đó chúng ta sẽ lựa chọn mô hình này để dự đoán trên tập test.
* Thực hiện dự đoán trên tập test:



* Dựa vào confusion Matrix chúng ta thấy tổng số dự đoán đúng của mô hình Random Forest khoảng hơn 660 trường hợp, dự đoán sai khoảng 150 trường hợp.
* Và mô hình này cho độ chính xác khoảng hơn 81%

1. **Kết luận (conclusion)**

Mô hình Random Forest là mô hình cho kết quả tốt nhất với độ chính xác khoảng 81% trong 3 mô hình mà nhóm em thử huấn luyện mô hình.

Dù độ chính xác chưa cao lắm nhưng có thể dùng nó để khảo sát sàng lọc trước

Tiếp tục phát triển đề tài, nhóm em sẽ tiếp tục sử các phương pháp, độ đo khác cũng như thử nghiệm các mô hình khác để cải thiện mô hình dự đoán được chính xác hơn.

1. **Phụ lục (appendices)**
2. **Đóng góp (contributions)**

**DANH SÁCH THÀNH VIÊN NHÓM VÀ NHIỆM VỤ**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ Tên** | **MSSV** | **Nhiệm vụ** |
| 16 | Nguyễn Anh Đắc | 19133020 | Random Forest |
| 27 | Nguyễn Thanh Tân Kỷ | 19133031 | Logistic Regression |
| 56 | Lại Hữu Trác | 19133059 | Tiền xử lý dữ liệu, EDA, Xử lý các giá trị outlier |
| 52 | Đào Thị Cẩm Tiên | 19133055 | KNN |

1. **Tham khảo (references)**

[1]. <https://viblo.asia/p/knn-k-nearest-neighbors-1-djeZ14ejKWz>

[2]. <https://codelearn.io/sharing/thuat-toan-k-nearest-neighbors-knn>

[3].<https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html>

[4]. <https://viblo.asia/p/phan-lop-bang-random-forests-trong-python-djeZ1D2QKWz>

[5]. <https://viblo.asia/p/confusion-matrix-ma-tran-nham-lan-ma-tran-loi-V3m5WQB7ZO7>

[6].<https://viblo.asia/p/logistic-regression-bai-toan-co-ban-trong-machine-learning-924lJ4rzKPM>