**A white rectangular frame with blue border

Description automatically generatedTRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🕯✡🕮🕮✡🕯**

Ảnh có chứa văn bản, mẫu họa

Mô tả được tạo tự động

**BÁO CÁO ĐỒ ÁN CUỐI KỲ   
MÔN HỌC KHO DỮ LIỆU**

**ĐỀ TÀI:**

**XÂY DỰNG CÁC MÔ HÌNH PHÂN TÍCH DỰ ĐOÁN NGUY CƠ BỆNH TIM MẠCH DỰA TRÊN THÓI QUEN SINH HOẠT, TIỀN SỬ BỆNH**

**Mã lớp học phần: DAMI330484\_23\_2\_03CLC**

**GVHD:** ThS. Trần Trọng Bình.

TP Hồ Chí Minh, tháng 5 năm 2024

**Đề tài: Xây dựng các mô hình phân tích dự đoán nguy cơ mắc bệnh tim mạch dựa vào các thói quen sinh hoạt, tiền sử bệnh**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN THÀNH VIÊN** | **MÃ SỐ SINH VIÊN** | **TỶ LỆ THAM GIA** |
| 1 | Tào Việt Đức | 21110169 | 100% |
| 2 | Vũ Ngọc Nam | 21110252 | 100% |
| 3 | Lê Thị Khánh Ngân | 21110253 | 100% |
| 4 | Nguyễn Hoàng Yến Nhi | 21110267 | 100% |

**Ghi chú:**

Tỷ lệ %: Mức độ phần trăm hoàn thành của từng sinh viên tham gia.

***Trưởng nhóm:*** Tào Việt Đức

**Nhận xét:**

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

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

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

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

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

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

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

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

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

**LỜI CẢM ƠN**

Lời đầu tiên, nhóm em xin gửi lời cảm ơn chân thành nhất đến ***Thầy Trần Trọng Bình*** *–* giảng viên bộ môn ***Data Mining*** của chúng em. Trong quá trình học tập và tìm hiểu bộ môn, chúng em đã nhận được sự quan tâm giúp đỡ, hướng dẫn rất tận tình và tâm huyết từ Thầy. Thầy đã giúp chúng em tích lũy thêm nhiều kiến thức để có cái nhìn sâu sắc và hoàn thiện hơn trong lĩnh vực Công nghệ thông tin. Để từ đó, ứng dụng những kiến thức mà Thầy truyền tải, nhóm em xin trình bày lại những gì mà mình đã học hỏi được thông qua việc thực hiện đề tài **“XÂY DỰNG CÁC MÔ HÌNH PHÂN TÍCH DỰ ĐOÁN NGUY CƠ BỆNH TIM MẠCH DỰA TRÊN THÓI QUEN SINH HOẠT, TIỀN SỬ BỆNH”.**

Kiến thức là vô hạn và sự tiếp nhận kiến thức của bản thân mỗi người luôn tồn tại những hạn chế nhất định. Do đó, trong phạm vi khả năng của bản thân, nhóm em đã rất cố gắng để hoàn thành đề tài một cách tốt nhất. Tuy nhiên, chắc chắn không tránh khỏi những thiếu sót, nhóm chúng em rất mong nhận được sự cảm thông và những ý kiến đóng góp đến từ Thầy để đề tài của nhóm em được hoàn thiện hơn.

Một lần nữa, ***nhóm em xin chân thành cảm ơn Thầy*** đã tận tình hướng dẫn, chỉ bảo các thành viên nhóm em trong suốt quá trình học tập và thực hiện đồ án này.

**Kính chúc Thầy sức khỏe, hạnh phúc thành công trên con đường sự nghiệp giảng dạy.**

***Nhóm 4***

**Mục lục**

[1.Tổng quan về đề tài 6](#_Toc143)

[1.1.Mục tiêu nghiên cứu 6](#_Toc28547)

[1.2.Tập dữ liệu 6](#_Toc8144)

[2.Khai phá dữ liệu 9](#_Toc30240)

[2.1.Khái niệm 9](#_Toc24953)

[2.2.Ứng dụng 9](#_Toc10440)

[3.Kỹ thuật phân loại và các thuật toán 9](#_Toc31292)

[3.1.Phân loại là gì 9](#_Toc1387)

[3.2.Thuật toán knn 11](#_Toc17453)

[3.3.Thuật toán logistic 12](#_Toc2600)

[3.4.Thuật toán decision tree 14](#_Toc2015)

[3.5.Thuật toán random forest 15](#_Toc23669)

[4.Kỹ thuật phân cụm và sử dụng thuật toán K-Means 18](#_Toc32628)

[4.1.Phân cụm là gì 18](#_Toc22113)

[4.2.Thuật toán k-means 21](#_Toc27475)

[5.Ứng dụng Deep learning vào mô hình khai phá dữ liệu - ANN 23](#_Toc4864)

[5.1.Sơ lược về deep learning 23](#_Toc3420)

[5.2.ANN là gì 25](#_Toc6993)

[5.3.Thuật toán ANN 25](#_Toc18119)

[6.Thực nghiệm và đánh giá 30](#_Toc7953)

[6.1. Thực nghiệm 30](#_Toc13858)

[6.1.1.Tiền xử lí 30](#_Toc16723)

[6.1.2.Logistic 33](#_Toc24064)

[6.1.3. K-NN 36](#_Toc32488)

[6.1.4.Decision Tree 38](#_Toc25173)

[6.1.5. Random Forest 40](#_Toc9420)

[6.1.6. ANN 43](#_Toc6558)

[6.1.7.K-means 44](#_Toc16181)

[6.2. Đánh giá 45](#_Toc22585)

[6.2.1. So sánh kết quả phân loại 45](#_Toc28161)

[6.1.2. Đánh giá mô hình ANN 47](#_Toc20385)

[KẾT LUẬN 47](#_Toc12681)

[TÀI LIỆU THAM KHẢO 49](#_Toc26730)

# **1.Tổng quan về đề tài**

## **1.1.Mục tiêu nghiên cứu**

Mục tiêu nghiên cứu chính là dựa vào dữ liệu sức khỏe và lối sống để tìm ra các yếu tố có thể gây ra bệnh tim. Qua đó, xây dựng các mô hình dự đoán bệnh tim bằng cách sử dụng các thuật toán học máy để dự đoán khả năng mắc bệnh tim dựa trên các yếu tố khác nhau trong tập dữ liệu qua đó có thể đưa ra các khuyến nghị về sức khỏe nhằm giúp cải thiện lối sống và chăm sóc sức khỏe để giảm nguy cơ mắc bệnh tim. Mô hình này có thể được sử dụng để phát triển các công cụ hỗ trợ quyết định trong lĩnh vực y tế, giúp bác sĩ và bệnh nhân có được những thông tin hữu ích để phòng ngừa và điều trị bệnh tim một cách hiệu quả hơn.

## **1.2.Tập dữ liệu**

Link dataset: <https://www.kaggle.com/code/abhijeetsinghmeena/cardiovasculardisease-7-models-accuracy-92/input>

Tập dữ liệu gồm 308854 dòng và 19 cột

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Column** | **Description** | **Data Type** |
| **0** | General\_Health | Mức độ sức khỏe tổng quát, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Very Good", "Fair", "Poor". | object |
| **1** | Checkup | Tần suất kiểm tra sức khỏe, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Within the past year", "Within the past 2 years". | object |
| **2** | Excersise | Thói quen tập thể dục, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Yes", "No". | object |
| **3** | Heart\_Disease | Tình trạng bệnh tim, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Yes", "No". | object |
| **4** | Skin\_Cancer | Tình trạng ung thư da, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Yes", "No". | object |
| **5** | Other\_Cancer | Tình trạng ung thư khác, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Yes", "No". | Object |
| **6** | Depression | Tình trạng trầm cảm, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Yes", "No". | Object |
| **7** | Diabetes | Tình trạng bệnh tiểu đường, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Yes", "No". | Object |
| **8** | Arthritis | Tình trạng viêm khớp, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Yes", "No". | object |
| **9** | Sex | Giới tính, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "Male", "Female". | Object |
| **10** | Age\_Category | Nhóm tuổi, biểu diễn dưới dạng chuỗi. Có thể là các giá trị như "18-24", "25-34", "35-44". | Object |
| **11** | Height\_(cm) | Chiều cao, biểu diễn dưới dạng số thực. | Float64 |
| **12** | Weight(kg) | Cân nặng, biểu diễn dưới dạng số thực. | Float64 |
| **13** | BMI | Chỉ số khối cơ thể (Body Mass Index), biểu diễn dưới dạng số thực. | Float64 |
| **14** | Smoking\_History | Lịch sử hút thuốc . | Object |
| **15** | Alcohol\_Consumption | Lượng tiêu thụ rượu, biểu diễn dưới dạng số thực. Có thể là lượng rượu tiêu thụ trung bình hàng tuần hoặc hàng tháng. | Float64 |
| **16** | Fruit\_Consumption | Lượng tiêu thụ trái cây, biểu diễn dưới dạng số thực. Có thể là số lần ăn trái cây trung bình hàng tuần. | Float64 |
| **17** | Green\_Vegetables\_Consumption | Lượng tiêu thụ rau xanh, biểu diễn dưới dạng số thực. Có thể là số lần ăn rau xanh trung bình hàng tuần. | Float64 |
| **18** | FriedPotato\_Consumption | Lượng tiêu thụ khoai tây chiên, biểu diễn dưới dạng số thực. Có thể là số lần ăn khoai tây chiên trung bình hàng tuần. | Float64 |

# **2.Khai phá dữ liệu**

## **2.1.Khái niệm**

Data mining – khai phá dữ liệu là quá trình phân loại, sắp xếp các tập hợp dữ liệu nhất định để xác định xu hướng, các mẫu và thiết lập các mối liên hệ hữu ích nhằm giải quyết các vấn đề nhờ phân tích dữ liệu. Mục tiêu của việc này là cho phép các doanh nghiệp có thể dự đoán được xu hướng tương lai, nhằm đưa ra các quyết định được hỗ trợ dữ liệu từ các tập dữ liệu khổng lồ

Quá trình khai phá dữ liệu là một quá trình phức tạp bao gồm kho dữ liệu chuyên sâu cũng như các công nghệ tính toán.

## **2.2.Ứng dụng**

Data Mining được áp dụng rộng rãi trong các nhóm phân tích dữ liệu và kinh doanh. Dưới đây là một số lĩnh vực ứng dụng phổ biến của kỹ thuật khai phá dữ liệu:

* Phân tích thị trường chứng khoán, tài chính, ngân hàng
* Phát hiện gian lận
* Phân tích doanh nghiệp, quản lý rủi ro
* Hỗ trợ người dùng phân tích và quản lý dữ liệu

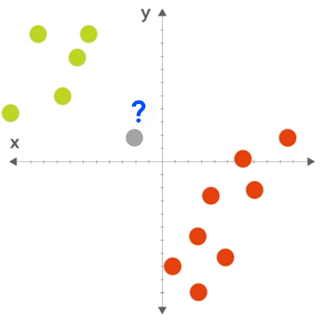
# **3.Kỹ thuật phân loại và các thuật toán**

## **3.1.Phân loại là gì**

Bài toán phân lớp là quá trình phân lớp 1 đối tượng dữ liệu vào 1 hay nhiều lớp đã cho trước nhờ 1 mô hình phân lớp (model).

* Mô hình này được xây dựng dựa trên 1 tập dữ liệu được xây dựng trước đó có gán nhãn (hay còn gọi là tập huấn luyện).
* Quá trình phân lớp là quá trình gán nhãn cho đối tượng dữ liệu.

Bài toán phân lớp (classification) và bài toán gom cụm (cluster) là 2 bài toán lớn trong lĩnh vực Machine Learnig (ML).



*Chấm xám sẽ thuộc lớp xanh hay đỏ?*

Như vậy, nhiệm vụ của bài toán phân lớp là cần tìm 1 **mô hình phần lớp** để khi có dữ liệu mới thì có thể xác định được dữ liệu đó thuộc vào phân lớp nào.

Có nhiều bài toán phân lớp dữ liệu như phân lớp nhị phân (binary classification), phân lớp đa lớp (multiclass classification), phân lớp đa trị.

* Bài toán phân lớp nhị phân là bài toán gán nhãn dữ liệu cho đối tượng vào 1 trong 2 lớp khác nhau dựa vào việc dữ liệu đó có hay không có các đặc trưng (feature) của bộ phân lớp.
* Bài toán phân lớp đa lớp là quá trình phân lớp dữ liệu với số lượng lớp lớn hơn 2. Như vậy với từng dữ liệu phải xem xét và phân lớp chúng vào những lớp khác nhau chứ không phải là 2 lớp như bài toán phân lớp nhị phân. Và thực chất bài toán phân lớp nhị phân là 1 bài toán đặt biệt của phân lớp đa lớp.

Ứng dụng của bài toán này được sử dụng rất nhiều và rộng rãi trong thực tế ví dụ như bài toán nhận dạng khuôn mặt, nhận diện giọng nói, phát hiện email spam, ...



*Phân lớp nhận dạng email spam*

Và đối với các bài toán phân lớp dữ liệu sử dụng các thuật toán học có giám sát (supervised learning) để xây dựng mô hình cho bài toán này.

## **3.2.Thuật toán knn**

K-NN là phương pháp để phân lớp các đối tượng dựa vào khoảng cách gần nhất giữa đối tượng cần phân lớp và tất cả các đối tượng trong dữ liệu huấn luyện. Phương pháp K-NN sẽ tìm K điểm trong bộ dữliệu huấn luyện mà gần với điểm cần phân lớp nhất. Sau đó, điểm này sẽ được gán vào lớp mà đa số láng giềng của nó thuộc về. K là số nguyên dương được xác định trước khi thực hiện thuật toán. Kĩ thuật phân loại sử dụng kNN để gán lớp cho đối tượng Zi được thực hiện như sau (M.Ranageri, 2010):

+Cho X là tập dữliệu huấn luyện với nhãn lớp là c, tập X được kí hiệu X = {(x, c)}, Z là bộ dữ liệu cần gán lớp, r là khoảng cách hình học, C là tập nhãn của lớp.

+ Tính toán khoảng cách giữa điểm Zi đến Xi kí hiệu là R(Zi, Xi)

+ Xác định khoảng cách ngắn nhất Rmin, và lựa chọn tập Uk(z) là tập của k mẫu huấn luyện gần nhất với Zi

+ Gán Zi vào lớp có nhãn là C mà có chứa những điểm hàng xóm gần với Zi nhất.

Đặc trưng của kĩ thuật kNN là xác định một số mẫu huấn luyện hoặc nguyên mẫu của nó, đây là phương pháp phân loại có độ chính xác dựa hoàn toàn vào khoảng cách. Do đó, nó là phương pháp phù hợp với ứng dụng dự đoán kết quả. Quá trình huấn luyện của phân lớp với k-NN là tương đối đơn giản, nhưng quá trình kiểm tra của k-NN sẽ chậm hơn.

Với k-NN việc tính khoảng cách từ điểm kiểm tra đến dữ liệu huấn luyện sẽ quyết định độ chính xác của lớp mà nó thuộc về, do đó quá trình này vô cùng quan trọng. Công thức để tính toán khoảng cách hay được dùng trong K-NN là:

Cho hai điểm X = (x1, x2, …, x2), Y = (y1, y2,…, yn) trong không gian R2. Khoảng cách từ một điểm p đến hai điểm X, Y được định nghĩa:

𝐷𝑝 =( xi –yi|p )1/p (1)

Trong đó:

Nếu p =1 khoảng cách này là khoảng cách Manhattan

Nếu p = 2 đây là khoảng cách Euclide

Nếu p = ∞, khoảng cách vô cùng được định nghĩa theo công thức Di = max{|xi – yi|}

## **3.3.Thuật toán logistic**

Hồi quy Logistic là một mô hình thống kê được sử dụng để phân loại nhị phân, tức dự đoán một đối tượng thuộc vào một trong hai nhóm. Hồi quy Logistic làm việc dựa trên nguyên tắc của hàm sigmoid – một hàm phi tuyến tự chuyển đầu vào của nó thành xác suất thuộc về một trong hai lớp nhị phân.

Logistic Regression sử dụng hàm phi tuyến để xác định xác suất của hai lớp 0 và 1.

Hồi quy Logistic hoạt động dựa trên hàm Sigmoid, được biểu diễn như sau:

𝑆(𝑧)=

Hàm Sigmoid nhận đầu vào là một giá trị z bất kỳ, và trả về đầu ra là một giá trị xác suất nằm trong khoảng [0,1]. Khi áp dụng vào mô hình Hồi quy Logistic với đầu vào là ma trận dữ liệu X và trọng số 𝑤w, ta có 𝑧=𝑋𝑤.

Việc huấn luyện của mô hình là tìm ra bộ trọng số w sao cho đầu ra dự đoán của hàm Sigmoid gần với kết quả thực tế nhất. Để làm được điều này, ta sử dụng hàm mất mát ([Loss](https://trituenhantao.io/tu-dien-thuat-ngu/loss/) Function) để đánh giá hiệu năng của mô hình. Mô hình càng tốt khi hàm mất mát càng nhỏ.

Hàm mất mát ([Loss](https://trituenhantao.io/tu-dien-thuat-ngu/loss/) Function) là một hàm số được sử dụng để đo lường mức độ lỗi mà mô hình của chúng ta tạo ra khi dự đoán các kết quả từ dữ liệu đầu vào. Trong bài toán Hồi quy Logistic, chúng ta sử dụng hàm mất mát [Cross-Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) (còn gọi là Log [Loss](https://trituenhantao.io/tu-dien-thuat-ngu/loss/)) để đánh giá hiệu năng của mô hình.

Hàm mất mát [Cross-Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) được định nghĩa như sau:  
 𝐿(𝑤)= yi log𝑝i +(1−𝑦i )log(1−𝑝i)]

Trong đó:

𝑛: số lượng mẫu dữ liệu trong tập huấn luyện.

𝑦i : giá trị thực tế của đầu ra thứ i.

𝑝i : xác suất dự đoán thuộc lớp 1 của mô hình cho đầu vào thứ i.

Hàm [Cross-Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) đo lường khoảng cách giữa hai phân phối xác suất 𝑦𝑖yi và 𝑝𝑖pi . Khi mô hình dự đoán chính xác, tức là nếu 𝑦i=1 thì 𝑝i càng gần 1, và nếu 𝑦i=0 thì 𝑝i càng gần 0, sau đó hàm mất mát sẽ tiến gần về 0.

Trong quá trình huấn luyện, chúng ta tìm cách cập nhật bộ trọng số 𝑤 sao cho giá trị hàm mất mát [Cross-Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/cross-entropy/) đạt giá trị nhỏ nhất, dẫn đến một mô hình dự đoán tốt nhất.

Để tìm giá trị tối ưu cho bộ trọng số 𝑤, chúng ta có thể sử dụng kỹ thuật [Gradient](https://trituenhantao.io/tu-dien-thuat-ngu/gradient/) Descent. Tại mỗi bước lặp, chúng ta cập nhật 𝑤w theo phương từm ứng với đạo hàm của hàm mất mát 𝐿(𝑤) theo 𝑤.

## **3.4.Thuật toán decision tree**

Cây quyết định là một cấu trúc phân cấp của các nút và các nhánh có các tính chất sau:

* + 1. Mỗi nút trong (internal node) biểu diễn một thuộc tính cần kiểm tra giá trị (an attribute to be

tested) đối với các các tập thuộc tính.

* + 1. Nút lá (leaf node) hay còn gọi là nút trả lời biểu thị cho một lớp các trường hợp mà nhãn của nó là tên của lớp, nó biểu diễn một lớp (a classififcation)
    2. Nút nhánh (branch) từ một nút sẽ tương ứng với một giá trị có thể của thuộc tính gắn với nút đó.
    3. Nhãn (lable) của nút này là tên của thuộc tính và có một nhánh nối nút này đến các cây con ứng với mỗi kết quả có thể có phép thử. Nhãn của nhánh này là các giá trị của thuộc tính đó. Nút trên cùng gọi là nút gốc.

Quá trình xây dựng một cây quyết định cụ thể bắt đầu bằng một nút rỗng bao gồm toàn bộ các đối tượng huấn luyện và làm như sau:

Bước1: Nếu tại nút hiện thời, tất cả các đối tượng huấn luyện đều thuộc vào một lớp nào đó thì nút này

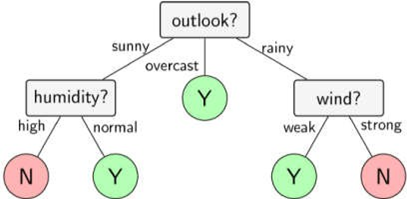
chính là nút lá có tên là nhãn lớp chung của các đối tượng.

Bước 2: Trường hợp ngược lại, sử dụng một độ đo, chọn thuộc tính điều kiện phân chia tốt nhất tập mẫu huấn luyện có tại nút.

Bước 3: Tạo một nút con của nút hiện thời bằng số các giá trị khác nhau của thuộc tính được chọn. Gán cho mỗi nhánh từ nút cha đến nút con một giá trị của thuộc tính rồi phân chia các đối tượng huấn luyện vào các nút con tương ứng

Bước 4: Nút con K được gọi là thuần nhất, trở thành lá, nếu tất cả các đối tượng mẫu tại đó đều thuộc vào cùng một lớp

Bước 5: Lặp lại bước 1 – 3 đối với mỗi nút chưa thuần nhất.

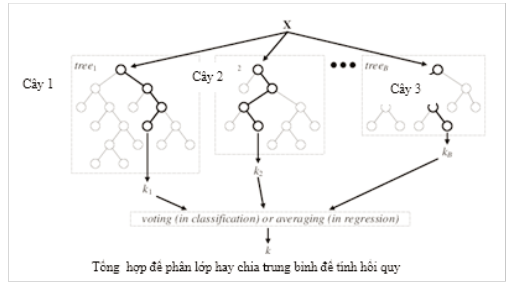


*Hình 1. Ví dụ về cây quyết định*

Có nhiều thuật toán để xây dựng cây quyết định như ID3, CART, J48, C4.5…Việc lựa chọn thuật toán nào để việc phân lớp đạt hiệu quả cao, kết quả đáng tin cậy phụ thuộc vào nhiều yếu tố, trong đó cấu trúc dữ liệu của các thuộc tính sẽ có ảnh hưởng lớn đến kết quả của các thuật toán.

## **3.5.Thuật toán random forest**

Random forest là một phương pháp thống kê mô hình hóa bằng máy (machine learning statistic) dùng để phục vụ các mục đích phân loại, tính hồi quy và các nhiệm vụ khác bằng cách xây dựng nhiều cây quyết định (Decision tree). Một cây quyết định là một cách đơn giản để biểu diễn một giao thức (Protocol). Nói cách khác, cây quyết định biểu diễn một kế hoạch, trả lời câu hỏi phải làm gì trong một hoàn cảnh nhất định.Mỗi Node của cây sẽ là các thuộc tính, và các nhánh là giá trị lựa chọn của thuộc tính đó. Bằng cách đi theo các giá trị thuộc tính trên cây, cây quyết định sẽ cho ta biết giá trị dự đoán. Nhómthuật toán cây quyết định có một điểm mạnh đó là có thể sử dụng cho cả bài toán Phân loại(Classification) và Hồi quy (Regression).Random Forest có khả năng tìm ra thuộc tính nào quan trọng hơn so với những thuộc tínhkhác. Trên thực tế, nó còn có thể chỉ ra rằng một số thuộc tính là không có tác dụng trong cây quyết định.



*Hình 1: Sơ đồ biểu diễn các cây quyết địnhtrong phương pháp random forest*

Từ hình 1 chúng ta thấy rằng Random Forest được cấu thành bởi một số cây quyết định. Các cây này cùng nhận đầu vào là đối tượng x và đưara quyết định về danh mục thuộc tính (Attribute category) của x. Các quyết định này sẽ được tổng hợp lại lấy trung bình để chọn ra quyết định cuối cùng.

Mô tả phương pháp random forest

+ Lựa chọn cây quyết định (decision treelearning): Cây quyết định là một phương pháp phổ biếncho các nhiệm vụ mô hình hóa bằng máy(machine learning). Các cây quyết định được lựachọn với các tiêu chí phù hợp để đáp ứng các yêu cầu nhiệm vụ phục vụ khai thác dữ liệu. Các cây quyết định được thiết kế với xu hướng nhận biết được cả những yếu tố bất thường: phù hợp vớicác mẫu có độ lệch nhỏ nhưng phương sai lớn.

+Thuật toán mô hình bằng máy:Thuật toán lấy mẫu cho phương pháp random forest ứng dụng cho các phương pháp sử dụng thuật toán mô tả thống kê để ước lượng số lượngtừ một mẫu dữ liệu (bagging). Ví dụ như một một tập mẫu X = x1, ..., xn với các câu trả lời Y= y1, ..., yn, lấy giá trị trung bình (B lần), chọn một mẫu ngẫu nhiên từ bộ mẫu phù hợp với cây quyết định:

Lặp b = 1,..., B:

n mẫu từ giá trị tọa độ (X, Y); gọi là (Xb, Yb);

lớp dữ liệu hay kết quả hồi quy fb của biến Xb,Yb;

Sau khi lấy mẫu, các phép tính toán cho cácmẫu là ẩn số x’ có thể được thực hiện bằng cách lấy trung bình các giá trị nội suy từ tất cả các câyhồi quy riêng lẻ của biến x’ hoặc lấy giá trị từ đa số của các mẫu trong cây quyết định:

Phương pháp thống kê này ước lượng một giá trị trung bình từ số lượng mẫu dữ liệu. Chúng ta cần rất nhiều mẫu từ tập dữ liệu, tính giá trị trung bình. Sau đó, tính trung bình tất cả các giá trị trung bình của các tập dữ liệu trong các cây quyết định thành phần để tính toán được tốt hơn giá trị trung bình thật. Kết quả dẫn đến hiệu suất mô hình được tính toán sẽ tốt hơn vì nó làm giảm phương sai của mô hình, mà không làm tăng độlệch. Điều này có nghĩa là khi thiết kế nhiều cây quyết định trong một tập các mẫu được lấy sẽđưa ra sự tương quan tốt hơn của các cây quyếtđịnh với nhau.

+Từ thuật toán mô hình bằng máy đến Random forest: Các bước trên đã mô tả cách thực hiện thuật toán thống kê để ước lượng giá trịt rung bình từ số lượng các cây quyết định của tập mẫu dữ liệu (bagging). Phương pháp random forest khác cơ bản so với phương pháp thống kê trên là chúng sử dụng thuật toán xử lý theo các cây quyết định (tree learning algorithm). Tại mỗi phần tử ở trong quy trình này được gán ngẫu nhiên các tập con thuộc tính của các mẫu. Lý do thực hiện quy trình này là sự tương quan của các cây quyết định thành phần trong một thuật toán thống kê để ước lượng giá trị trung bình từ số lượng các cây quyết định thông thường: nếu một hoặc một vài thuộc tính là các yếu tố dự báo mạnh cho biến đầu ra, các tính năng này sẽ được chọn trong nhiều cây B, chúng sẽ trở nên tương quan. Random forest có thể sắp xếp sự quan trọng của các biến trong các bài toán phân loại hay hồi quy. Các phương pháp sắp xếp có thể được môtả trong các nghiên cứu của Breiman. Bước đầu tiên để xác định các biến quan trọng trong 1 tậpdữ liệu là làm phù hợp phương pháp random forest với tập dữ liệu:

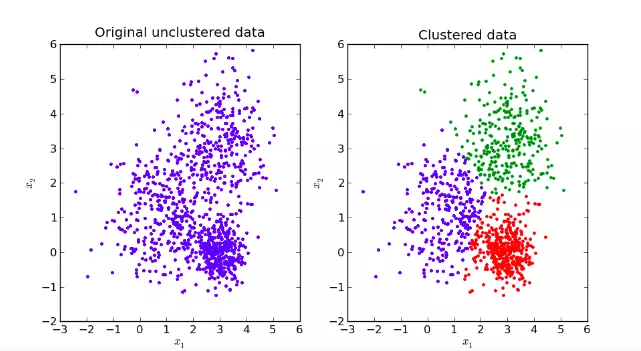
Dn ={(Xn, Yn)}n với i=1

Trong quá trình này, các lỗi dự báo xảy ra(out-of-bag error) tại mỗi điểm xử lý được ghi lại và tính giá trị trung bình. Để xác định được tính quan trọng của đối tượng thứ i sau khi lấy mẫu,các giá trị của mẫu i được hoán vị trong tập mẫu và các lỗi dự báo được tính toán lại trong tập dữliệu. Độ quan trọng của đối tượng được tínhbằng điểm, các điểm được tính toán bằng cách lấy trung bình của độ chênh lệch giữa các lỗi dự báo trước và sau khi hoán vị. Các đối tượng có giá trị lớn được xếp quan trọng hơn các điểm có giá trị nhỏ

# **4.Kỹ thuật phân cụm và sử dụng thuật toán K-Means**

## **4.1.Phân cụm là gì**

Phân cụm dữ liệu là bài toán gom nhóm các đối tượng dữ liệu vào thánh từng cụm (cluster) sao cho các đối tượng trong cùng một cụm có sự tương đồng theo một tiêu chí nào đó.



Đặc điểm:

* Số cụm dữ liệu không được biết trước
* Có nhiều các tiếp cận, mối cách lại có vài kỹ thuật
* Các kỹ thuật khác nhau thường mang lại kết quả khác nhau.

Các độ đo khoảng cách

Tính chất của độ đo khoảng cách:

* Tính không âm (non-negative): d(x,y) >=0 và d(x, y) = 0 khai và chỉ khi x trùng y.
* Tính đối xứng (symmetic): d(x, y) = d(y, x)
* Tính tam giác (traingle inequality): d(x, y) + d(y, z) >= d(x, z)

Độ do Euclid chuẩn và độ đo Manhattan

* Cho hai điểm x = (x1, x2, ..., xm) và y = (y1, y2, ..., ym)
* Độ đo Euclid được xác định theo công thức

*d(x,y)=(xi -yi |r )1/r*

* Độ đo Euclid chuẩn (r = 2)

*d(x,y)=*

* Độ đo Manhattan

*d(x,y)=*

Độ đo Cosine

* Cho hai vecto x = (x1, x2, ..., xm) và y = (y1, y2, ..., ym)
* Độ đo Cosine được tính như sau
* Trong không gian dương:
  + Thoả mãn cả 3 tính chất
  + Giá trị nằm trong khoảng [0, 1]

Độ đo Hamming

* Được sử dụng khi các vector ở dạng logic (true/false, 0/1)
* Khoảng cách giữa hai vecto được xác định là số chiều mà ở đó các giá trị tương ứng của hai vecto là khác nhau.
* Thỏa mãn cả 3 tính chất
* VD: v1(0, 1, 0, 1, 0) và v2 (1, 1, 0, 1, 0) vậy d(v1, v2) = 1

Độ đo Jaccard

* x, y là hai tập hợp
* Chỉ số Jaccard

*J(x,y)=*

* Độ đo Jaccard

*d(x,y)=1-J(x,y)*

* Thỏa mãn cả 3 tính chất

Độ đo Kullback-Leibler (KL)

* Cho x = (x1, x2, ..., xm) và y = (y1, y2, ..., ym) là hai phân phối xác xuất rời rạc.
* Độ đo KL được tính như sau:

*DKL (x||y)=*

Trong đó không xét những vị trí có xi = 0 hoặc yi = 0.

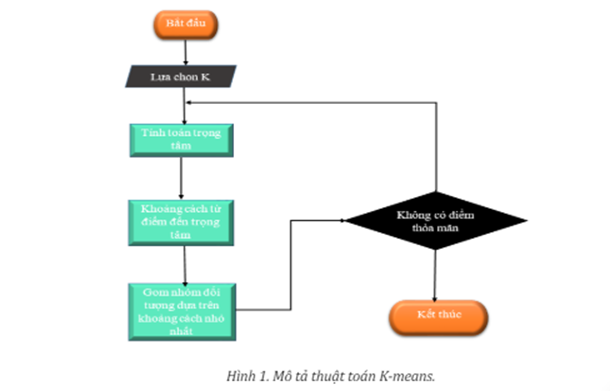
* KL không thoải mãn tính chất đối xứng, tức DKL(x||y) có thể khác DKL(y||x)
* Đó đó, có thể tính độ đo dựa trên KL như sau:

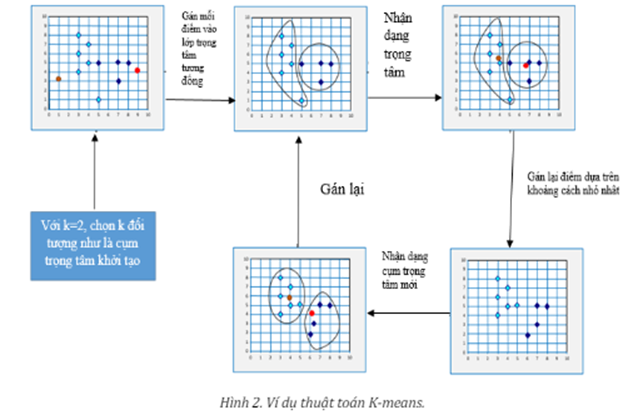
*d(x,y)=*

## **4.2.Thuật toán k-means**

Thuật toán K-means là tìm phương pháp phân nhóm các đối tượng (objects) đã cho vào K cụm (K là số các cụm được xác định trước, K > 0) sao cho tổng bình phương khoảng cách giữa các đối tượng đến tâm nhóm là nhỏ nhất.

Thuật toán K-means được mô tả trên hình 1 và hình 2.





Thuật toán K-means trong bài toán phân loại dữ liệu: Trong bài toán phân loại dữliệu, thuật toán Kmeans được triển khai theo các bước như sau (Alex Berson et al., 2005):

Bước 1: chọn K cụm trọng tâm khởi tạo, z1, z2, z3, …, zn, với 0 < K ≤ n

Bước 2: phân phối mẫu trong K-means. Mẫu thường được gán với cụm trung tâm gần nhất theo công thức: x Î Si(n) nếu |x – zi(n)| ≤ |x – zj(n)| với j = 1, 2, 3, …, k; i ≠ j; Si(n) là bộ mẫu của trọng tâm zi(n), trong đó n chỉ số bước lặp của bài toán.

Bước 3: tính toán trọng tâm cụm mới từ mỗi cụm Si(n). Tìm giá trị mới cho mỗi zi. Trọng tâm cụm mới, zi(n+1) sẽ là giá trị trung bình của các điểm trong Si(n) như sau:

*𝑧i (n+1) =*

Trong đó ci là tập điểm thuộc về cụm thứ i.

Bước 4: so sánh zi(n) và zi(n+1) với mọi i.

Tính toán khoảng cách giữa mỗi cặp điểm trong mỗi lần lặp liên tiếp:

Nếu không có sự thay đổi đáng kể, kết thúc phương pháp, một vài tiêu chí cho kết thúc như:

+ Nếu |zi(n+1) – zi(n)| < T với mọi i

+ Nếu 𝑧i (𝑛 + 1) – 𝑧i (𝑛)| < 𝑇 với mọi i.

Nếu không thì tiếp tục lặp các lần lặp tiếp theo từ bước 2

Trong thuật toán K-means việc chọn được giá trị k sẽ có thể giúp tăng tốc được thuật toán, tối ưu và cải tiến thuật toán tốt hơn. Có nhiều phương pháp để có thể lựa chọn được giá trị k như sử dụng ý kiến của chuyên gia, thử mô hình với các giá trị của k và từđó chọn k tốt nhất hay sử dụng kỹ thuật CV (Cross - Validation), ….

# **5.Ứng dụng Deep learning vào mô hình khai phá dữ liệu - ANN**

## **5.1.Sơ lược về deep learning**

Deep Learning là gì? **Deep Learning (học sâu)** cũng có thể được coi là một lĩnh vực thuộc [Machine Learning](https://tanca.io/blog/machine-learning-la-gi-tim-hieu-ve-cach-hoat-dong-va-ung-dung) – nơi máy tính tự học và cải thiện thông qua các thuật toán. Thế nhưng, Deep Learning được xây dựng dựa trên những khái niệm phức tạp hơn.

Chủ yếu hoạt động với mạng thần kinh nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người. Thực ra các khái niệm liên quan đến mạng nơ-ron nhân tạo và Deep Learning đã được phát triển những năm 1960. Nhưng nó bị giới hạn bởi lượng dữ liệu và khả năng tính toán tại thời điểm đó.

Trong những năm gần đây, những tiến bộ trong phân tích big data đã cho phép chúng ta tận dụng tối đa khả năng của mạng lưới thần kinh nhân tạo. Mạng nơ-ron nhân tạo (neural networks) là động lực chính đằng sau sự phát triển của Deep Learning.

Mạng lưới thần kinh sâu (DNN - Deep neural networks) bao gồm nhiều lớp tế bào thần kinh khác nhau, có khả năng thực hiện các tính toán rất phức tạp. Deep Learning hiện đang phát triển rất nhanh và được coi là một trong những bước đột phá lớn nhất trong Machine Learning.

Cách thức hoạt động của mô hình học sâu - deep learning

Mạng thần kinh nhân tạo trong Deep Learning được xây dựng để mô phỏng khả năng tư duy của bộ não con người. Cách thức hoạt động như sau:

* Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, càng nhiều lớp thì mạng sẽ càng “sâu”. Trong mỗi lớp là các node (nút mạng) và được liên kết với các lớp khác bên cạnh.
* Mỗi kết nối giữa các nút sẽ có một trọng số tương ứng, trọng số càng cao thì mức độ ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.
* Mỗi nơ-ron sẽ có một chức năng kích hoạt, về cơ bản chịu trách nhiệm "chuẩn hóa" đầu ra từ nơ-ron này.
* Dữ liệu được người dùng nhập vào mạng thần kinh đi qua tất cả các lớp và trả về kết quả ở layer cuối cùng, được gọi là lớp đầu ra (output layer).
* Trong quá trình đào tạo mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm tập giá trị của các trọng số sao cho phán đoán đúng nhất.

Các hệ thống Deep Learning yêu cầu phần cứng rất mạnh để có thể xử lý lượng lớn dữ liệu và thực hiện các phép tính phức tạp. Nhiều mô hình Deep Learning có thể mất hàng tuần hoặc thậm chí hàng tháng để triển khai trên phần cứng tiên tiến nhất hiện nay.

Ưu nhược điểm học sâu (deep learning)

Deep Learning là một bước ngoặt lớn trong lĩnh vực trí tuệ nhân tạo - artificial intelligence. Nó cho phép các nhà khoa học dữ liệu khác xây dựng nhiều mô hình có độ chính xác cao trong các lĩnh vực nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, data…

Một số ưu điểm nổi bật của Deep Learning bao gồm:

* Cấu trúc neural networks linh hoạt, dễ dàng thay đổi để phù hợp với nhiều thuật toán khác nhau.
* Có khả năng giải nhiều vấn đề phức tạp với độ chính xác rất cao.
* Khả năng tự động hóa cao, tự điều chỉnh và tự tối ưu hóa.
* Có khả năng thực hiện tính toán song song, hiệu năng tốt, xử lý lượng dữ liệu lớn.

Bên cạnh ưu điểm, Deep Learning vẫn còn tồn tại nhiều hạn chế:

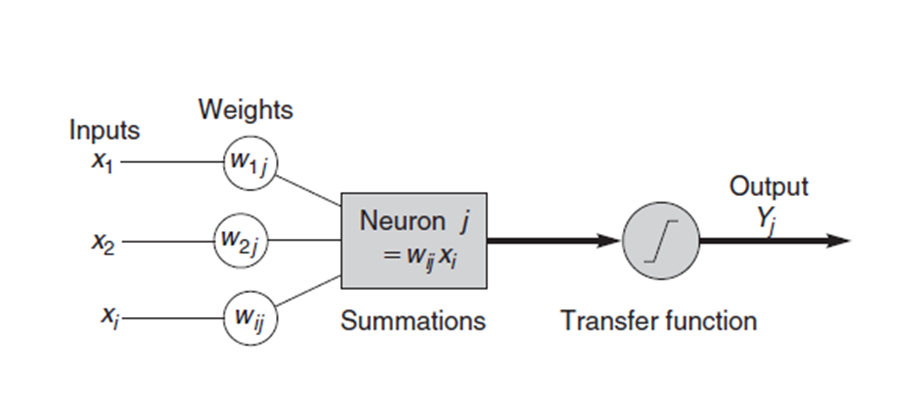
* Cần khối lượng dữ liệu khổng lồ để khai thác tối đa khả năng của Deep Learning.
* Chi phí tính toán cao vì phải xử lý nhiều mô hình phức tạp.
* Không có nền tảng lý thuyết vững chắc để chọn các công cụ tối ưu cho Deep Learning.

## **5.2.ANN là gì**

Artificial Neural Networks là một mô hình toán học hoặc mô hình tính toán được xây dựng dựa trên các khái niệm về neural sinh học. Nó bao gồm các đơn vị neural nhân tạo, có khả năng kết nối với nhau và xử lý thông tin thông qua truyền dẫn tín hiệu qua các kết nối và tính toán giá trị mới tại các đơn vị.

## **5.3.Thuật toán ANN**

- Cấu trúc và các thành phần của một mô hình ANN



Inputs: Mỗi Input tương ứng với 1 thuộc tính (attribute) của dữ liệu (patterns). Ví dụ như trong ứng dụng của ngân hàng xem xét có chấp nhận cho khách hàng vay tiền hay không thì mỗi Input là một thuộc tính của khách hàng như thu nhập, nghề nghiệp, tuổi, số con,…

Output: Kết quả của một ANN là một giải pháp cho một vấn đề, ví dụ như với bài toán xem xét chấp nhận cho khách hàng vay tiền hay không thì output là yes (cho vay) hoặc no (không cho vay).

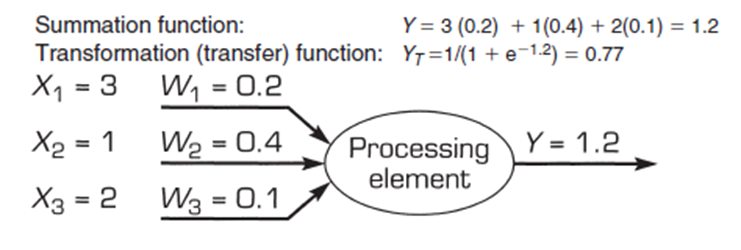
Connection Weights (Trọng số liên kết) : Đây là thành phần rất quan trọng của một ANN, nó thể hiện mức độ quan trọng (độ mạnh) của dữ liệu đầu vào đối với quá trình xử lý thông tin (quá trình chuyển đổi dữ liệu từ Layer này sang layer khác). Quá trình học (Learning Processing) của ANN thực ra là quá trình điều chỉnh các trọng số (Weight) của các input data để có được kết quả mong muốn.

Summation Function (Hàm tổng): Tính tổng trọng số của tất cả các input được đưa vào mỗi Neuron (phần tử xử lý PE). Hàm tổng của một Neuron đối với n input được tính theo công thức sau:

***Tổng trọng số****=*

Transformation (Transfer) Function (Hàm chuyển đổi):

Hàm tổng (Summation Function) của một Neuron cho biết khả năng kích hoạt (Activation) của neuron đó còn gọi là kích hoạt bên trong (internal activation). Các Nueron này có thể sinh ra một output hoặc không trong ANN (nói cách khác rằng có thể output của 1 Neuron có thể được chuyển đến layer tiếp trong mạng Neuron theo hoặc không). Mối quan hệ giữa Internal Activation và kết quả (output) được thể hiện bằng hàm chuyển đổi (Transfer Function).



Việc lựa chọn Transfer Function có tác động lớn đến kết quả của ANN. Hàm chuyển đổi phi tuyến được sử dụng phổ biến trong ANN là sigmoid (logical activation) function.

*YT=*

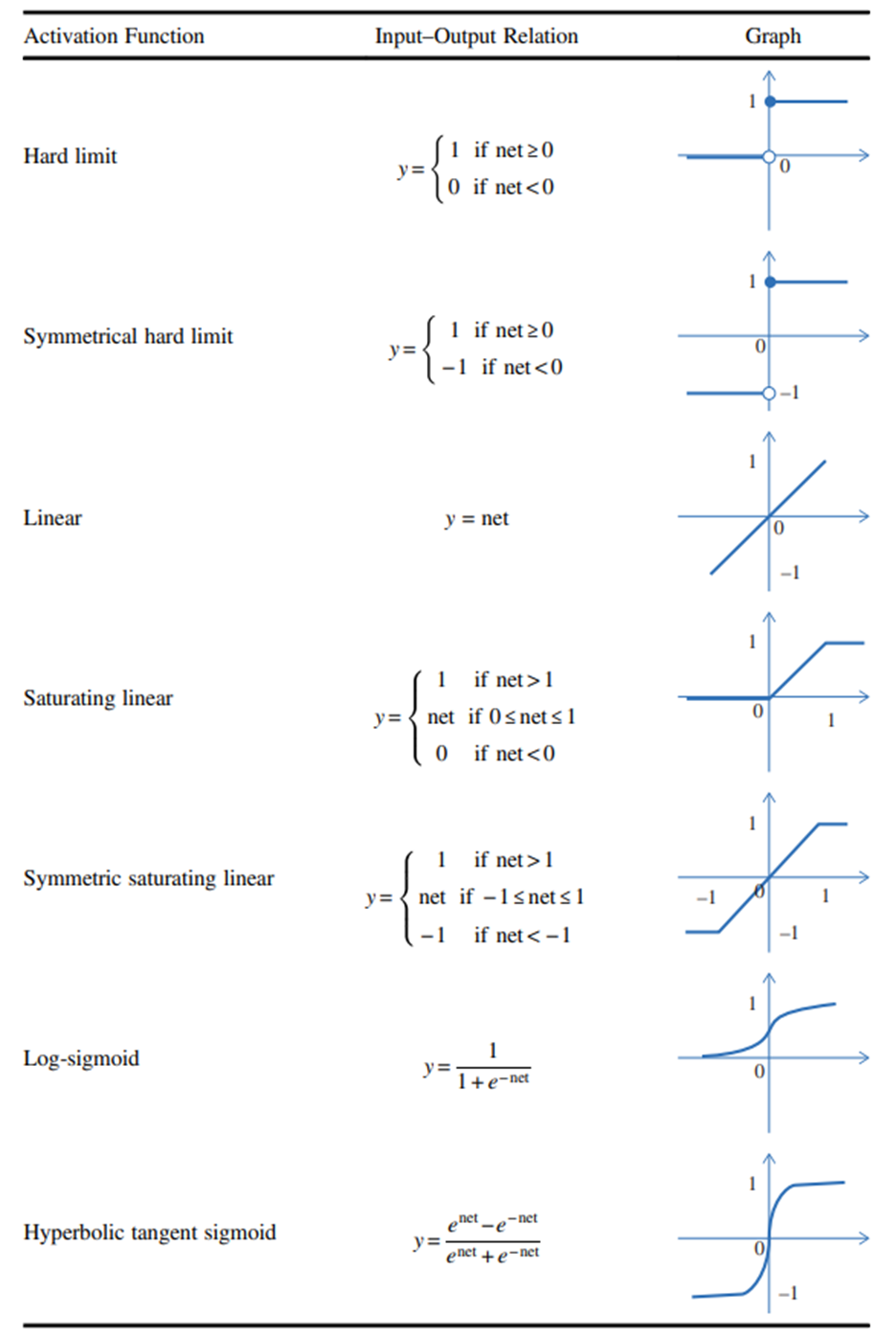
Trong đó :

YT: Hàm chuyển đổi

Y: Hàm tổng

Kết quả của Sigmoid Function thuộc khoảng [0,1] nên còn gọi là hàm chuẩn hóa (Normalize Function).

Kết quả xử lý tại các Neuron (Output) đôi khi rất lớn, vì vậy transfer function được sử dụng để xử lý output này trước khi chuyển đến layer tiếp theo. Đôi khi thay vì sử dụng Transfer Function người ta sử dụng giá trị ngưỡng (Threshold value) để kiểm soát các output của các neuron tại một layer nào đó trước khi chuyển các output này đến các Layer tiếp theo. Nếu output của một neuron nào đó nhỏ hơn Threshold thì nó sẽ không được chuyển đến Layer tiếp theo.



# **6.Thực nghiệm và đánh giá**

## **6.1. Thực nghiệm**

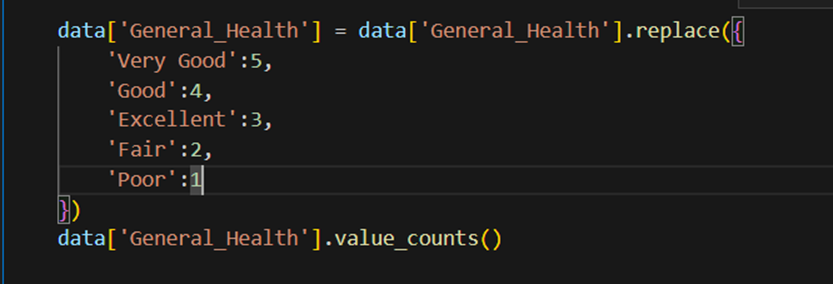
### **6.1.1.Tiền xử lí**

#### **6.1.1.1.Encoding các cột dữ liệu**

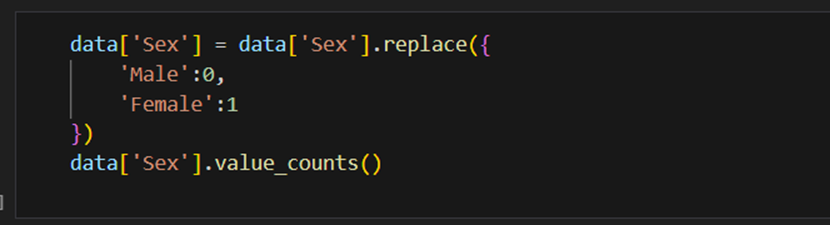
Thực hiện thay giá trị các cột 'Exercise', 'Heart\_Disease', 'Skin\_Cancer', 'Other\_Cancer', 'Depression', 'Arthritis', 'Smoking\_History' có giá trị ‘Yes’, ‘No’ lần lượt là 1 và 0

|  |  |
| --- | --- |
| Dữ liệu gốc | Encoding |
|  |  |

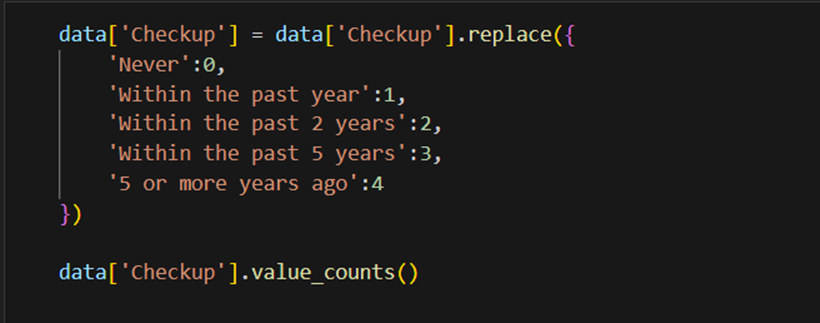
-Thực hiện encoding giá trị cột General\_Health:



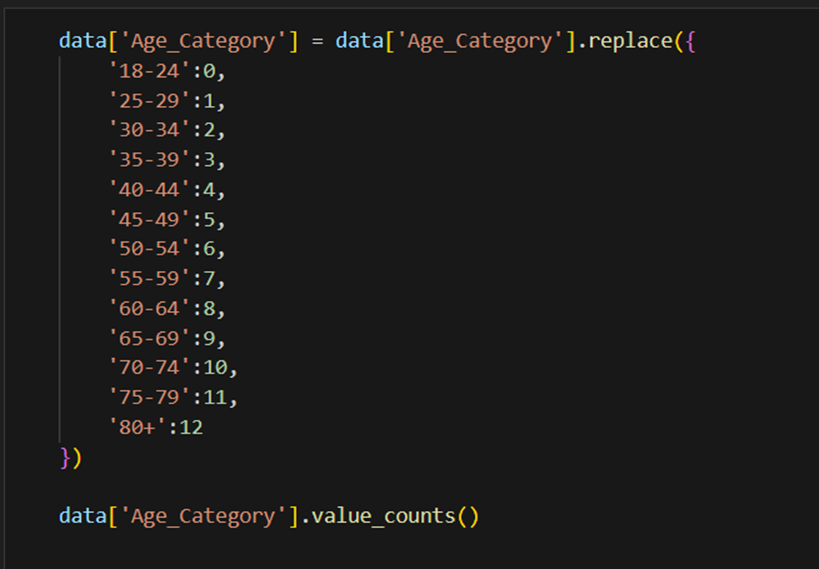
-Thực hiện encoding giá trị cột Sex:



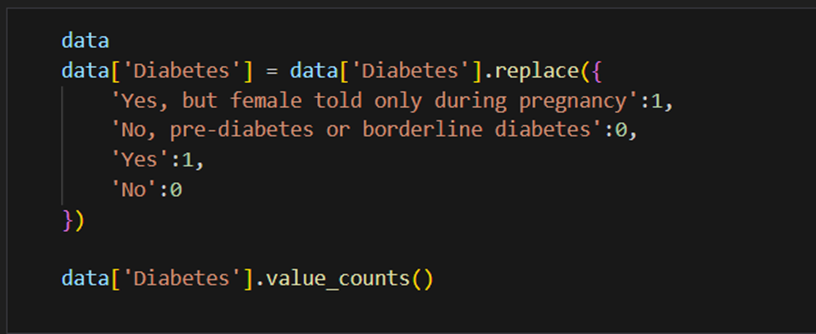
-Thực hiện encoding giá trị cột Checkup:



-Thực hiện encoding giá trị cột Age\_Category:

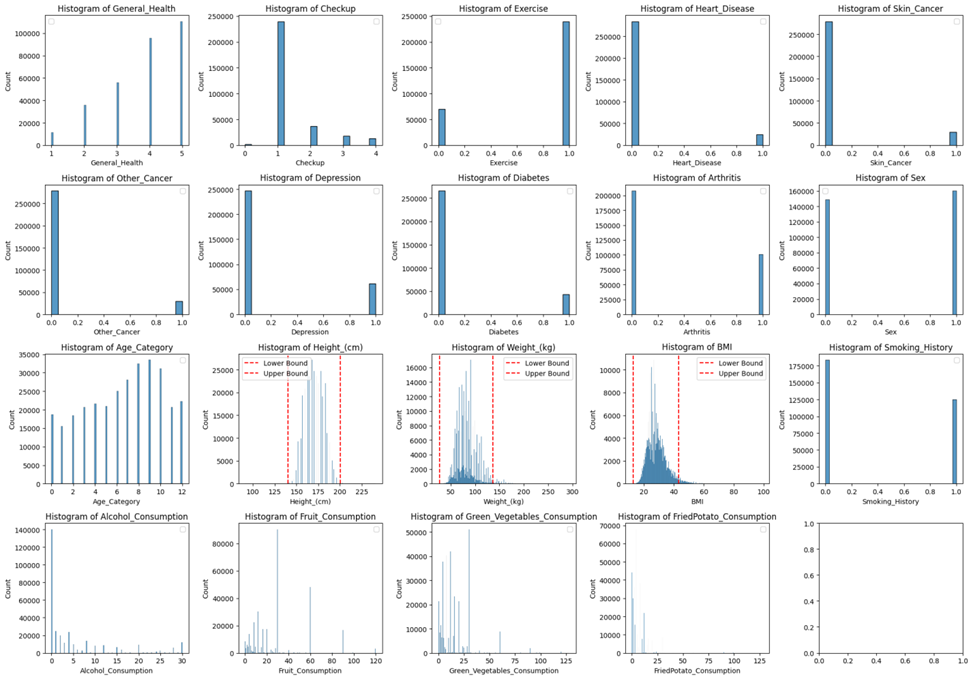


-Thực hiện encoding giá trị cột Diabetes

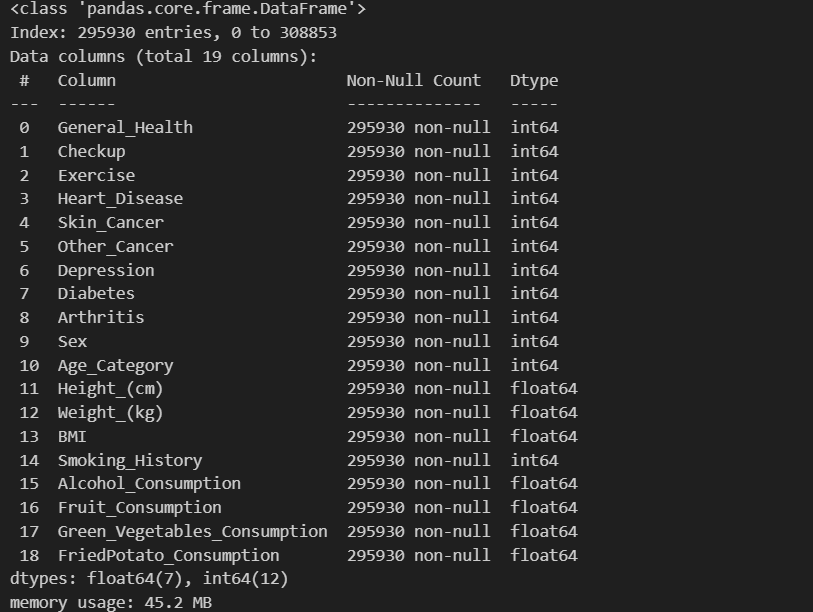


#### 6.1.1.2. Hisplots và thực hiện loại bỏ các outlier

Bằng cách trực quan hóa phân phối của từng cột dữ liệu, chúng ta có thể thấy các outlier và loại bỏ chúng



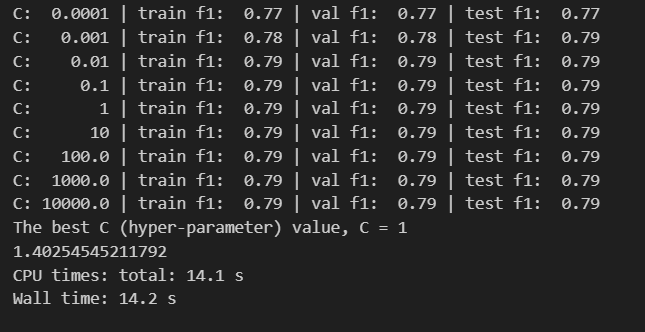
Sau khi trực quan hóa ta thấy có một vài giá trị ngoại lệ trong các cột Height, Weight và BMI mà chúng ta cần loại bỏ.

 Sau khi loại bỏ các outlier thì từ 308854 dòng còn 295930 dòng

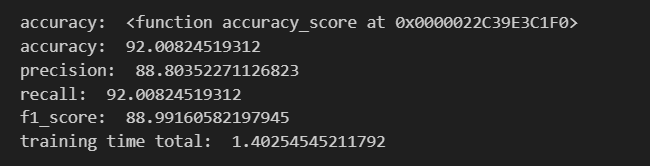
### **6.1.2.Logistic**

Tiến hành chia dữ liệu đã được xử lí thành 2 tập train\_val và test theo tỉ lệ 80/20 sau đó chia tiếp hành chia tiếp tập train\_val thành 2 tập train và test theo tỉ lệ 75/25

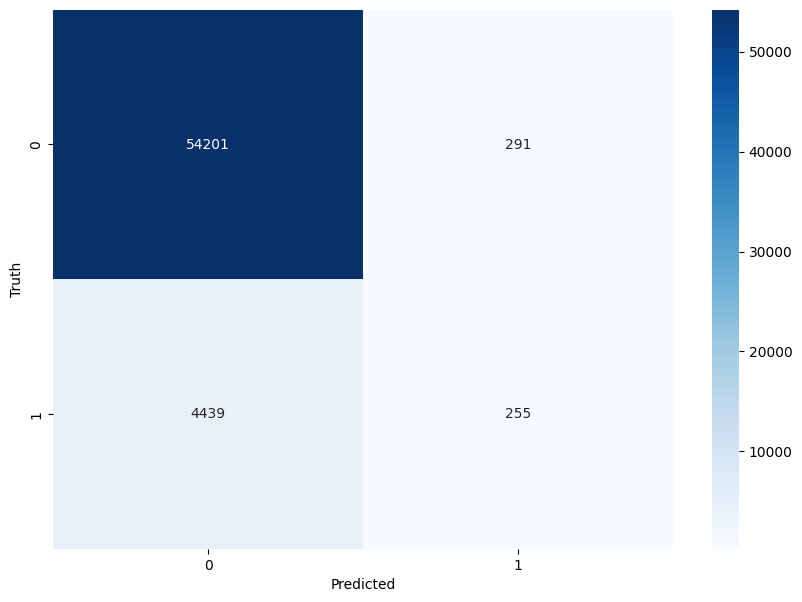
Dựa vào các tập train, val ,test đã được chia tiến hành xây dựng mô hình Logistic với các hyper-parameter khác nhau để tìm best\_C (Best\_C được chọn có val\_f1 lớn nhất) cho ra kết quả:

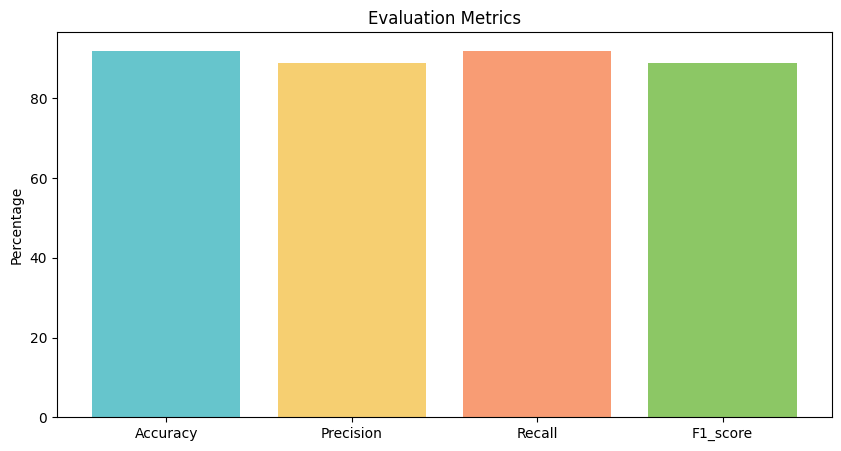


Tiến hành đánh giá hiệu suất mô hình :



Hình dưới là một confusion matrix, qua confusion matrix này ta thấy lớp dự đoán không bệnh của mô hình khá cao (54201) trong khi dự đoán có bệnh lại thấp hơn (255), tỉ lệ này chệnh nhau nhiều cho thấy sự mất cân bằng trong tập dữ liệu vì vậy cần dùng những độ đo thích hợp cho việc biểu diễn độ chính xác của thuật toán.

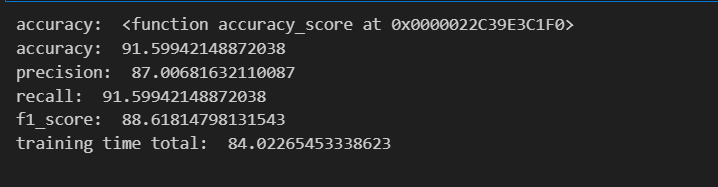
 Theo hình dưới, Accuracy trong trường hợp này cao nhưng không phản ánh đúng được độ chính xác của dữ liệu, vì vậy chúng ta dùng các độ đo như recall, precision, f1-score, và các kết quả cho ra đều cao cho thấy khả năng mô hình dự đoán chính xác.



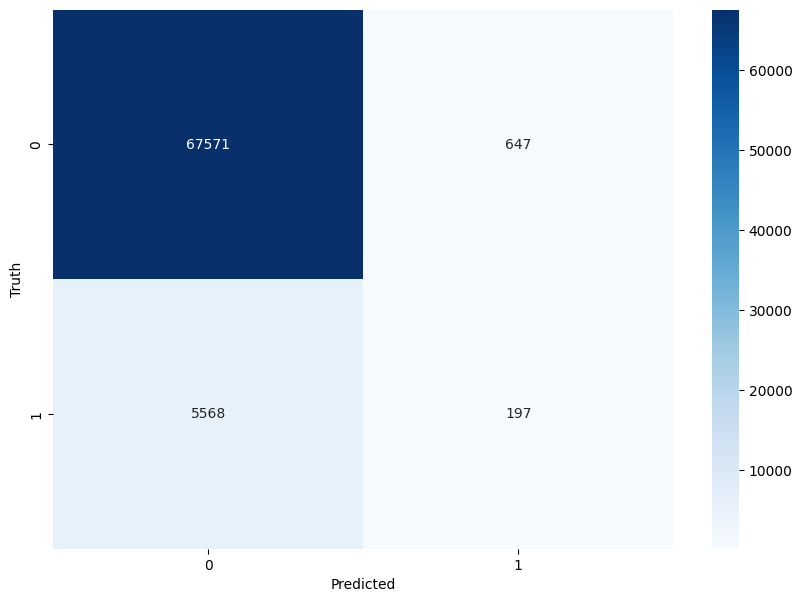
### **6.1.3. K-NN**

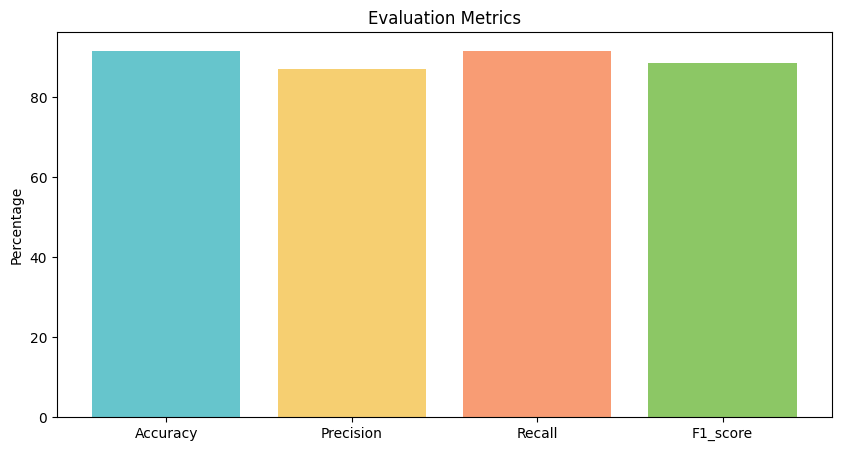
Tiến hành chia dữ liệu đã được xử lí thành 2 tập train và test theo tỉ lệ 80/20 sau đó tiền hành xây dựng mô hình

* Tiến hành đánh giá hiệu suất mô hình



Hình dưới là một confusion matrix, qua confusion matrix này ta thấy lớp dự đoán không bệnh của mô hình khá cao (67571) trong khi dự đoán có bệnh lại thấp hơn (197), tỉ lệ này chệnh nhau nhiều cho thấy sự mất cân bằng trong tập dữ liệu vì vậy cần dùng những độ đo thích hợp cho việc biểu diễn độ chính xác của thuật toán.

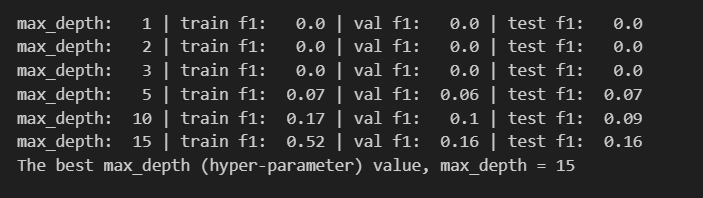
Theo hình dưới, Accuracy trong trường hợp này cao nhưng không phản ánh đúng được độ chính xác của dữ liệu, vì vậy chúng ta dùng các độ đo như recall, precision, f1-score, và các kết quả cho ra đều cao cho thấy khả năng mô hình dự đoán chính xác.

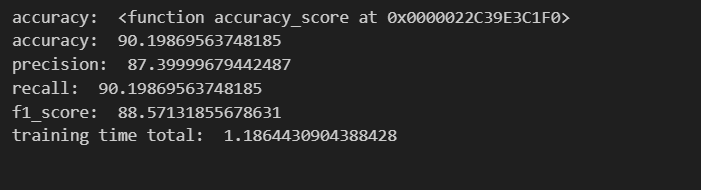


### **6.1.4.Decision Tree**

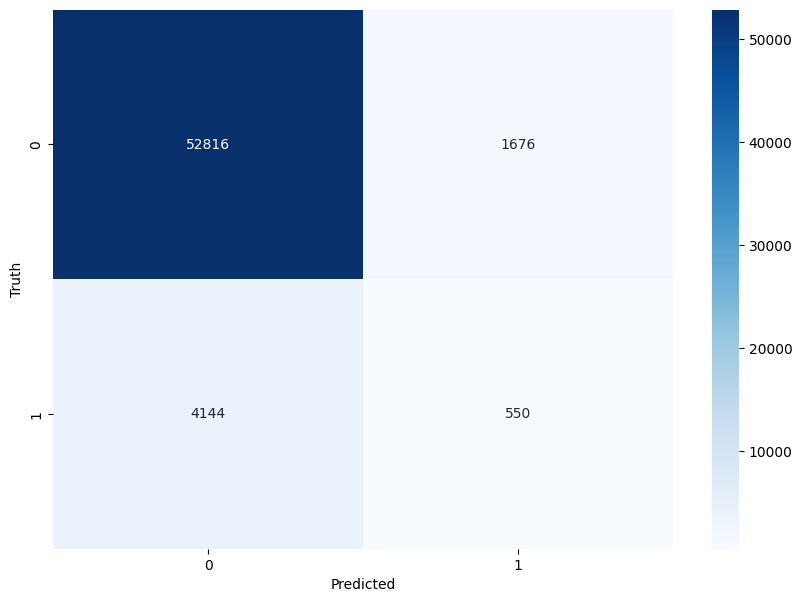
Tiến hành chia dữ liệu đã được xử lí thành 2 tập train\_val và test theo tỉ lệ 80/20 sau đó chia tiếp hành chia tiếp tập train\_val thành 2 tập train và test theo tỉ lệ 75/25

Dựa vào các tập train, val ,test đã được chia tiến hành xây dựng mô hình Decision Tree với các depth khác nhau để tìm best\_max\_depth (Best\_max\_depth được chọn có val\_f1 lớn nhất) cho ra kết quả:

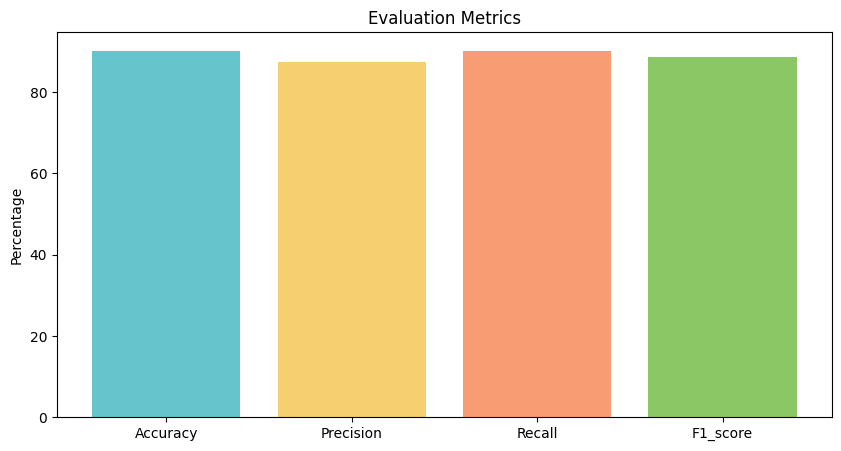
Tiến hành đánh giá hiệu suất mô hình :



Hình dưới là một confusion matrix, qua confusion metric này ta thấy lớp dự đoán không bệnh của mô hình khá cao (52816) trong khi dự đoán có bệnh lại thấp hơn (550), tỉ lệ này chệnh nhau nhiều cho thấy sự mất cân bằng trong tập dữ liệu vì vậy cần dùng những độ đo thích hợp cho việc biểu diễn độ chính xác của thuật toán.



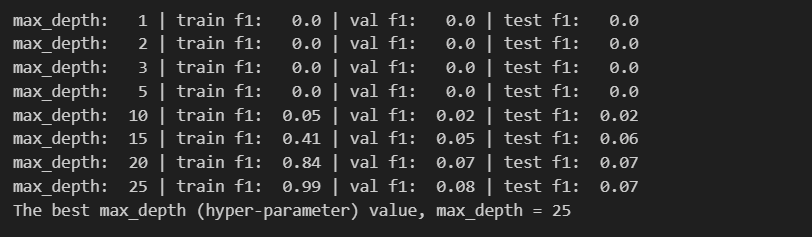
Theo hình dưới, Accuracy trong trường hợp này cao nhưng không phản ánh đúng được độ chính xác của dữ liệu, vì vậy chúng ta dùng các độ đo như recall,pre,f1, và các kết quả cho ra đều cao => mô hình dự đoán chính xác



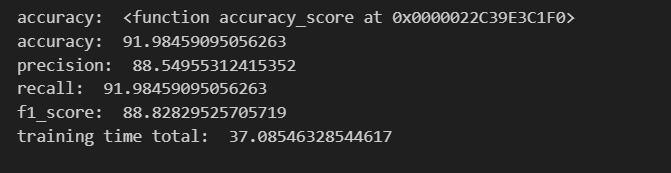
### **6.1.5. Random Forest**

Tiến hành chia dữ liệu đã được xử lí thành 2 tập train\_val và test theo tỉ lệ 80/20 sau đó chia tiếp hành chia tiếp tập train\_val thành 2 tập train và test theo tỉ lệ 75/25.

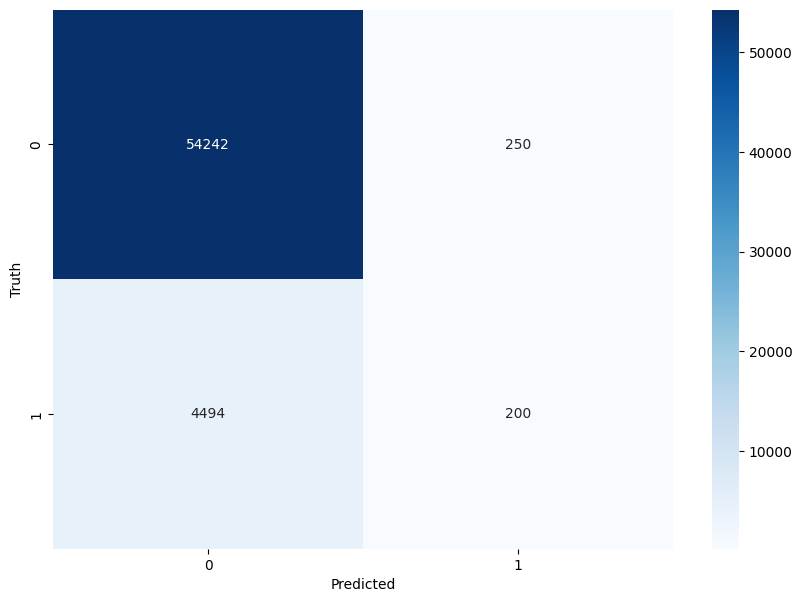
Dựa vào các tập train, val ,test đã được chia tiến hành xây dựng mô hình Random Forest với các depth khác nhau để tìm best\_max\_depth (Best\_max\_depth được chọn có val\_f1 lớn nhất) cho ra kết quả:



Tiến hành đánh giá hiệu suất mô hình :

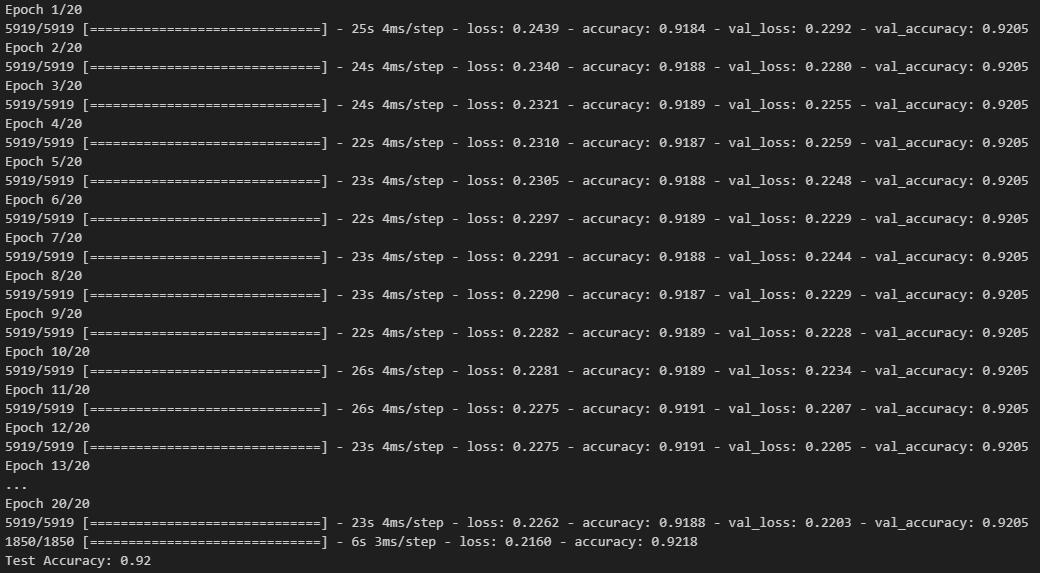


Hình dưới là một confusion matrix, qua confusion metric này ta thấy lớp dự đoán không bệnh của mô hình khá cao (54242) trong khi dự đoán có bệnh lại thấp hơn ( 195), tỉ lệ này chệnh nhau nhiều cho thấy sự mất cân bằng trong tập dữ liệu vì vậy cần dùng những độ đo thích hợp cho việc biểu diễn độ chính xác của thuật toán.



Theo hình dưới, Accuracy trong trường hợp này cao nhưng không phản ánh đúng được độ chính xác của dữ liệu, vì vậy chúng ta dùng các độ đo như recall, precision, f1-score, và các kết quả cho ra đều cao cho thấy khả năng mô hình dự đoán chính xác

### **6.1.6. ANN**



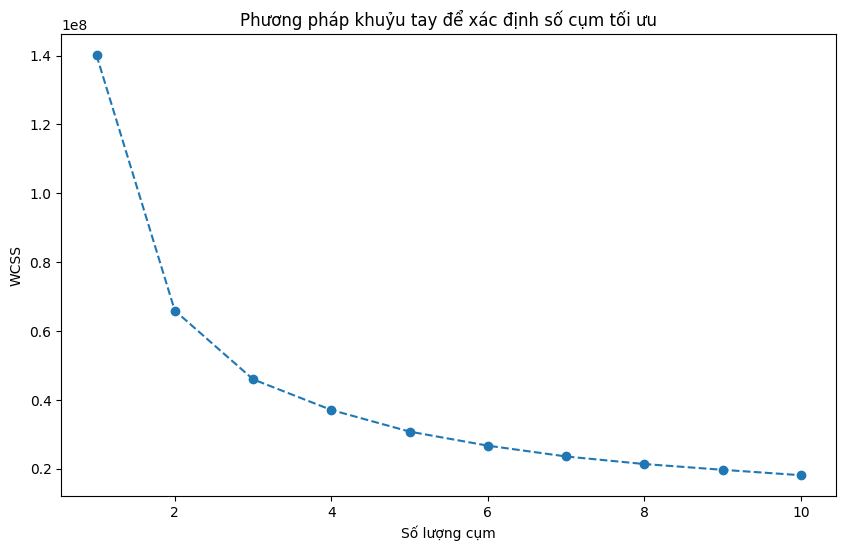
Mô hình ANN này có hiệu suất rất tốt, với độ chính xác đạt 92.05% trên tập dữ liệu, cho thấy mô hình đã học được các đặc điểm phức tạp của mã captcha và có khả năng phân loại chúng một cách chính xác. Độ chính xác cao trên cả hai tập dữ liệu cho thấy mô hình có khả năng học và nhận diện các đặc điểm quan trọng của mã captcha hiệu quả.

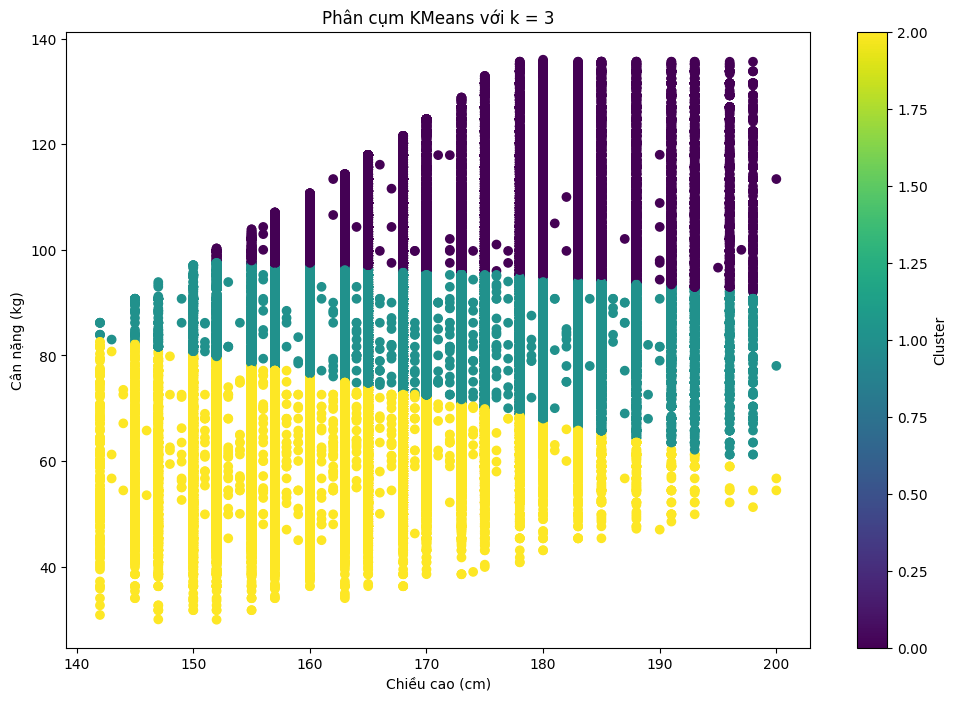
Hơn nữa, mô hình hội tụ nhanh chóng, chỉ sau 20 epoch đã đạt được độ chính xác 92.05% trên tập dữ liệu đào tạo, chứng tỏ khả năng học hỏi hiệu quả. Điều này cho thấy mô hình có thể phân loại chính xác, khẳng định hiệu suất và khả năng ứng dụng thực tiễn cao.

### **6.1.7.K-means**

-Chọn ra các thuộc tính để phân loại bao gồm: General\_Health, Heart\_Disease, Age\_Category, Height\_(cm), Weight\_(kg), BMI

-Vẽ biểu đồ khuỷu tay để xác định số cụm tối ưu

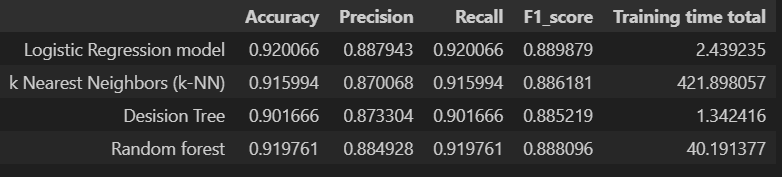
Chọn k=3 để phân cụm

Nhận xét: Cụm có cân nặng cao hơn có nguy cơ mắc bệnh tim mạch nhiều hơn. Từ đây, bằng cách duy trì cân nặng hợp lý và sống lành mạnh, có thể giảm đáng kể nguy cơ mắc các bệnh tim mạch và cải thiện chất lượng cuộc sống.

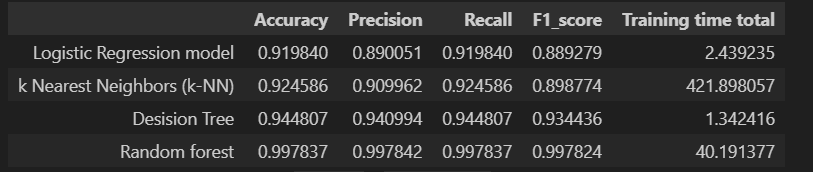
## **6.2. Đánh giá**

### **6.2.1. So sánh kết quả phân loại**

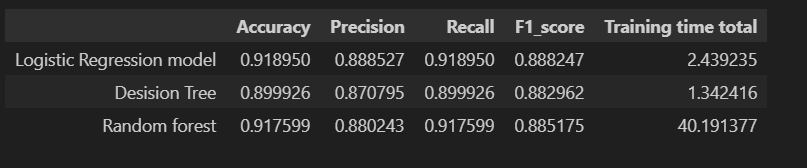
-Tập test



-Tập train:



-Tập val



Mô hình Logistic Regression có độ chính xác cao nhất (0,920066), nhưng độ chính xác (0,887943) và độ thu hồi (0,920066) tương đối thấp. Điểm F1 của mô hình này là 0,889879 và thời gian đào tạo tổng cộng là 2,439235.

Mô hình k Nearest Neighbors (k-NN) có độ chính xác (0,915994) và độ thu hồi (0,915994) tương đối cao, nhưng điểm F1 (0,886181) thấp hơn so với mô hình Logistic Regression. Thời gian đào tạo tổng cộng của mô hình này là 421,898057.

Mô hình Decision Tree có độ chính xác (0,901666), độ thu hồi (0,901666) và điểm F1 (0,885219) tương đối thấp. Thời gian đào tạo tổng cộng của mô hình này là 1,342416.

Mô hình Random Forest có độ chính xác (0,919761), độ chính xác (0,884928) và độ thu hồi (0,919761) tương đối cao. Điểm F1 của mô hình này là 0,888096 và thời gian đào tạo tổng cộng là 40,191377.

Qua kết quả trên ta thấy mô hình có khả năng bị overffiting do kết quả của tập train cao mà tập test lại thấp hơn. Tuy nhiên tất cả các thuật toán đều cho ra kết quả có độ chính xác cao. Tuy nhiên accurracy ko phản ánh rõ được độ chính xác của mô hình

### **6.1.2. Đánh giá mô hình ANN**

A black background with white text

Description automatically generated

Mô hình đạt độ chính xác 92.31% trên tập huấn luyện và 92.00% trên tập kiểm tra. Độ chính xác trên tập huấn luyện và kiểm tra không quá chênh lệch, điều này cho thấy mô hình không bị overfitting (quá khớp) hoặc underfitting (chưa khớp đủ). Độ chính xác cao cho thấy mô hình có khả năng phân loại chính xác hầu hết các trường hợp, cả bệnh nhân mắc và không mắc bệnh tim.

# **KẾT LUẬN**

Đề tài về "Xây dựng các mô hình phân tích dự đoán nguy cơ bệnh tim mạch dựa trên thói quen sinh hoạt, tiền sử bệnh" là một bước quan trọng trong việc ứng dụng công nghệ thông tin và khoa học dữ liệu vào lĩnh vực y tế. Trong quá trình nghiên cứu, chúng tôi đã sử dụng một loạt các thuật toán học máy như KNN, Random Forest, Logistic Regression, ANN, K-means và Decision Tree để phân tích và dự đoán nguy cơ bệnh tim mạch.

Kết quả cho thấy các mô hình đã được xây dựng có khả năng dự đoán nguy cơ bệnh tim mạch một cách chính xác và đáng tin cậy. Đặc biệt, mô hình Random Forest và Logistic Regression đã cho ra kết quả tốt nhất, với độ chính xác cao và khả năng tổng quát hóa tốt trên dữ liệu mới.

Tuy nhiên, cũng cần lưu ý rằng việc dự đoán nguy cơ bệnh tim mạch không chỉ phụ thuộc vào dữ liệu về thói quen sinh hoạt và tiền sử bệnh, mà còn phụ thuộc vào nhiều yếu tố khác như di truyền, môi trường sống và tình trạng sức khỏe tổng thể của mỗi cá nhân.

Với sự phát triển của công nghệ và sự tiến bộ trong lĩnh vực khoa học dữ liệu, chúng ta có thể hy vọng rằng các mô hình dự đoán nguy cơ bệnh tim mạch sẽ ngày càng được cải thiện và áp dụng rộng rãi trong thực tiễn y tế, giúp phát hiện sớm và phòng tránh bệnh tim mạch một cách hiệu quả.

# TÀI LIỆU THAM KHẢO

**<https://bktt.vn/Ph%C3%A2n_c%E1%BB%A5m_d%E1%BB%AF_li%E1%BB%87u>**

**<https://viblo.asia/p/data-mining-khai-pha-du-lieu-data-science-series-1Je5EAX45nL>**

**<https://www.toponseek.com/blogs/data-mining-la-gi/>**

<https://tapchi.humg.edu.vn/images/paper/11.%20Nguyen%20T%20Huu%20Phuong.pdf>

<https://jgac.vn/journal/article/view/344/331>

<https://trituenhantao.io/machine-learning-co-ban/bai-6-logistic-regression-hoi-quy-logistic/>

<https://tapchi.humg.edu.vn/images/paper/11.%20Nguyen%20T%20Huu%20Phuong.pdf>