TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**MÔN KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI:**

**Sử dụng cây quyết định để dự đoán thời tiết**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: NGUYỄN DUY LONG**  **: NGUYỄN TRUNG KIÊN** | | **Giảng viên hướng dẫn** | **: VŨ VĂN ĐỊNH** | | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | | **Lớp** | **: D14CNPM1** | | | **Khóa** | **: 2018-2023** | | |  |
|  |  |

***Hà Nội, tháng .. năm ....***

**PHIẾU CHẤM ĐIỂM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên sinh viên** | **Nội dung thực hiện** | **Điểm** | **Chữ ký** |
| 1 | Nguyễn Duy Long |  |  |  |
| 2 | Nguyễn Trung Kiên |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Họ và tên giảng viên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

**LỜI MỞ ĐẦU**

     Trong nhiều năm qua, cùng với sự phát triển của công nghệ thông tin và ứng dụng của công nghệ thông tin trong nhiều lĩnh vực của đời sống xã hội, thì lượng dữ liệu được các cơ quan thu thập và lưu trữ ngày một nhiều lên. Người ta lưu trữ những dữ liệu này vì cho rằng nó ẩn chứa những giá trị nhất định nào đó. Tuy nhiên theo thống kê thì chỉ có một lượng nhỏ của những dữ liệu này (khoảng từ 5% đến 10%) là luôn được phân tích, số còn lại họ không biết sẽ phải làm gì và có thể làm gì với những dữ liệu này, nhưng họ vẫn tiếp tục thu thập và lưu trữ vì hy vọng những dữ liệu này sẽ cung cấp cho họ những thông tin quý giá một cách nhanh chóng để đưa ra những quyết định kịp thời vào một lúc nào đó.

Chính vì vậy, các phương pháp quản trị và khai thác cơ sở dữ liệu truyền thống ngày càng không đáp ứng được thực tế đã làm phát triển một khuynh hướng kỹ thuật mới đó là Kỹ thuật phát hiện tri thức và khai phá dữ liệu (KDD - Knowledge Discovery and Data Mining). Kỹ thuật phát hiện tri thức và khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau trên thế giới, tại Việt Nam kỹ thuật này còn tương đối mới mẻ tuy nhiên cũng đang được nghiên cứu và bắt đầu đưa vào một số ứng dụng thực tế.

Vì vậy, hiện nay ở nước ta vấn đề phát hiện tri thức và khai phá dữ liệu đang thu hút được sự quan tâm của nhiều người và nhiều công ty phát triển ứng dụng công nghệ thông tin.

# CHƯƠNG 1: TỔNG QUAN

## 1.1. Giới thiệu về khai phá dữ liệu

Khai phá dữ liệu (data mining): Là quá trình tính toán để tìm ra các mẫu trong các bộ dữ liệu lớn liên quan đến các phương pháp tại giao điểm của máy học, thống kê và các hệ thống cơ sở dữ liệu. Đây là một lĩnh vực liên ngành của khoa học máy tính… Mục tiêu tổng thể của quá trình khai thác dữ liệu là trích xuất thông tin từ một bộ dữ liệu và chuyển nó thành một cấu trúc dễ hiểu để sử dụng tiếp. Ngoài bước phân tích thô, nó còn liên quan tới cơ sở dữ liệu và các khía cạnh quản lý dữ liệu, xử lý dữ liệu trước, suy xét mô hình và suy luận thống kê, các thước đo thú vị, các cân nhắc phức tạp, xuất kết quả về các cấu trúc được phát hiện, hiện hình hóa và cập nhật trực tuyến. Khai thác dữ liệu là bước phân tích của quá trình "khám phá kiến thức trong cơ sở dữ liệu" hoặc KDD.



Hình 1.1: Data mining (khai pá dữ liệu)

### **1.1.1. Diễn giải về khai phá dữ liệu**

Khai phá dữ liệu là một bước của quá trình [khai thác tri thức](https://vi.wikipedia.org/w/index.php?title=Khai_th%C3%A1c_tri_th%E1%BB%A9c&action=edit&redlink=1) (*Knowledge Discovery Proces*s), bao gồm:

- Xác định vấn đề và không gian dữ liệu để giải quyết vấn đề (*Problem understanding and data understanding*).

- Chuẩn bị dữ liệu (*Data preparation*), bao gồm các quá trình làm sạch dữ liệu (*data cleaning*), tích hợp dữ liệu (*data integration*), chọn dữ liệu (*data selection*), biến đổi dữ liệu (*data transformation*).

- Khai thác dữ liệu (*Data mining*): xác định *nhiệm vụ khai thác dữ liệu* và lựa chọn *kỹ thuật khai thác dữ liệu*. Kết quả cho ta một *nguồn tri thức thô*.

- Đánh giá (*Evaluation*): dựa trên một số tiêu chí tiến hành *kiểm tra* và *lọc* nguồn tri thức thu được.

- Triển khai (*Deployment*).

Quá trình khai thác tri thức không chỉ là một quá trình tuần tự từ bước đầu tiên đến bước cuối cùng mà là một quá trình lặp và có quay trở lại các bước đã qua.

### **1.1.2. Các phương thức khai phá dữ liệu**

- Phân loại (Classification): Là phương pháp dự báo, cho phép phân loại một đối tượng vào một hoặc một số lớp cho trước.

- Hồi qui (Regression): Khám phá chức năng học dự đoán, ánh xạ một mục dữ liệu thành biến dự đoán giá trị thực.

- Phân nhóm (Clustering): Một nhiệm vụ mô tả phổ biến trong đó người ta tìm cách xác định một tập hợp hữu hạn các cụm để mô tả dữ liệu.

- Tổng hợp (Summarization): Một nhiệm vụ mô tả bổ sung liên quan đến phương pháp cho việc tìm kiếm một mô tả nhỏ gọn cho một bộ (hoặc tập hợp con) của dữ liệu.

- Mô hình ràng buộc (Dependency modeling): Tìm mô hình cục bộ mô tả các phụ thuộc đáng kể giữa các biến hoặc giữa các giá trị của một tính năng trong tập dữ liệu hoặc trong một phần của tập dữ liệu.

- Dò tìm biến đổi và độ lệch (Change and Deviation Dectection): Khám phá những thay đổi quan trọng nhất trong bộ dữ liệu.

### **1.1.3 . Một số tính năng nổi bật của Data mining**

Đưa ra các mẫu dự đoán theo xu hướng trong dữ liệu.

- Tính toán kết quả đã phân tích.

- Tạo thông tin phản hồi để phân tích.

- Tập trung phân tích các dữ liệu lớn.

- Phân chia cụm dữ liệu một cách trực quan.

### **1.1.4. Các công cụ khai phá dữ liệu**

RapidMiner: Công cụ đầu tiên phải kể tới đó là RapidMiner. Đây là công cụ khai phá dữ liệu khá phổ biến hiện nay. Được viết trên nền tảng JAVA nhưng không yêu cầu mã hóa để vận hành. Ngoài ra, nó còn cung cấp các chức năng khai thác dữ liệu khác nhau như tiền xử lý dữ liệu, biểu diễn dữ liệu, lọc, phân cụm...

Weka: Công cụ được cho ra đời tại Đại học Wichita là một phần mềm khai thác dữ liệu mã nguồn mở. Tương tự như RapidMiner, công cụ này không yêu cầu mã hóa và sử dụng GUI đơn giản.

Sử dụng Weka, người dùng có thể gọi trực tiếp các thuật toán học máy hoặc nhập chúng bằng mã Java. Weka được trang bị đa dạng chức năng như trực quan hóa, tiền xử lý, phân loại, phân cụm...

Knime: Với khả năng hoạt động vô cùng mạnh mẽ tích hợp nhiều thành phần khác nhau của học máy và khai phá dữ liệu để cung cấp một nền tảng. KNime hỗ trợ người dùng rất nhiều trong việc xử lý và phân tích dữ liệu, trích xuất, chuyển đổi và tải dữ liệu.

Apache Mahout: Từ nền tảng Big Data Hadoop, người ta đã cho cho ra đời thêm Apache Mahout với mục đích giải quyết nhu cầu ngày càng tăng về khai phá dữ liệu và hoạt động phân tích trong Hadoop. Nó được trang bị nhiều chức năng học máy khác nhau như phân loại, hồi quy, phân cụm...

Oracle Data Mining: Khi sử dụng Oracle Data Mining. nó cho phép người dùng thực hiện khai phá dữ liệu trên cơ sở dữ liệu SQL để trích xuất các khung hình và biểu đồ. Các phân tích sẽ hiển thị một cách trực quan giúp người dùng dễ dàng đưa ra dự đoán cho kế hoạch tương lai.

TeraData: TeraData cung cấp dịch vụ kho chứa các công cụ khai phá dữ liệu. Nhờ khả năng thông minh được trang bị, công cụ có thể dựa trên tần suất sử dụng dữ liệu của người dùng và thực hiện việc cho phép truy cập nhanh hay chậm.

Với một dữ liệu bạn thường xuyên cần sử dụng, TeraData sẽ cho phép truy cập nhanh hơn là một dữ liệu ít được sử dụng. Đối với dữ liệu, nhập kho là một yêu cầu cần thiết.

Orange: Công cụ được lập trình bằng Python với giao diện trực quan và tương tác dễ dàng. Phần mềm Orange được biết đến bởi việc tích hợp các công cụ khai phá dữ liệu và học máy thông minh, đơn giản.

### **1.1.5. Quy trình khai phá dữ liệu**

Bước 1: Làm sạch dữ liệu – Trong bước này, dữ liệu được làm sạch sao cho không có tạp âm hay bất thường trong dữ liệu.

Bước 2: Tích hợp dữ liệu – Trong quá trình tích hợp dữ liệu, nhiều nguồn dữ liệu sẽ kết hợp lại thành một.

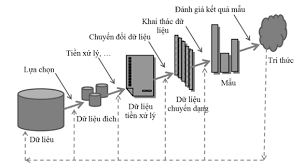
Bước 3: Lựa chọn dữ liệu – Trong bước này, dữ liệu được trích xuất từ cơ sở dữ liệu.

Bước 4: Chuyển đổi dữ liệu – Trong bước này, dữ liệu sẽ được chuyển đổi để thực hiện phân tích tóm tắt cũng như các hoạt động tổng hợp.

Bước 5: Khai phá dữ liệu – Trong bước này, chúng tôi trích xuất dữ liệu hữu ích từ nhóm dữ liệu hiện có.

Bước 6: Đánh giá mẫu – Chúng tôi phân tích một số mẫu có trong dữ liệu.

Bước 7: Trình bày thông tin – Trong bước cuối cùng, thông tin sẽ được thể hiện dưới dạng cây, bảng, biểu đồ và ma trận.



Hình 1.2: Quy trình khai phá dữ liệu

### **1.1.6. Ứng dụng của khai phá dữ liệu**

- Trong kinh doanh (business)

- Trong tài chính (finance) và tiếp thị bán hàng (sales marketing)

- Trong thương mại (commerce) và ngân hàng (bank)

- Trong bảo hiểm (insurance)

- Trong khoa học (science) và y sinh học (biomedicine)

- Trong điều khiển (control) và viễn thông (telecommunication)

## **1.2. Tiền xử lý**

Trong qui trình khai phá dữ liệu, công việc xử lý dữ liệu trước khi đưa vào các mô hình là rất cần thiết, bước này làm cho dữ liệu có được ban đầu qua thu thập dữ liệu (gọi là dữ liệu gốc original data) có thể áp dụng được (thích hợp) với các mô hình khai phá dữ liệu (data mining model) cụ thể. Các công việc cụ thể của tiền xử lý dữ liệu bao gồm những công việc như:

- Filtering Attributes: Chọn các thuộc tính phù hợp với mô hình

- Filtering samples: Lọc các mẫu (instances, patterns) dữ liệu cho mô hình

- Clean data: Làm sạch dữ liệu như xóa bỏ các dữ liệu bất thường

- Transformation: Chuyển đổi dữ liệu cho phù hợp với các mô hình như chuyển đổi dữ liệu từ numeric qua nomial hay ordinal

- Discretization (rời rạc hóa dữ liệu): Nếu bạn có dữ liệu liên tục nhưng một vài mô hình chỉ áp dụng cho các dữ liệu rời rạc (như luật kết hợp chẳn hạn) thì bạn phải thực hiện việc rời rạc hóa dữ liệu.

## **1.3. Giới thiệu bài toán**

**-** Bài toán:*“ Dự đoán mưa sử dụng thuật toán Phân lớp ”*

**-** Dữ liệu của bài toán: “ *Dự đoán mưa* ” đây là tập dữ liệu về ngày mưa ở Áo. Có 23 thuộc tính lần lượt là: “Date, Location, MinTemp, MaxTemp, Rainfall, Evaporatic, Sunshine, WindGustDir,WindGustSpeed, WindDir9am, WindDir3pm, 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am', 'Temp3pm', 'RainToday', 'RainTomorrow'”.

**CHƯƠNG 2: GIẢI QUYẾT BÀI TOÁN CÂY QUYẾT ĐỊNH**

**2.1. Tổng quan về thuật toán cây quyết đinh**

**2.1.1. Giới thiệu thuật toán**

Trong lĩnh vực [máy học](https://vi.wikipedia.org/wiki/M%C3%A1y_h%E1%BB%8Dc), cây quyết định là một kiểu mô hình dự báo (*predictive model*), nghĩa là một ánh xạ từ các quan sát về một sự vật/hiện tượng tới các kết luận về giá trị mục tiêu của sự vật/hiện tượng. Mỗi một nút trong (*internal node*) tương ứng với một biến; đường nối giữa nó với nút con của nó thể hiện một giá trị cụ thể cho biến đó. Mỗi nút lá đại diện cho giá trị dự đoán của biến mục tiêu, cho trước các giá trị của các biến được biểu diễn bởi đường đi từ nút gốc tới nút lá đó. Kỹ thuật học máy dùng trong cây quyết định được gọi là học bằng cây quyết định, hay chỉ gọi với cái tên ngắn gọn là cây quyết định.

Học bằng cây quyết định cũng là một phương pháp thông dụng trong [khai phá dữ liệu](https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u). Khi đó, cây quyết định mô tả một cấu trúc cây, trong đó, các lá đại diện cho các phân loại còn cành đại diện cho các kết hợp của các thuộc tính dẫn tới phân loại đó[1]. Một cây quyết định có thể được học bằng cách chia [tập hợp](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p) nguồn thành các tập con dựa theo một kiểm tra giá trị thuộc tính [1]. Quá trình này được lặp lại một cách đệ quy cho mỗi tập con dẫn xuất. Quá trình [đệ quy](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%87_quy) hoàn thành khi không thể tiếp tục thực hiện việc chia tách được nữa, hay khi một phân loại đơn có thể áp dụng cho từng phần tử của tập con dẫn xuất. Một bộ phân loại [rừng ngẫu nhiên](https://vi.wikipedia.org/w/index.php?title=R%E1%BB%ABng_ng%E1%BA%ABu_nhi%C3%AAn&action=edit&redlink=1) (*random forest*) sử dụng một số cây quyết định để có thể cải thiện tỉ lệ phân loại.

Cây quyết định cũng là một phương tiện có tính mô tả dành cho việc tính toán các [xác suất có điều kiện](https://vi.wikipedia.org/wiki/X%C3%A1c_su%E1%BA%A5t_c%C3%B3_%C4%91i%E1%BB%81u_ki%E1%BB%87n).

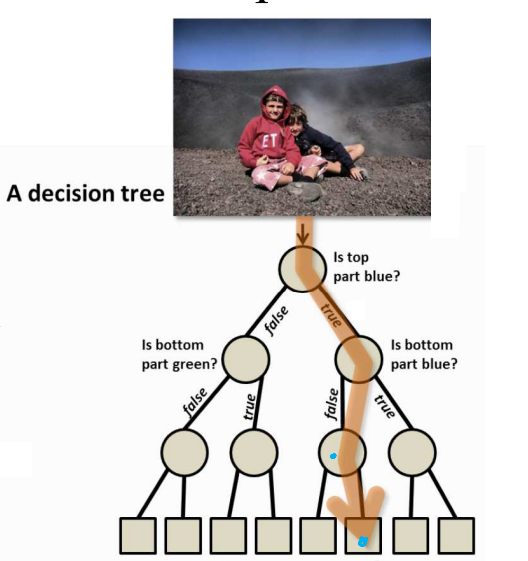
Cây quyết định có thể được mô tả như là sự kết hợp của các kỹ thuật toán học và tính toán nhằm hỗ trợ việc mô tả, phân loại và tổng quát hóa một tập dữ liệu cho trước.

Dữ liệu được cho dưới dạng các bản ghi có dạng:

*(x, y) = (x1, x2, x3..., xk, y)*

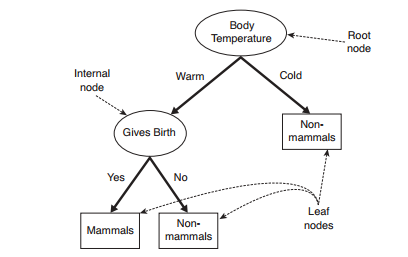
Biến phụ thuộc (*dependant variable*) *y* là biến mà chúng ta cần tìm hiểu, phân loại hay tổng quát hóa. *x1*, *x2*, *x3*... là các biến sẽ giúp ta thực hiện công việc đó.

Tóm lại, 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 (series of rules). Khi cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các đối tượng chưa biết (unseen data).



**2.2.2. Cấu trúc của thuật toán**

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. Ví dụ câu hỏi ở node đầu tiên yêu cầu câu trả lời là “yes” hoặc là “no” thì sẽ có 1 node con chịu trách nhiệm cho phản hồi là “yes”, 1 node là “no”.



**2.2. Các thuật toán liên quan đến cây quyết định**

**2.2.1. ID3 (Iterative Dichotomiser 3) Algorithm**

Thuật toán ID3 có thể được tóm tắt như sau:

* Lấy tất cả các thuộc tính chưa được sử dụng và đếm entropy liên quan đến mẫu thử của các thuộc tính đó
* Chọn thuộc tính có entropy lớn nhất
* Nối node với thuộc tính đó

**2.2.2. ID3 (Examples, Target\_Attribute, Attributes)**

Begin:

* Tạo node gốc
* Nếu tất cả các examples (ví dụ) đều nằm cùng một lớp là dương thì return một nút lá được gán nhãn bởi lớp đó (lớp +).
* Nếu tất cả các examples (ví dụ) đều nằm cùng một lớp là âm thì return một nút lá được gán nhãn bởi lớp đó (lớp -).
* Nếu tập thuộc tính Attributes là rỗng thì return Cây quyết định có nút Root được gắn với nhãn lớp bằng giá trị chung nhất trong tập thuộc tính ở tập ví dụ (Examples).

      Else begin:

* Chọn thuộc tính A – Thuộc tính trong tập Attributes có khả năng phân loại “tốt nhất” đối với Examples
* Thuộc tính kiểm tra cho nút Root ← A và lấy nó làm gốc cho cây hiện tại
* Với mỗi giá trị vi của A:
  + Bổ sung một nhánh cây mới nằm phía dưới node gốc, tương ứng với trường hợp A = vi
  + Xác định Examples(vi) sao cho tập con của Examples có giá trị vi cho A
  + Nếu Examples(vi) rỗng

*# Tạo một nút lá được gắn nhãn = giá trị đích phổ biến nhất trong Examples. Sau đó gắn nút lá này vào nhánh cây mới vừa tạo.*

* Ngược lại, gắn vào nhánh cây mới vừa tạo một cây con sinh ra bởi ID3 (Examples(vi), Target\_Attribute, Attributes – {A})
* Kết thúc
* Trả về giá trị node gốc

**2.2.3. C4.5 Algorithm**

C4.5 là thuật toán phân lớp dữ liệu dựa trên cây quyết định hiệu quả và phổ biến trong những ứng dụng khai phá cơ sở dữ liệu có kích thước nhỏ. C4.5 xây dựng cây quyết định từ tập training data tương tự như ID3.

Tập training data S gồm các mẫu đã được phân  loại sẵn s1, s2,… Mỗi mẫu si = x1,x2,… với x1, x2 là 1 vector biểu diễn cho thuộc tính hoặc đặc điểm của mẫu. Một tập vector C = c1,c2,… với c1, c2 biểu diễn cho mỗi lớp mà mỗi mẫu thuộc về.

Với mỗi lớp của cây, C4.5 chọn một thuộc tính của dữ liệu mà phân chia tập các mẫu thành nhiều tập con, được nâng cao chất lượng một cách hiệu quả nhất. Tiêu chuẩn của nó là thu được **information gain** (sự khác biệt về entropy) – kết quả từ việc chọn một thuộc tính cho việc chia tách dữ liệu. Quyết định đưa ra dựa trên tập thuộc tính với information gain được chuẩn hóa cao nhất. Thuật toán C4.5 sau đó sẽ lặp lại với các danh sách con nhỏ hơn.

Một vài trường hợp cơ bản:

* Tất cả các mẫu đều thuộc cùng một lớp. Khi điều này xảy ra, nó chỉ đơn giản tạo ra một nút lá cho cây quyết định và cho biết chọn lớp đó.
* Không có tính năng nào cung cấp bất kỳ information gain. Trong trường hợp này, C4.5 tạo ra một node quyết định (decision node) bằng cách sử dụng giá trị kì vọng của lớp đó.
* Với các trường hợp khác, C4.5 lần nữa tạo ra một node quyết định bằng cách sử dụng giá trị kì vọng của lớp đó.

Thuật toán được giải mã như sau:

* Bắt đầu:

ComputerClassFrequency(T);

* Kiểm tra các trường hợp cơ bản rồi tạo ra node quyết định

  if OneClass or FewCases

              return a leaf;

      Create a decision node N;

* Gán N.test = Thuộc tính với information gain được chuẩn hóa cao nhất

ForEach Attribute A

ComputeGain(A);

N.test=AttributeWithBestGain;

* Thuật toán C4.5 sau đó sẽ lặp lại với các danh sách con nhỏ hơn và gán những node như là các node con

if (N.test is continuous)

find Threshold;

ForEach T’ in the splitting of T

If ( T’ is Empty )

Child of N is a leaf

else

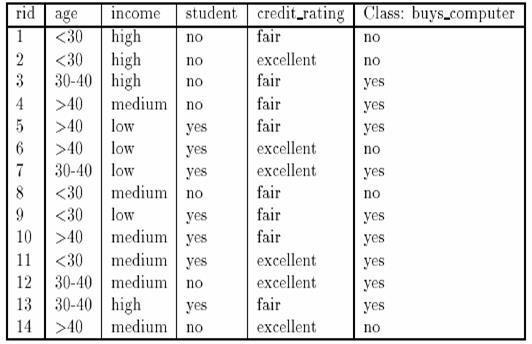
Child of N=FormTree(T’);

ComputeErrors of N;

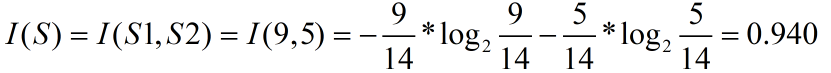
return N

Ví dụ mô tả cách tính information gain:

Với thuộc tính rời rạc:



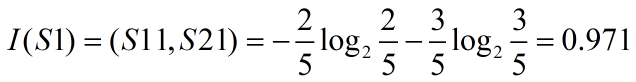
Trong tập dữ liệu trên: S1 là tập những bản ghi có giá trị phân lớp là yes, S2 là tập những bản ghi có giá trị phân lớp là no. Khi đó:



Tính G(S,A) với A lần lượt là từng thuộc tính:

A = age. Thuộc tính age đã được rời rạc hóa thành các giá trị <30, 30-40, và >40.

* với age = “<30”:

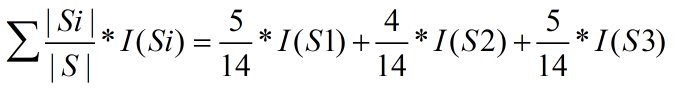


* với age = “30-40”:

https://tapit.vn/wp-content/uploads/2017/07/ct3.png

* với age = “>40”:

https://tapit.vn/wp-content/uploads/2017/07/ct4.png



https://tapit.vn/wp-content/uploads/2017/07/ct6.png

Tính tương tự với các thuộc tính khác ta được:  
   – A = income: Gain (S, income) = 0.029  
   – A = student: Gain (S, student) = 0.151  
   – A = credit\_rating: Gain ( S, credit\_rating) = 0.048  
Thuộc tính age là  thuộc tính có độ đo Information Gain lớn nhất. Do vậy age được chọn làm thuộc tính phát triển tại node đang xét.

**2.3. Ứng dụng của thuật toán**

Trong [lý thuyết quyết định](https://vi.wikipedia.org/wiki/L%C3%BD_thuy%E1%BA%BFt_quy%E1%BA%BFt_%C4%91%E1%BB%8Bnh) (chẳng hạn [quản lý rủi ro](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_r%E1%BB%A7i_ro)), một cây quyết định ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): decision tree) là một [đồ thị](https://vi.wikipedia.org/wiki/L%C3%BD_thuy%E1%BA%BFt_%C4%91%E1%BB%93_th%E1%BB%8B) của các quyết định và các hậu quả có thể của nó (bao gồm rủi ro và hao phí tài nguyên). Cây quyết định được sử dụng để xây dựng một [kế hoạch](https://vi.wikipedia.org/wiki/K%E1%BA%BF_ho%E1%BA%A1ch) nhằm đạt được [mục tiêu](https://vi.wikipedia.org/wiki/M%E1%BB%A5c_ti%C3%AAu) mong muốn. Các cây quyết định được dùng để hỗ trợ quá trình ra quyết định. Cây quyết định là một dạng đặc biệt của [cấu trúc cây](https://vi.wikipedia.org/wiki/C%E1%BA%A5u_tr%C3%BAc_c%C3%A2y).

**2.4. Ưu và nhược điểm của thuật toán**

**2.4.1. Ưu điểm:**

So với các phương pháp [khai phá dữ liệu](https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u) khác, cây quyết định là phương pháp có một số ưu điểm:

* Cây quyết định dễ hiểu. Người ta có thể hiểu mô hình cây quyết định sau khi được giải thích ngắn.
* Việc chuẩn bị dữ liệu cho một cây quyết định là cơ bản hoặc không cần thiết. Các kỹ thuật khác thường đòi hỏi [chuẩn hóa dữ liệu](https://vi.wikipedia.org/wiki/Chu%E1%BA%A9n_h%C3%B3a_d%E1%BB%AF_li%E1%BB%87u), cần tạo các biến phụ (dummy variable) và loại bỏ các giá trị rỗng.
* Cây quyết định có thể xử lý cả dữ liệu có giá trị bằng số và dữ liệu có giá trị là tên thể loại. Các kỹ thuật khác thường chuyên để phân tích các bộ dữ liệu chỉ gồm một loại biến. Chẳng hạn, các luật quan hệ chỉ có thể dùng cho các biến tên, trong khi [mạng nơ-ron](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_n%C6%A1-ron) chỉ có thể dùng cho các biến có giá trị bằng số.
* Cây quyết định là một mô hình hộp trắng. Nếu có thể quan sát một tình huống cho trước trong một mô hình, thì có thể dễ dàng giải thích điều kiện đó bằng logic Boolean. Mạng nơ-ron là một ví dụ về mô hình hộp đen, do lời giải thích cho kết quả quá phức tạp để có thể hiểu được.
* Có thể thẩm định một mô hình bằng các kiểm tra thống kê. Điều này làm cho ta có thể tin tưởng vào mô hình.
* Cây quyết định có thể xử lý tốt một lượng dữ liệu lớn trong thời gian ngắn. Có thể dùng [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n) để phân tích các lượng dữ liệu lớn trong một thời gian đủ ngắn để cho phép các nhà chiến lược đưa ra quyết định dựa trên phân tích của cây quyết định.

**2.4.2. Nhược điểm:**

* Nhược điểm của cây quyết định - khó giải quyết được những vấn đề có dữ liệu phụ thuộc thời gian liên tục - dễ xảy ra lỗi khi có quá nhiều lớp chi phí tính toán để xây dựng mô hình cây quyết định cao

**2.5. Kết luận:**

* Hiệu của phân lớp của cây quyết định (Series of Rules) phụ thuộc rất lớn vào training data. Chẳng hạn cây quyết định được tạo ra bởi chỉ giới hạn 10 samples training data trong ví dụ trên thì hiệu quả ứng dụng cây quyết định để dự đoán các trường hợp khác là không cao (thường training data phải đủ lớn và tin cậy)Cây quyết định là một phương pháp phân lớp rất hiệu quả và dễ hiểu. Tuy nhiên có một số chú ý khi sử dụng cây quyết định trong xây dựng các mô hình phân lớp như sau:
* Có rất nhiều thuật toán phân lớp như ID3, J48, C4.5, CART (Classification and Regression Tree),… Việc chọn thuật toán nào để có hiệu quả phân lớp cao tùy thuộc vào rất nhiều yếu tố, trong đó cấu trúc dữ liệu ảnh hưởng rất lớn đến kết quả của các thuật toán. Chẳng hạn như thuật toán ID3 và CART cho hiệu quả phân lớp rất cao đối với các trường dữ liệu số (quantitative value) trong khi đó các thuật toán như J48, C4.5 có hiệu quả hơn đối với các dữ liệu Qualititive value (ordinal, Binary, nominal).

**CHƯƠNG 3: ỨNG DỤNG THUẬT TOÁN**

**3.1.Thuật toán phân lớp cây quyết định dự đoán mưa ở Úc**

**3.1.1. Phát biểu bài toán:**

Bài toán dự đoán thời tiết là đưa ra bộ thông tin thời tiết của 1 ngày sau đó dự đoán ngày mai có mưa không

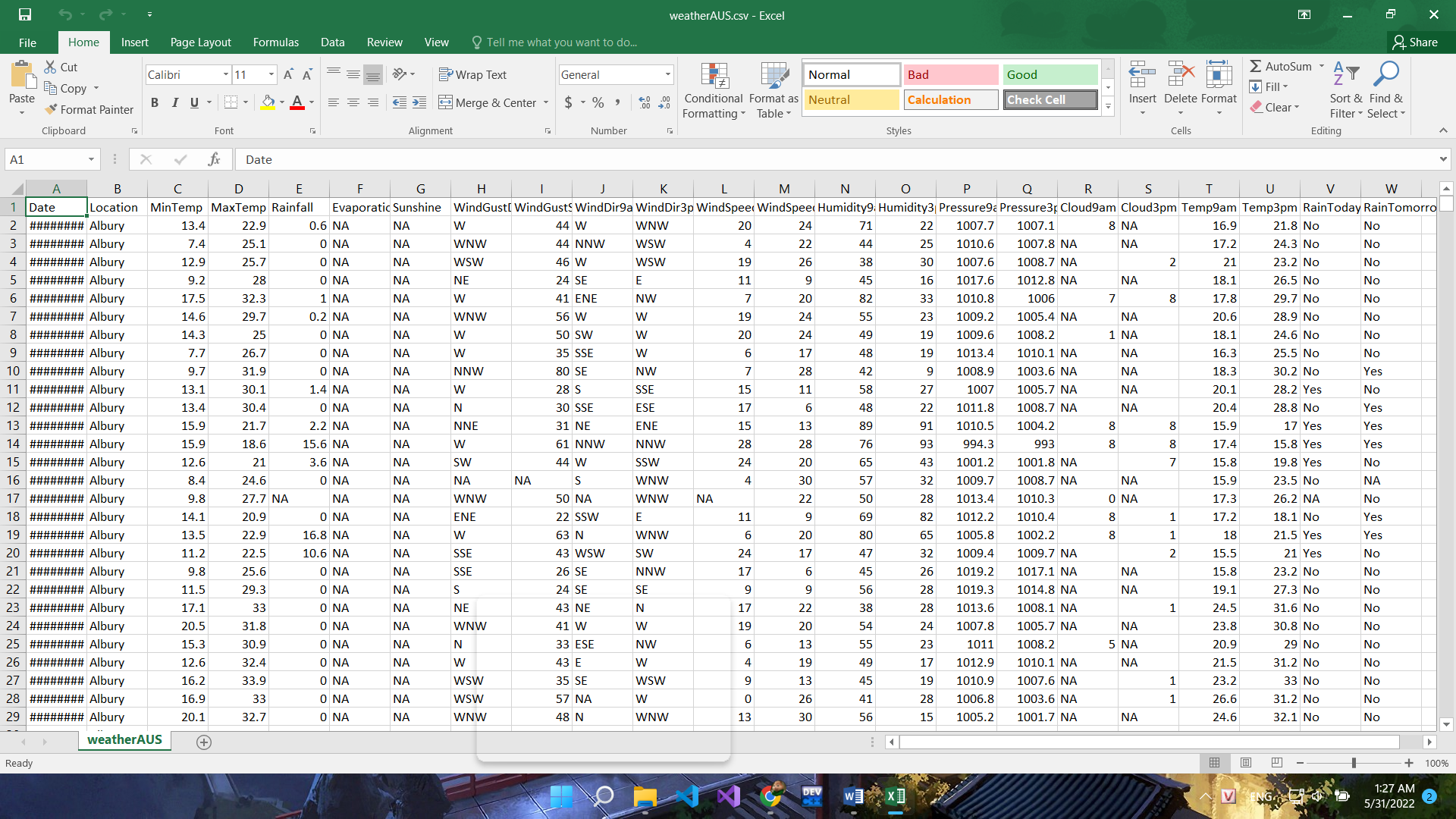
- Input: Thông tin thời tiết của ngày hôm nay

- Output: Dự đoán thời tiết ngày mai

**3.1.2 Yêu cầu bài toán:**

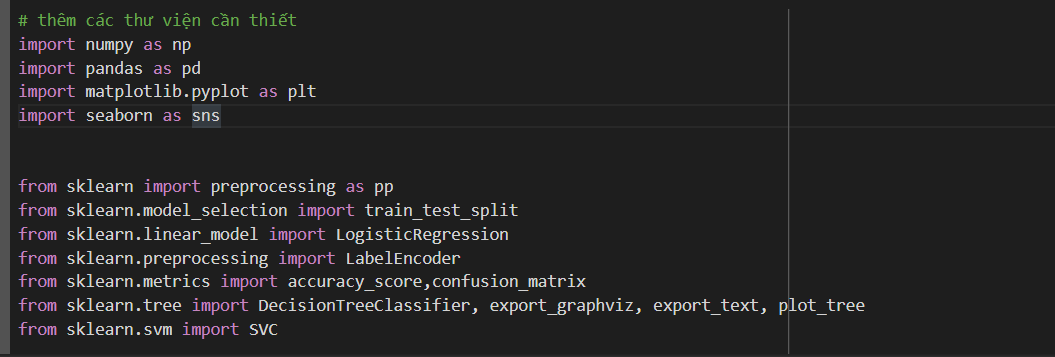
* Lấy dữ liệu từ bộ dữ liệu cung cấp
* Phân tích bộ dữ liệu
* Chuẩn đoán thời tiết

**3.1.3. Xây dựng bộ dữ liệu:**

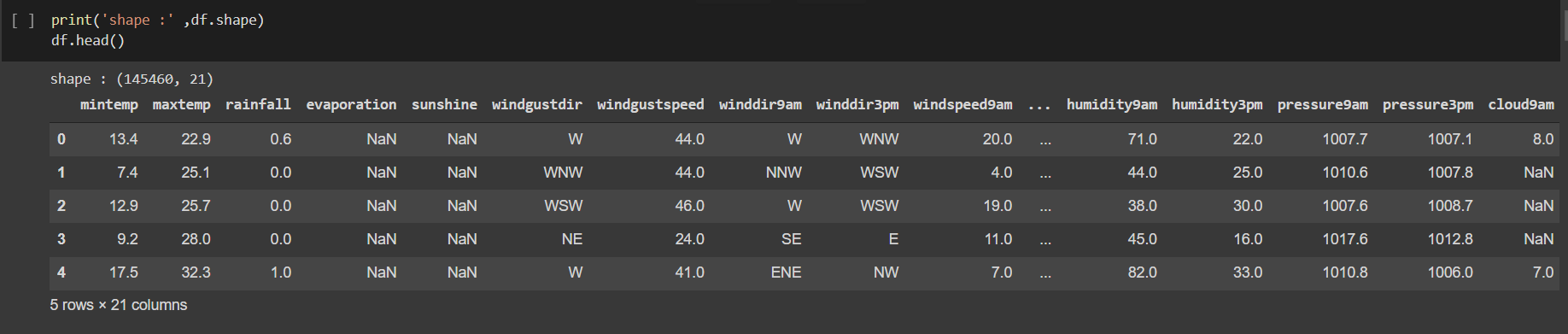


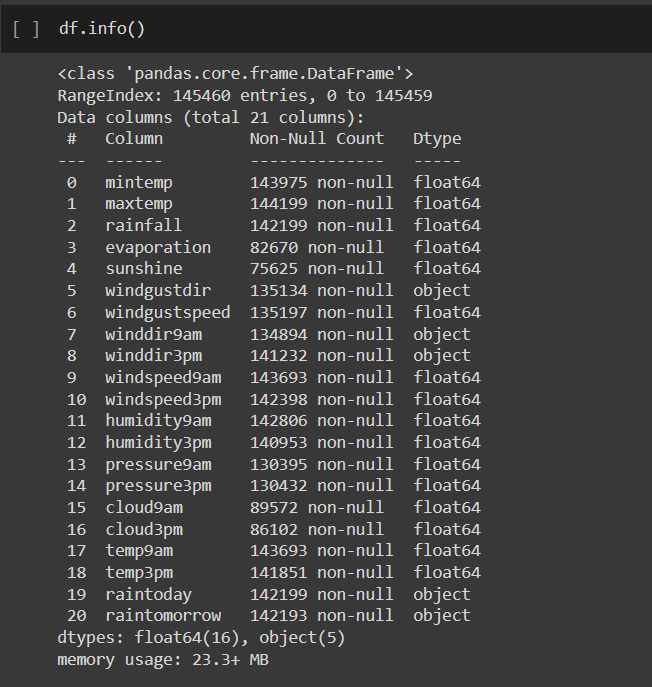
**3.2. Tiến hành xây dựng**

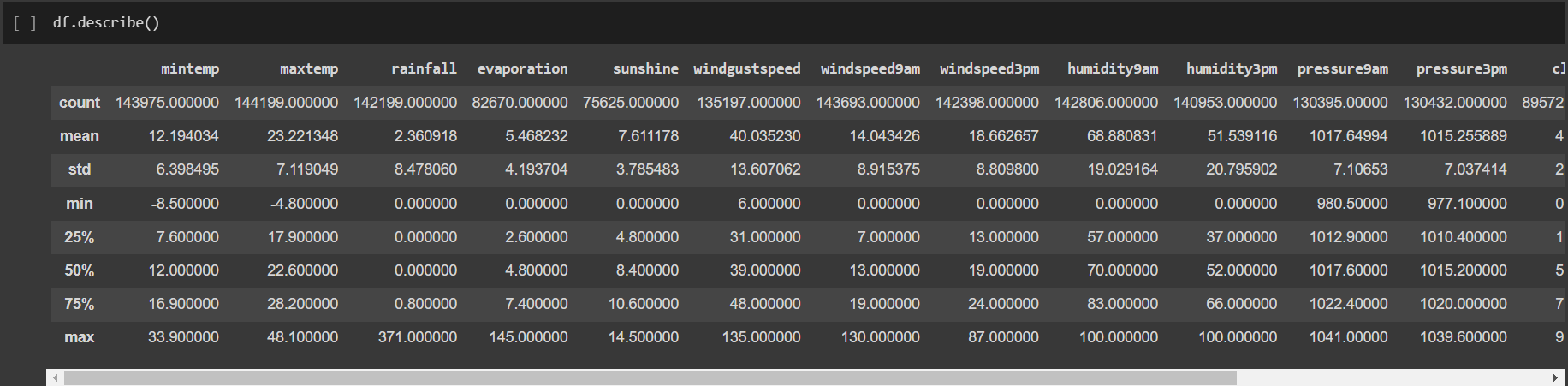
- Khai báo thư viện



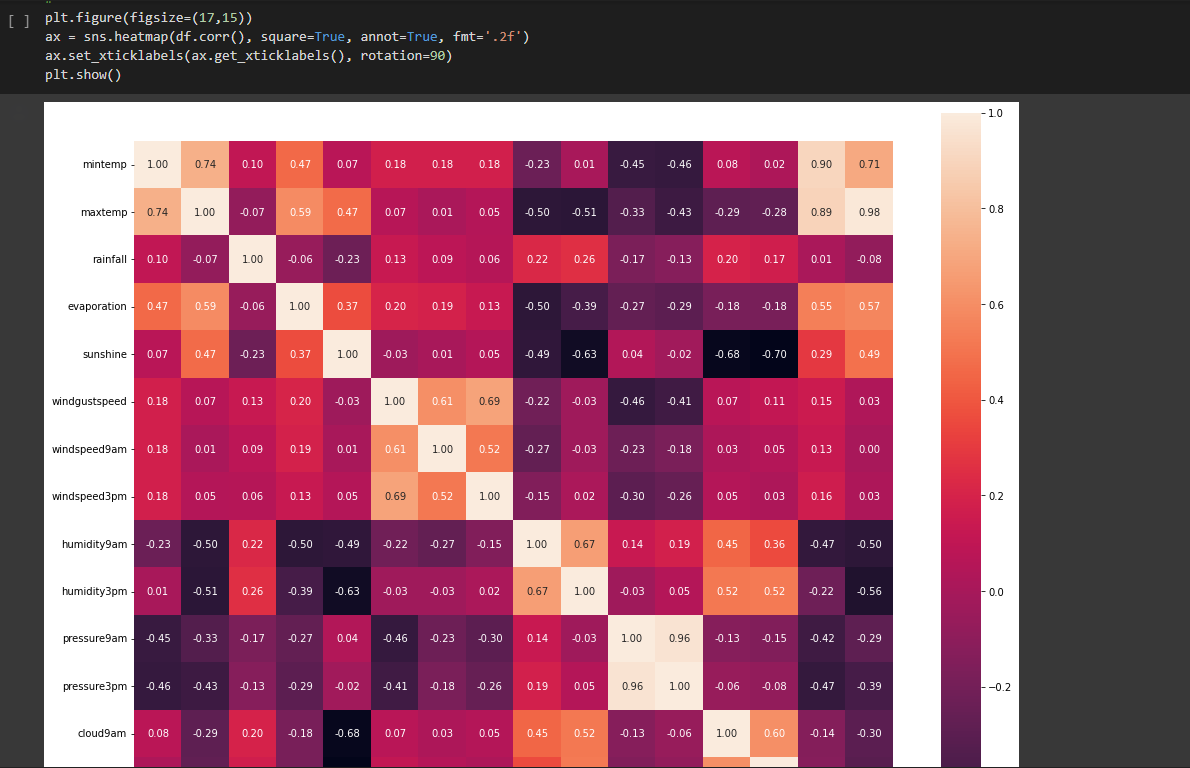
- Thông tin dữ liệu





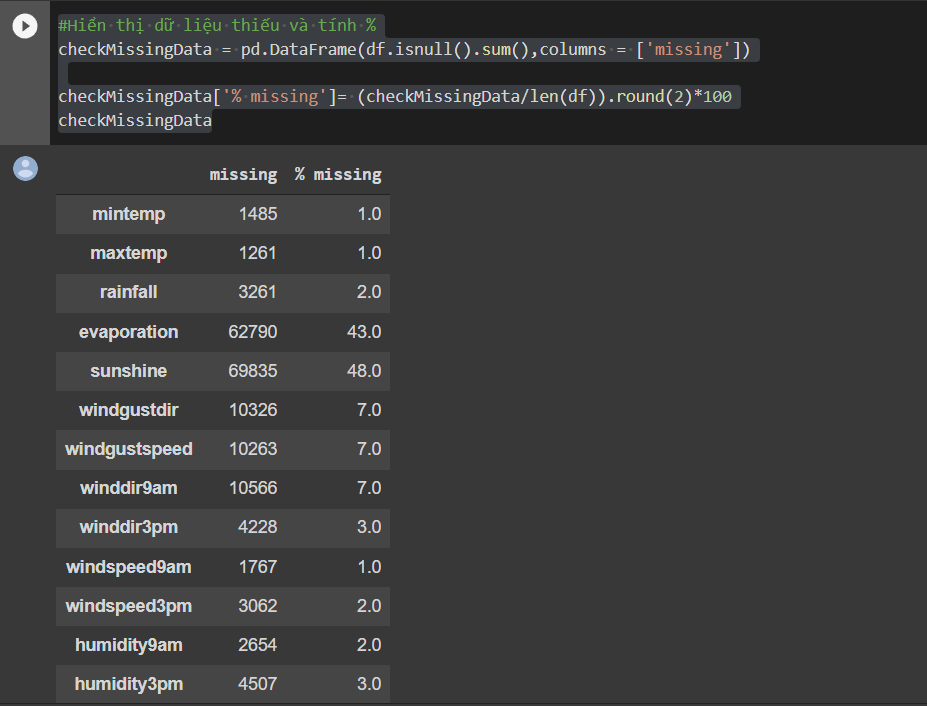


-Biểu đồ tương quan giữa các cột dữ liệu (-1 -> 1)

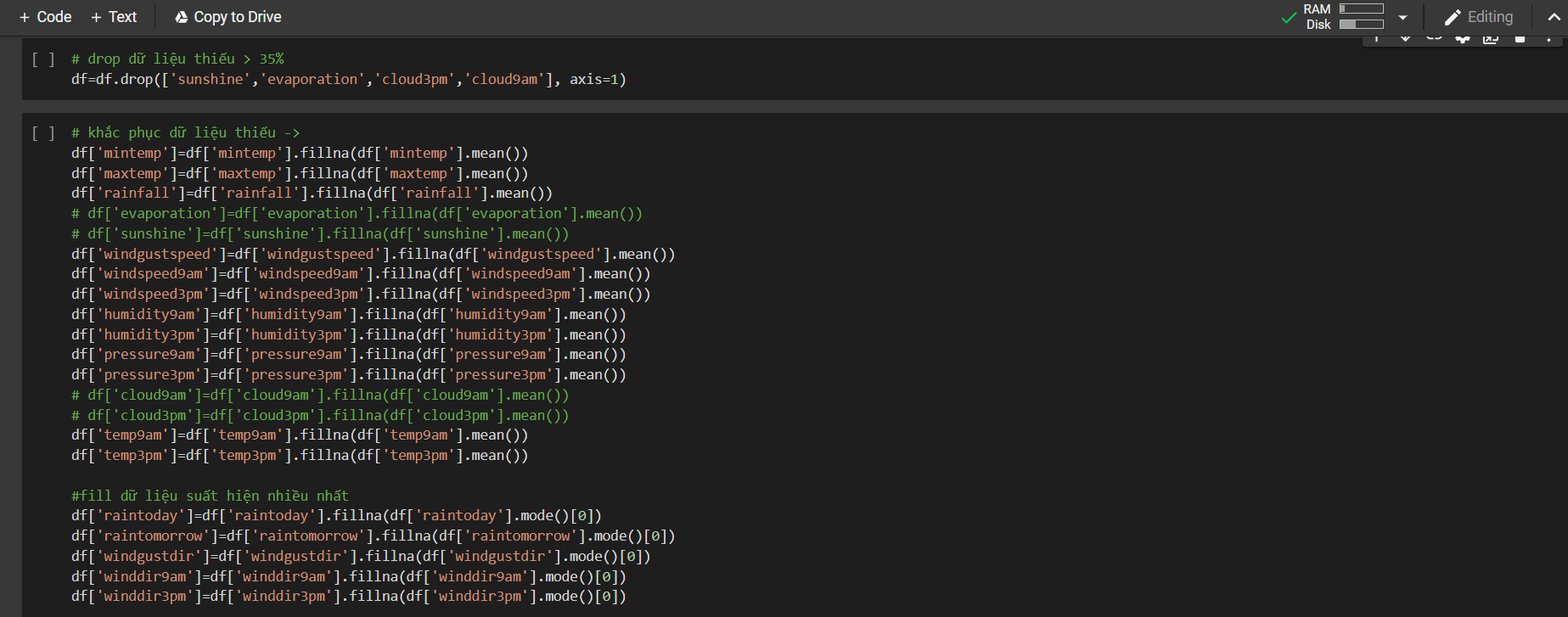


-Xử lý dữ liệu

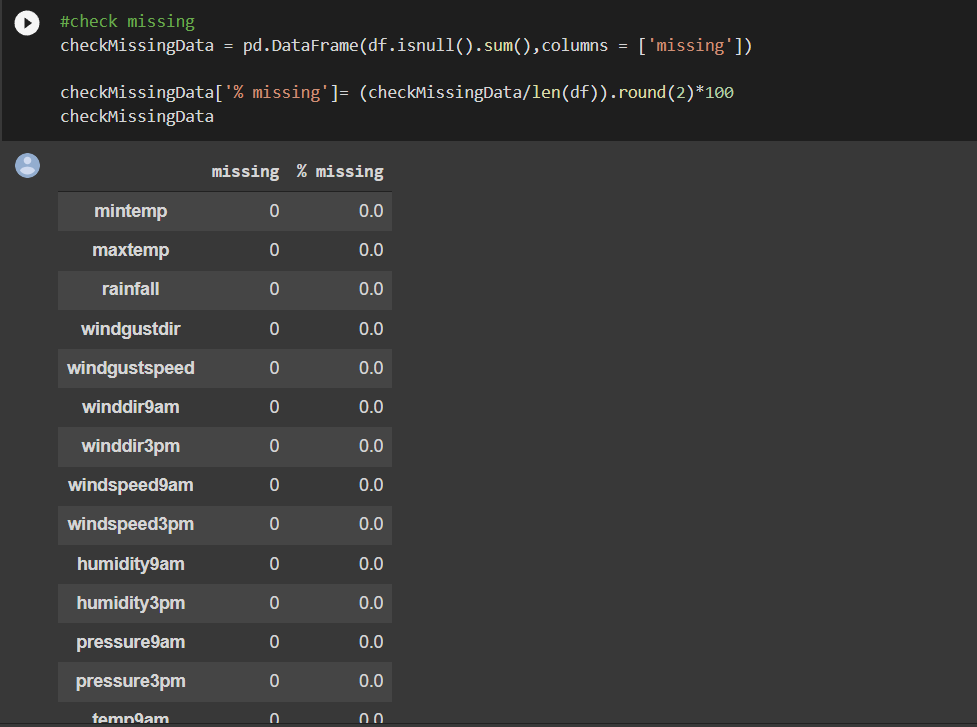
+ Kiểm tra dữ liệu còn thiếu



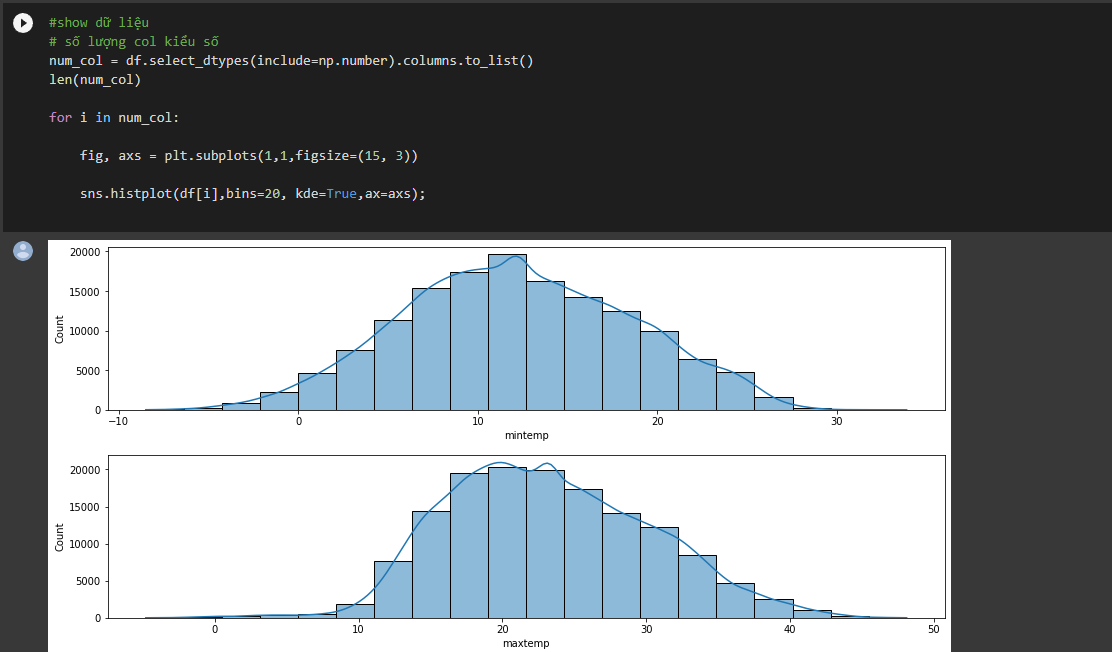
+ Drop những cột có dữ liệu thiếu < 35% và khắc phục dữ liệu bị thiếu



+ Kiểm tra dữ liệu bị thiếu 1 lần nữa

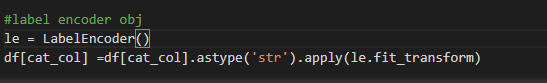


-Biểu đồ các cột dữ liệu

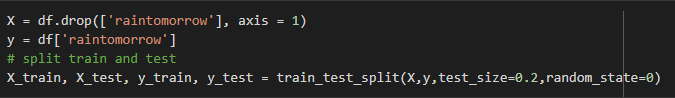




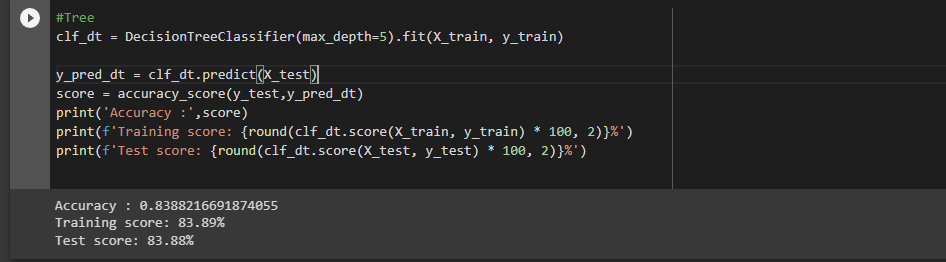
-Chuyển dữ liệu kiểu string sang số



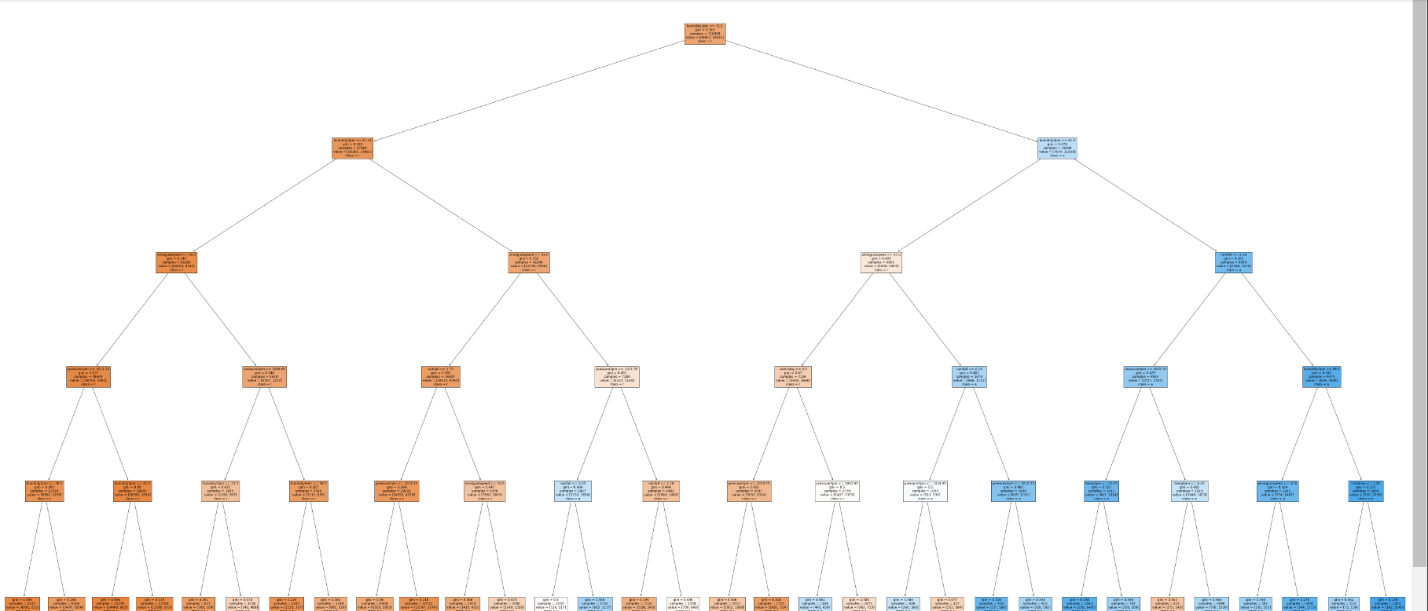
-Gán các cột dữ liệu cho các tham số



-Hiển thị Acuracy Score,Tranning Score,Test Score



-Biểu đồ cây quyết định



**KẾT LUẬN**

Học máy đang được nghiên cứu và ứng dụng trong rất nhiều linh vực, trong đó cây quyết định (Decision tree) cùng với K-means, SVN (Support Vector Machines),... là những giải thuật rất quan trọng của học máy. Không chi với học máy, Cây quyết định còn là một trong những mô hình dự đoán được sử dụng phổ biến trong khai phá dữ liệu (Data mining). Trong nội dung của bài báo này Chúng em đã tiến hành nghiên cứu về giải thuật cây quyết định, đồng thời sử dụng cây quyết định cho một bài toán cụ thể đó là dự đoán thời tiết xây dựng cây quyết định với thuật toán ID3 bằng ngôn ngữ Python cho tập dữ liệu mẫu. Kết quả xây dựng cây quyết định được trình bày một cách trực quan như trong hình 5; Có thể thấy rằng cây quyết định là một phương pháp để phân lớp các đối tượng khá hiệu quả và dễ hiểu. Tuy nhiên để đảm bảo hiệu quả và độ tin cây của các quyết định thì tập dữ liệu mẫu (Training data) phải đủ lớn và đáng tin cây khi đó các tập luật được sinh ra mới là các tập luật tốt. Như tập dữ liệu mẫu ở trên chi với 15 bản ghi thì hiệu quả ứng dụng cây quyết định để dự đoán các trường hợp khác là không cao.