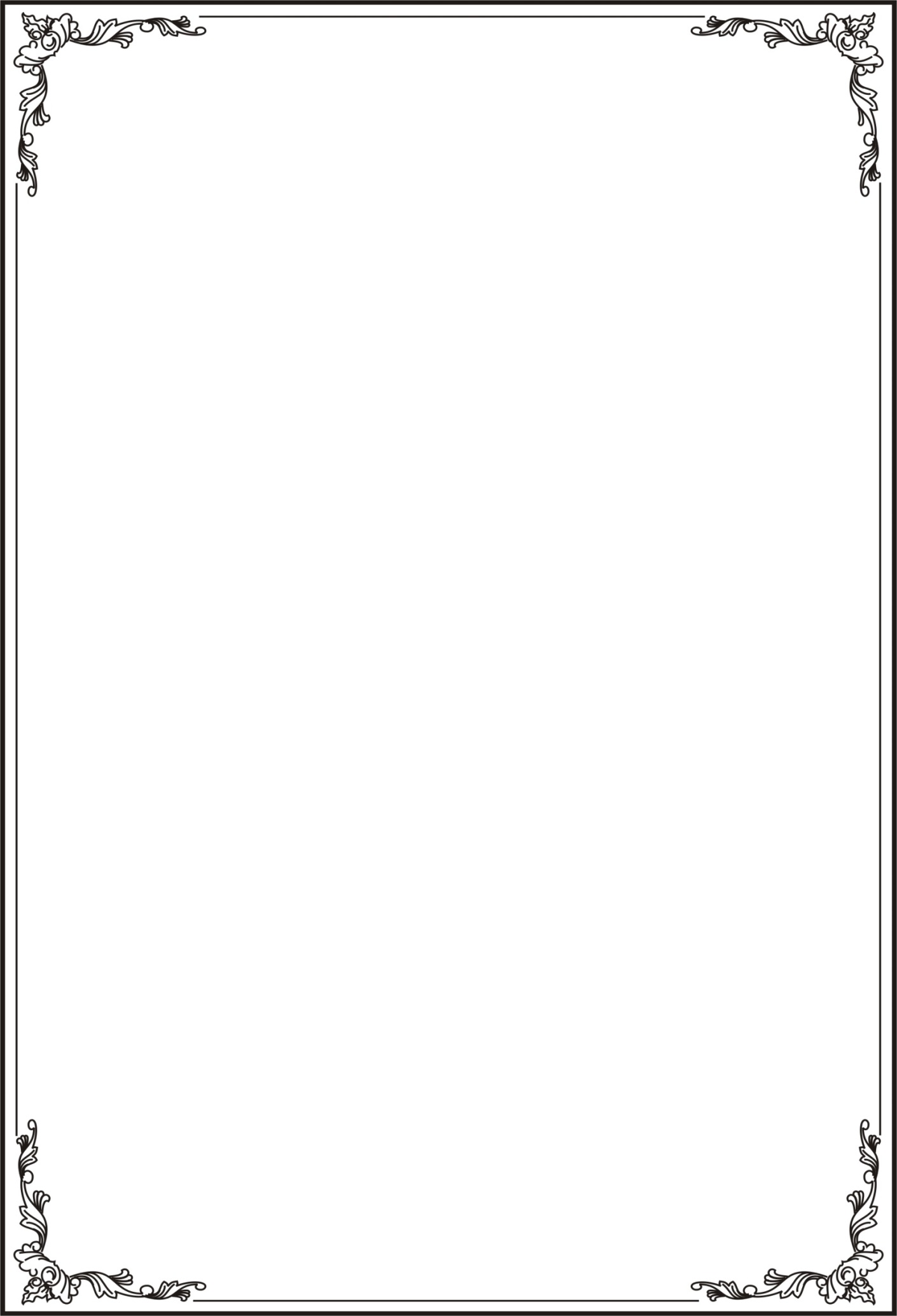
****

**PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**----------------**



**BÁO CÁO MÔN KHAI PHÁ DỮ LIỆU**

**Tên đề tài:**

**Phân loại lưu lượng truy cập Internet trên tường lửa của các trường đại học**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ***Vũ Thị Hạnh*** |
| **Sinh viên thực hiện:** | ***Trần Ngọc Hải - 2051067527*** |
|  | ***Cao Hỷ Khang - 2051067547*** |
|  | ***Lê Trường Duy - 2051067559*** |
| **Lớp:** | ***S23 – 62TH*** |

**TP. Hồ Chí Minh, ngày 22 tháng 10 năm 2023**

Phụ lục

[Lời mở đầu 3](#_Toc146135770)

[Thuật Ngữ 4](#_Toc146135771)

[Bảng phân công công việc 7](#_Toc146135772)

[Chương I: XÁC ĐỊNH VÀ PHÂN TÍCH YÊU CẦU 8](#_Toc146135773)

[1. Giới thiệu chung 8](#_Toc146135774)

[2. Xác định yêu cầu 8](#_Toc146135775)

[3. Dự kiến kết quả đạt được 9](#_Toc146135776)

[4. Công cụ thực hiện đồ án 9](#_Toc146135777)

[Chương II: CƠ SỞ LÝ THUYẾT 9](#_Toc146135778)

[1. Tổng quan môn học 9](#_Toc146135779)

[2. Thuật toán Linear Regression 10](#_Toc146135780)

[3. Thuật toán Cây quyết định 10](#_Toc146135781)

[Chương III: MÔ HÌNH 10](#_Toc146135782)

[1. Mô tả các chức năng của tập dữ liệu 10](#_Toc146135783)

[2. Tiền xử lý dữ liệu (Data Preprocessing) 11](#_Toc146135784)

[3. Thuật toán hồi quy (Linear Regression) 14](#_Toc146135785)

[4. Thuật toán Cây quyết định (Decision Tree) 14](#_Toc146135786)

[Chương IV: Kết quả đạt được 14](#_Toc146135787)

[Chương V: DEMO - Hướng phát triển 14](#_Toc146135788)

# Lời mở đầu

Với sự phát triển mạnh mẽ của Internet và nhu cầu truy cập thông tin ngày càng cao, các trường đại học hiện nay đều cho phép sinh viên truy cập Internet qua mạng LAN của nhà trường. Tuy nhiên, việc sử dụng Internet cũng tiềm ẩn nhiều nguy cơ về lừa đảo, tin tặc và các nội dung đồi trụy, không lành mạnh. Do đó, công tác giám sát và quản lý lưu lượng truy cập Internet qua tường lửa là vô cùng quan trọng.

Trong đề tài này, chúng em sẽ đi sâu vào việc phân loại lưu lượng Internet đi qua tường lửa của các trường Đại học bằng cách sử dụng các kỹ thuật Data Mining. Chúng em sử dụng các thuộc tính như Cổng nguồn, Cổng đích, Cổng nguồn NAT, Cổng đích NAT, Byte, Gói, Thời gian, pkts\_sent, và pkts\_received,… để khám phá thông tin ẩn bên trong dữ liệu. Qua đó, phân tích thống kê các loại lưu lượng, từ đó đề xuất các giải pháp tăng cường bảo mật và quản lý hiệu quả hơn việc sử dụng Internet lành mạnh trong trường học. Chúng em hy vọng rằng báo cáo này sẽ đánh dấu một bước đi quan trọng trong việc nghiên cứu và quản lý lưu lượng truy cập Internet tại trường đại học của chúng ta.

Đặc biệt, nhóm em gửi lời cảm ơn sâu sắc và chân thành nhất đến Giảng viên Vũ Thị Hạnh. Cô đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp ý kiến quý báo giúp nhóm em hoàn thành tốt báo cáo đồ án của mình. Trong thời gian thực hiện đề tài, nhóm em đã vận dụng được những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới.

Xin chân thành cảm ơn các quý Thầy Cô !

# DANH MỤC CÁC THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| **Thuật ngữ viết tắt** | **Thuật ngữ đầy đủ** | **Ý nghĩa** |
| Firewall | Thiết bị an ninh mạng hạn chế truy cập giữa các mạng. | Tường lửa |
| Source Port | Cổng trên máy nguồn gửi dữ liệu | Cổng nguồn |
| Destination Port | Cổng trên máy đích nhận dữ liệu. | Cổng đích |
| NAT Source Port | Thay đổi cổng nguồn khi dữ liệu đi qua NAT. | Cổng nguồn sau khi chuyển đổi NAT |
| NAT Destination Port | Thay đổi cổng đích khi dữ liệu đi qua NAT. | Cổng đích sau khi chuyển đổi NAT |
| Action | Hành động mà tường lửa sau khi kiểm tra sẽ thực hiện | Trạng thái Allow, Deny, Drop, Reset-Both của đường truyền |
| Bytes | Kích thước dữ liệu | Số byte |
| Bytes Sent | Kích thước dữ liệu gửi đi | Số byte gửi đi |
| Bytes Received | Kích thước dữ liệu gửi đi | Số byte nhận được |
| Packets | Đơn vị dữ liệu nhỏ nhất trong mạng | Số gói tin |
| Elapsed Time (sec) | Thời gian trôi qua. | Thời gian xử lý gói tin |
| pkts\_sent | Gói tin gửi | Số gói tin đã gửi |
| pkts\_received | Gói tin nhận | Số gói tin đã nhận |
| CNPM | Cổng, địa chỉ mạng, giao thức | Công nghệ phần mềm |
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| NAT | Network Address Translation | Chuyển đổi Địa chỉ Mạng |

# Chương I: XÁC ĐỊNH VÀ PHÂN TÍCH YÊU CẦU

## Giới thiệu chung

Những năm gần đây, trên Internet thường xuyên xảy ra tình trạng có rất nhiều kẻ với mưu đồ xấu, lợi dụng thời cơ để xâm nhập vào những máy tính có kết nối với mạng Internet. Sau hàng loạt vụ tấn công mạng diễn ra liên tiếp với quy mô lớn, vấn đề bảo mật máy tính chính là một trong những chủ đề đang được quan tâm nhiều hơn bao giờ hết. Để đảm bảo an toàn trên internet, bạn cần đặc biệt chú ý tường lửa (Firewall). Nhờ có chiếc rào chắn Firewall này sẽ giúp bạn tạo ra một bức tường thành kiên cố, nâng cao tính bảo mật cho toàn bộ hệ thống máy tính của bạn. Vậy tường lửa là gì?

Tường lửa (Firewall) là một hệ thống an ninh mạng, có thể dựa trên phần cứng hoặc phần mềm, sử dụng các quy tắc để kiểm soát traffic vào, ra khỏi hệ thống. Tường lửa hoạt động như một rào chắn giữa mạng an toàn và mạng không an toàn. Nó kiểm soát các truy cập đến nguồn lực của mạng thông qua một mô hình kiểm soát chủ động. Nghĩa là, chỉ những traffic phù hợp với chính sách được định nghĩa trong tường lửa mới được truy cập vào mạng, mọi traffic khác đều bị từ chối.

Bất kì máy tính nào kết nối tới Internet cũng cần có firewall, giúp quản lý những gì được phép vào mạng và những gì được phép ra khỏi mạng. Việc có một “người gác cổng” như vậy để giám sát mọi việc xảy ra rất quan trọng bởi 2 lý do:

Thứ nhất, bất kì máy tính kết nối mạng nào thường kết nối vĩnh viễn với Internet. Thứ 2, mỗi máy tính trực tuyến lại có một chữ ký điện tử riêng, được gọi là Internet Protocol address (hay còn gọi là địa chỉ IP): Nếu không có firewall hỗ trợ, nó chẳng khác gì chuyện bạn bật tất cả đèn lên và mở rộng cửa để đón trộm vào.

Một firewall được cấu hình chính xác sẽ ngăn chặn điều này xảy ra và giúp máy tính “ẩn” một cách hiệu quả, cho phép người dùng thoải mái thưởng thức những gì thế giới trực tuyến mang lại. Firewall không giống chương trình diệt virus. Thay vào đó, nó làm việc cùng với những công cụ này nhằm đảm bảo rằng máy tính được bảo vệ từ hầu hết các mối tấn công nguy hại phổ biến.

Đề tài phân loại và phân tích lưu lượng truy cập Internet trên tường lửa của trường đại học nhằm mục đích nghiên cứu và phân tích các thông tin lưu lượng đi qua tường lửa của mạng máy tính trong trường đại học.

Với số lượng lớn sinh viên và nhu cầu truy cập Internet ngày càng tăng, việc giám sát và phân tích lưu lượng mạng là hết sức cần thiết để đảm bảo an toàn thông tin cho nhà trường. Do đó, đề tài đề xuất sử dụng các kỹ thuật học máy và phân loại dữ liệu để tự động nhận diện các luồng lưu lượng, phân loại thành lưu lượng bình thường, đáng ngờ hoặc nguy hiểm.

Kết quả của đề tài sẽ giúp nhà trường có cái nhìn sâu sắc về thống kê và xu hướng lưu lượng Internet, từ đó có chiến lược quản lý và đảm bảo an ninh mạng hợp lý, phù hợp.

## Xác định yêu cầu

Tiền xử lý dữ liệu: loại bỏ dữ liệu không cần thiết, xử lý dữ liệu bị thiếu, nhiễu, chuẩn hóa dữ liệu về cùng một domain.

Xây dựng các mô hình phân loại dữ liệu sử dụng các thuật toán để phân loại lưu lượng thành các nhãn: Allow (cho phép), Deny (từ chối), Drop (loại bỏ), Reset-Both (đặt lại kết nối).

Đánh giá các mô hình phân loại và chọn ra mô hình phù hợp nhất dựa trên các tiêu chí: độ chính xác, độ nhạy, độ đặc hiệu, v.v.

Phân tích các kết quả dự đoán của mô hình, thống kê tỷ lệ các loại lưu lượng.

Đề xuất các giải pháp nhằm tối ưu hóa việc quản lý và lọc lưu lượng Internet trong trường đại học dựa trên kết quả.

## Dự kiến kết quả đạt được

* Thu thập được dữ liệu.
* Hoàn thành công đoạn tiền xử lý dữ liệu.
* Xây dựng được các mô hình.
* Mô hình đạt khoảng chính xác cao.
* Phân tích và thông kê được tỷ lệ truy cập.
* Đề xuất.

## Công cụ thực hiện đồ án

* Nhóm tụi em dùng **Colaboratory** hay còn gọi là **Google Colab** để thực hiện đồ án.
* Google Colab là một sản phẩm từ Google Research, nó cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, machine learning và giáo dục.

# Chương II: CƠ SỞ LÝ THUYẾT

## Tổng quan môn học

Khoa học dữ liệu (Data Science) là một lĩnh vực nghiên cứu và ứng dụng dữ liệu để hiểu, phân tích, dự đoán và tạo ra giá trị từ thông tin chứa trong dữ liệu. Khoa học dữ liệu sử dụng kết hợp các phương pháp từ nhiều lĩnh vực như thống kê, toán học, khoa học máy tính, và nguồn dữ liệu thực tế để trích xuất thông tin hữu ích và kiến thức từ dữ liệu.

Khai phá dữ liệu (Data Mining) là quá trình tự động hoặc bán tự động tìm kiếm thông tin có giá trị từ tập dữ liệu lớn. Mục tiêu chính của khai phá dữ liệu là khám phá các mẫu, quy luật, sự kết nối, và thông tin ẩn trong dữ liệu mà có thể không dễ thấy bằng cách thường xuyên.

Tóm lại, khoa học dữ liệu là một lĩnh vực tổng hợp hơn, bao gồm cả việc khai phá dữ liệu, trong khi khai phá dữ liệu tập trung chủ yếu vào việc khám phá dữ liệu một cách tự động để tìm ra thông tin thú vị và quy luật trong dữ liệu.

## Thuật toán Ramdom Forest

Random Forest là thuật toán xây dựng nhiều cây quyết định phân loại/hồi quy độc lập. Các cây được xây dựng từ tập huấn luyện ngẫu nhiên kéo tầng và các biến đầu vào quan trọng nhất được chọn ngẫu nhiên.

Khi dự đoán, mỗi cây sẽ đưa ra kết quả riêng và kết quả chung sẽ là kết quả phổ biến nhất trong tất cả các cây. Random Forest có khả năng phân loại tốt trên cả dữ liệu cân bằng và dữ liệu không cân bằng. Thuật toán dễ triển khai và có các tham số quan trọng cần điều chỉnh là số cây và số biến được chọn tại mỗi nút cây. Điều này giúp giảm bớt sự phụ thuộc vào các biến đầu vào.

Random Forest có thể xác định được các biến quan trọng nhất thông qua việc đánh giá độ phụ thuộc của các cây vào mỗi biến.

## Thuật toán Cây quyết định

Cây quyết định (Decision Tree) là một mô hình học máy phân loại và dự đoán được sử dụng trong nhiều ứng dụng khác nhau. Nó là một biểu đồ cây có cấu trúc, mô phỏng quá trình ra quyết định hoặc dự đoán bằng cách sử dụng luật điều kiện và giá trị của các thuộc tính.

Một cây quyết định được xây dựng từ dữ liệu huấn luyện bằng cách phân chia tập dữ liệu thành các phần con (subsets) dựa trên các thuộc tính. Quá trình này tiếp tục cho đến khi mỗi phần con chỉ chứa các điểm dữ liệu thuộc cùng một lớp hoặc có giá trị dự đoán gần nhau.

1. **Thuật toán Naive Bayes**

Là một thuật toán máy học phân loại dựa trên nguyên tắc xác suất và lý thuyết xác suất Bayes. Ý tưởng cơ bản của thuật toán Naive Bayes là sử dụng các thông tin về xác suất để dự đoán xem một mẫu dữ liệu thuộc vào một lớp (hoặc nhãn) cụ thể nào. Đây là một phần của lý thuyết xác suất Bayes, mà công thức cơ bản được biểu thị như sau:

P(y | X) = P(X | y) \* P(y) / P(X)

• P(y | X) là xác suất điều kiện của lớp y khi đã biết mẫu dữ liệu X.

• P(X | y) là xác suất của mẫu dữ liệu X khi đã biết lớp y.

• P(y) là xác suất tiên nghiệm của lớp y.

• P(X) là xác suất của mẫu dữ liệu X.

Thuật toán Naive Bayes có một số ưu điểm và nhược điểm đáng xem xét. Ưu điểm của nó bao gồm tính đơn giản và tốc độ huấn luyện nhanh, đặc biệt phù hợp cho dữ liệu nhỏ. Nó cũng hiệu quả trong việc phân loại. Tuy nhiên, Naive Bayes có nhược điểm khi không thể xử lý tốt dữ liệu có đội đối, và giả định về tính độc lập của các đặc trưng có thể không phù hợp với thực tế. Ngoài ra, thuật toán này đòi hỏi ước tính các xác suất từ dữ liệu, và kết quả có thể bị ảnh hưởng khi dữ liệu đào tạo không đủ hoặc có sự thiên vị.

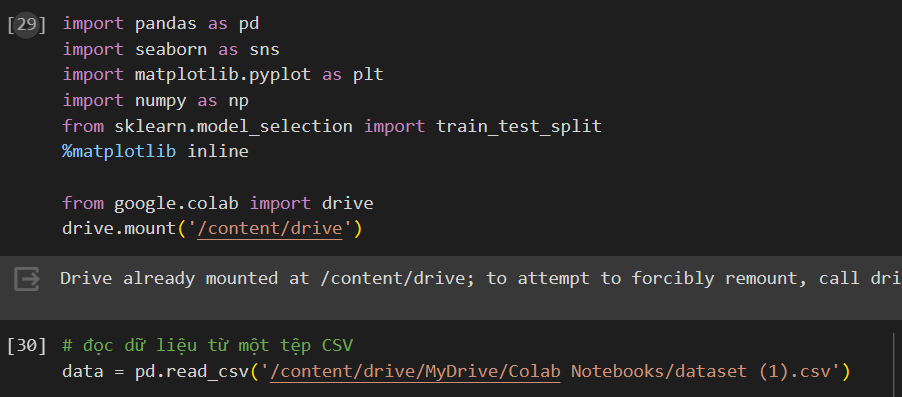
# Chương III: MÔ HÌNH

## Mô tả các chức năng của tập dữ liệu

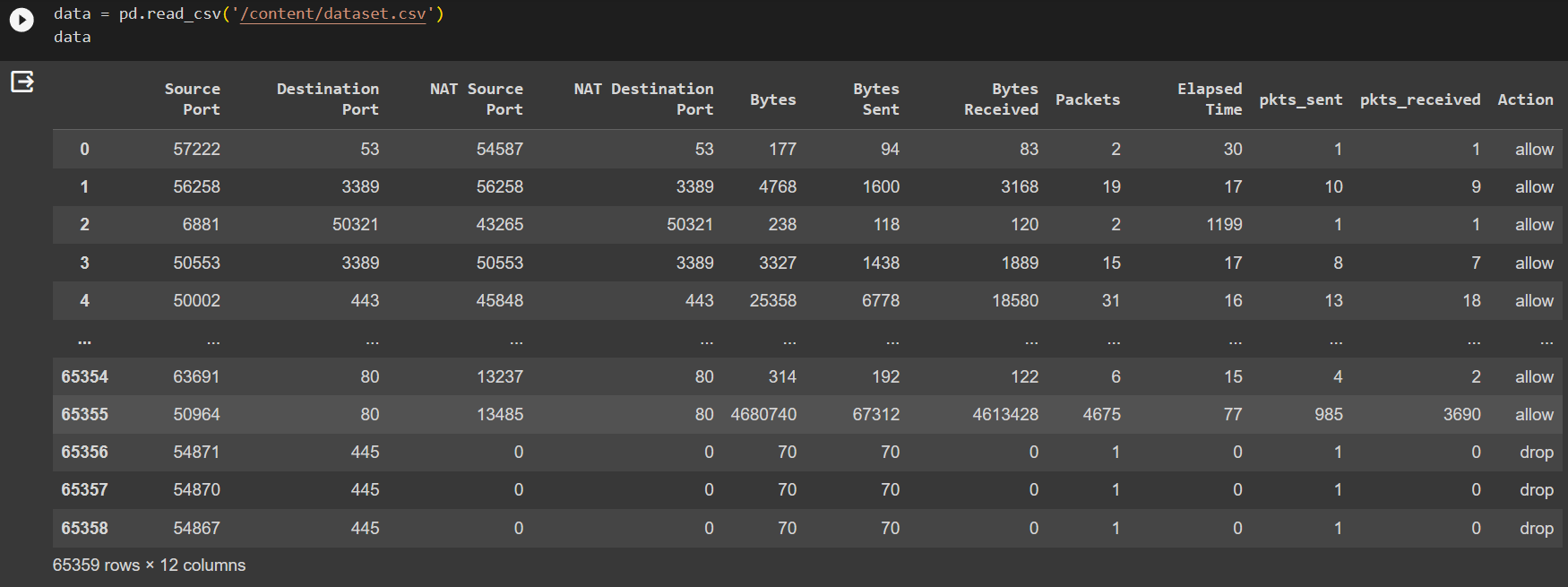
* **Source Port** và **Destination Port**: Các cổng nguồn và cổng đích thường được sử dụng trong giao thức truyền thông mạng để xác định nguồn và đích của gói tin mạng. Tương quan giữa chúng có thể giúp xác định loại kết nối mạng (ví dụ: dịch vụ, ứng dụng) đang được sử dụng.
* **NAT Source Port** và **NAT Destination Port**: Các trường này có thể liên quan đến quá trình Network Address Translation (NAT), trong đó địa chỉ IP hoặc cổng của gói tin được thay đổi để đi qua một bộ định tuyến hoặc tường lửa. Tương quan giữa chúng có thể giúp theo dõi các kết nối NAT.
* **Action**: Trường Action thường chỉ hành động được thực hiện trên gói tin mạng, ví dụ: chấp nhận (accept), từ chối (deny), hoặc thả (drop). Tương quan với các trường khác có thể giúp bạn hiểu cách gói tin được xử lý.
* **Bytes**, **Bytes Sent**, và **Bytes Received**: Các trường này liên quan đến lưu lượng dữ liệu truyền qua kết nối mạng. Bytes Sent thường chỉ lượng dữ liệu gửi đi, trong khi Bytes Received chỉ lượng dữ liệu nhận được. Tương quan giữa chúng có thể giúp bạn đánh giá tải trọng của kết nối.
* **Packets**, **pkts\_sent**, và **pkts\_received**: Các trường này liên quan đến số lượng gói tin truyền qua kết nối mạng. Packets thường chỉ tổng số gói tin, pkts\_sent chỉ số lượng gói tin gửi đi, và pkts\_received chỉ số lượng gói tin nhận được.
* **Elapsed Time (sec)**: Trường này thường liên quan đến thời gian mà kết nối mạng hoặc hoạt động đang diễn ra.

## Tiền xử lý dữ liệu (Data Preprocessing)

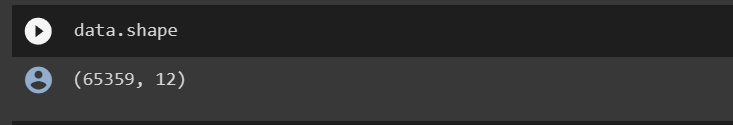
* QUY TRÌNH KHAI THÁC DỮ LIỆU
* Khai báo thư viện và lấy dữ liệu từ google drive



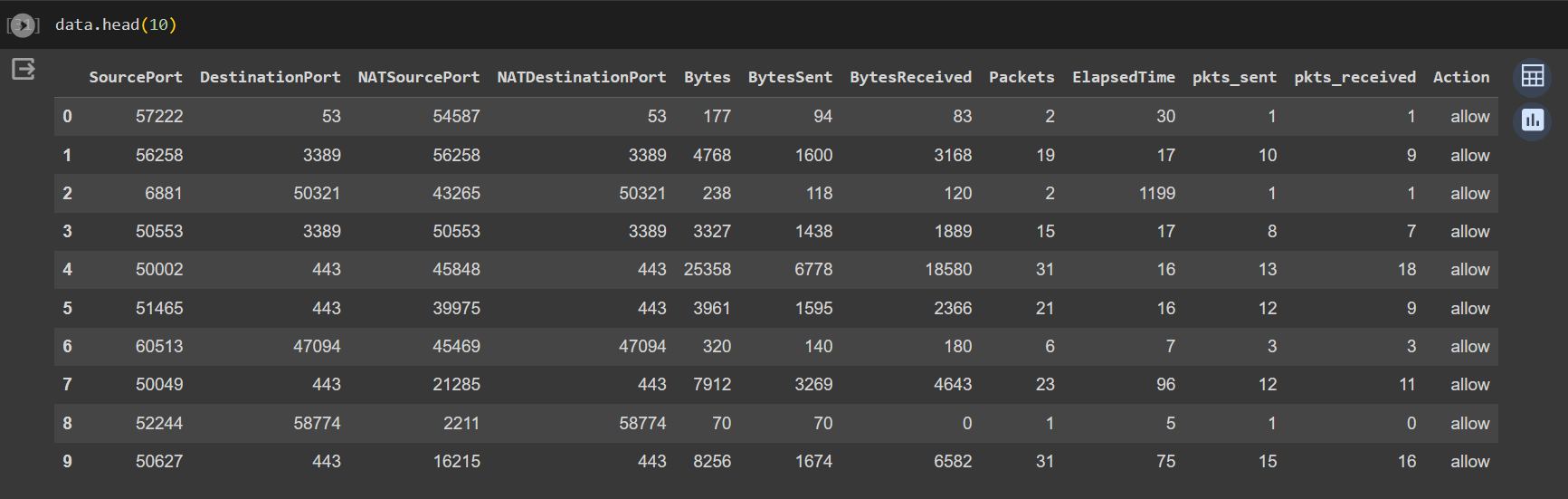
* Bộ dữ liệu thô vừa lấy chưa qua xử lý



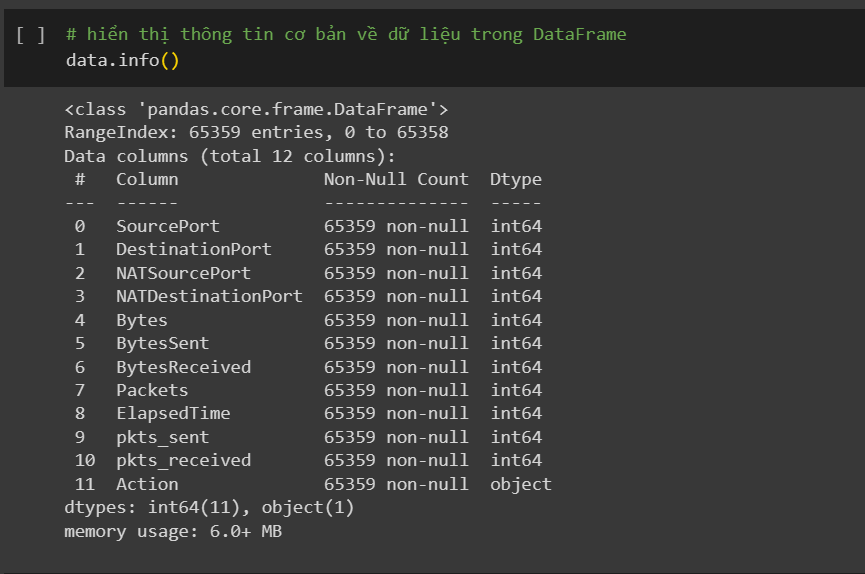
* Tổng số dòng, cột dữ liệu thô



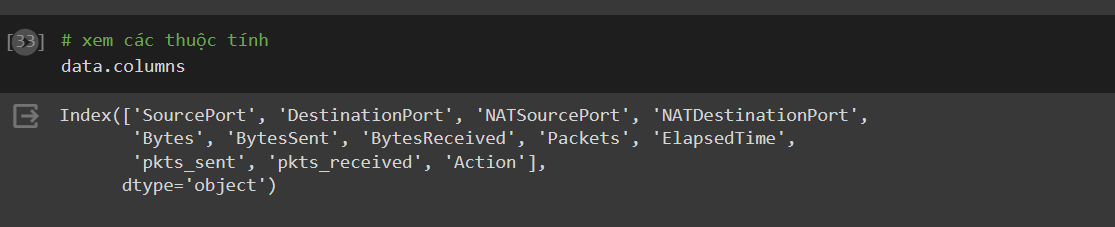
* Lấy 10 dòng dữ liệu đầu tiên



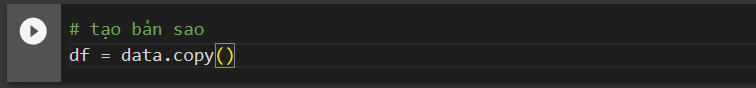
* Tổng quan về dữ liệu thô



* Qua dữ liệu trên ta thấy được kiểu dữ liệu của các thuộc tính đều là số, trừ thuộc tính “Action” là kiểu ký tự.
* Xem tên các thuộc tính xuất hiện trong dữ liệu

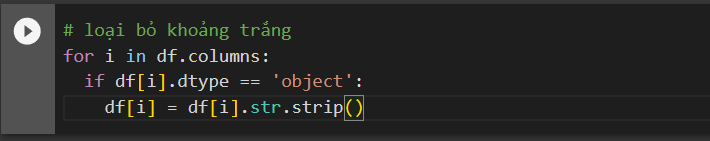


* Tiền Xữ Lý Dữ Liệu
* Tiến hành tạo bản sao dữ liệu

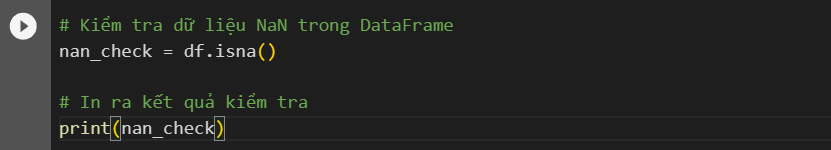


Sau đó:

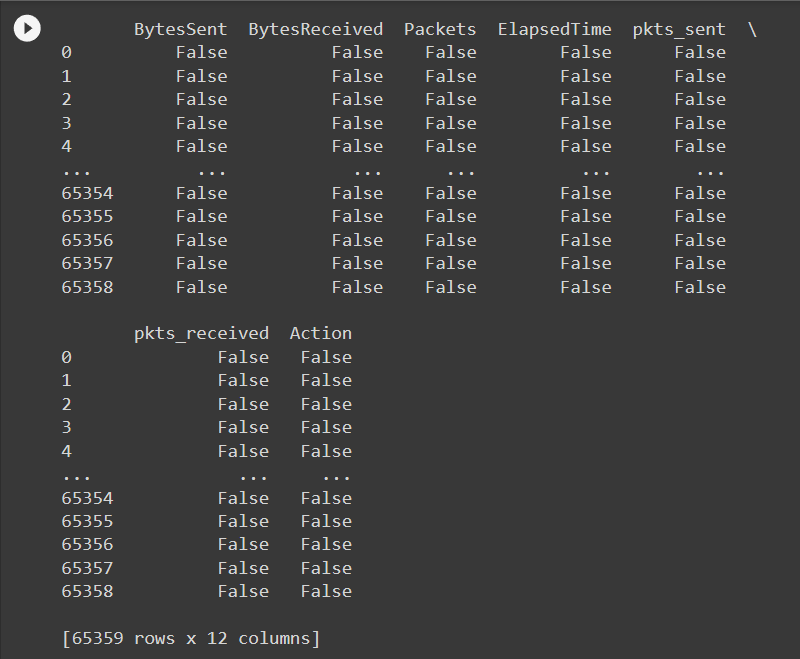
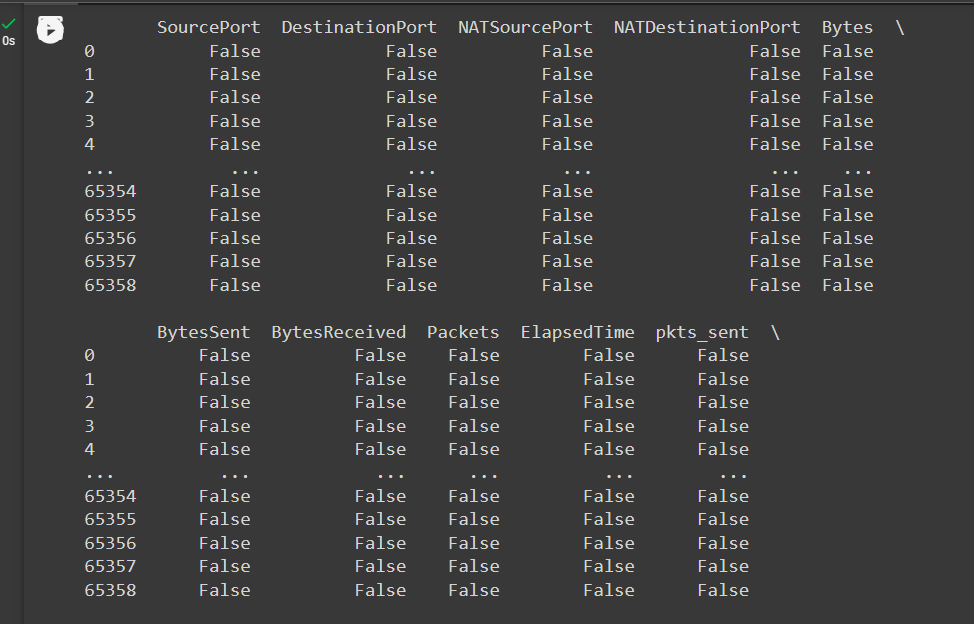
* Loại bỏ khoảng trắng trong dữ liệu



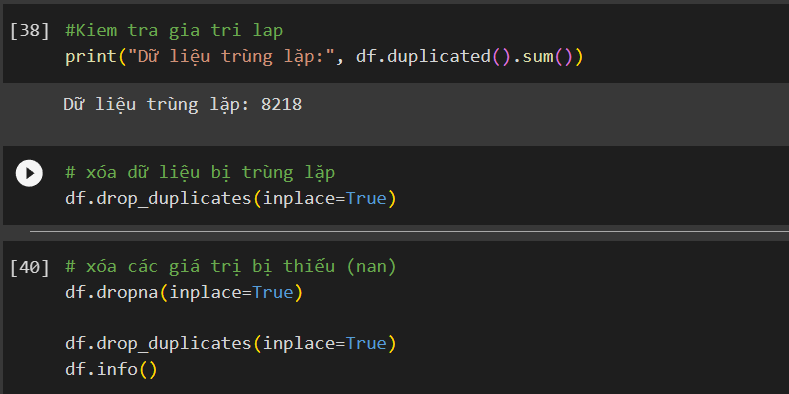
* Kiểm tra các dữ liệu mang giá trị NaN



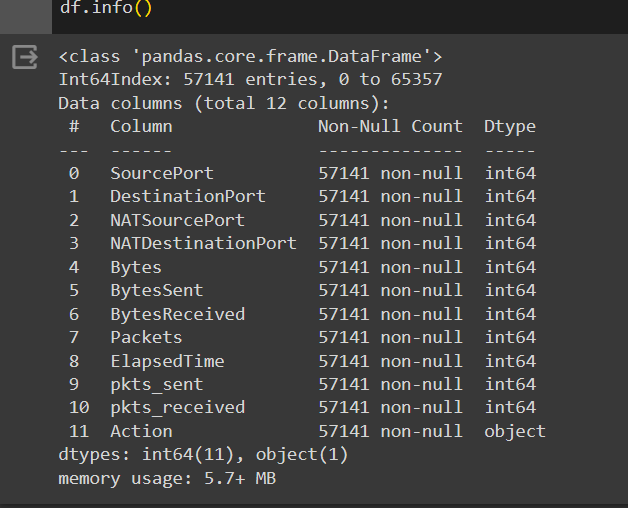
Trong kết quả sau, những dữ liệu False ( không NaN) sẽ được giữ lại, còn dữ liệu True (Nan) ta sẽ tiếng hành xóa :



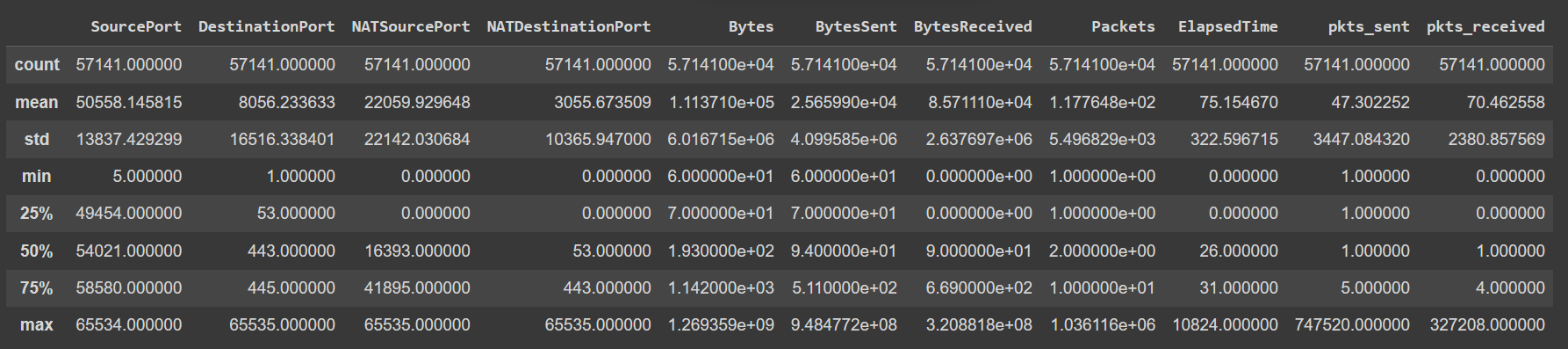
* Kiểm tra và xóa giá trị trùng lặp và giá trị NaN



Ta thấy có tới 8218 dữ liệu bị trùng lặp. Sau khi thực hiện xóa bộ dữ liệu của ta từ 65359 dòng còn 57141 dòng dữ liệu, các thuộc tính trùng lặp và Nan đã được xóa:



* Thống kê dữ liệu



* Nhìn kết quả, ta thấy:

- Số lượng (count): Có 57141 giá trị trong mỗi cột, cho thấy không có giá trị thiếu (null) trong dữ liệu.

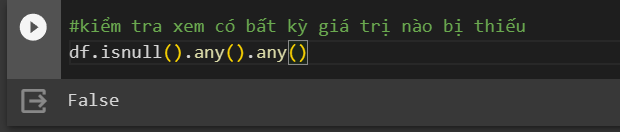
- Trung bình (mean): Giá trị trung bình của các cột có sự biến động khá lớn.

- Độ lệch chuẩn (std): cột Bytes có độ lệch chuẩn là khoảng 6016714, cho thấy sự biến đổi lớn trong dữ liệu.

- Giá trị tối thiểu (min) và giá trị tối đa (max): Cho biết giá trị nhỏ nhất và lớn nhất trong mỗi cột. Ví dụ, giá trị tối thiểu của cột SourcePort là 5 và giá trị tối đa là 65534.

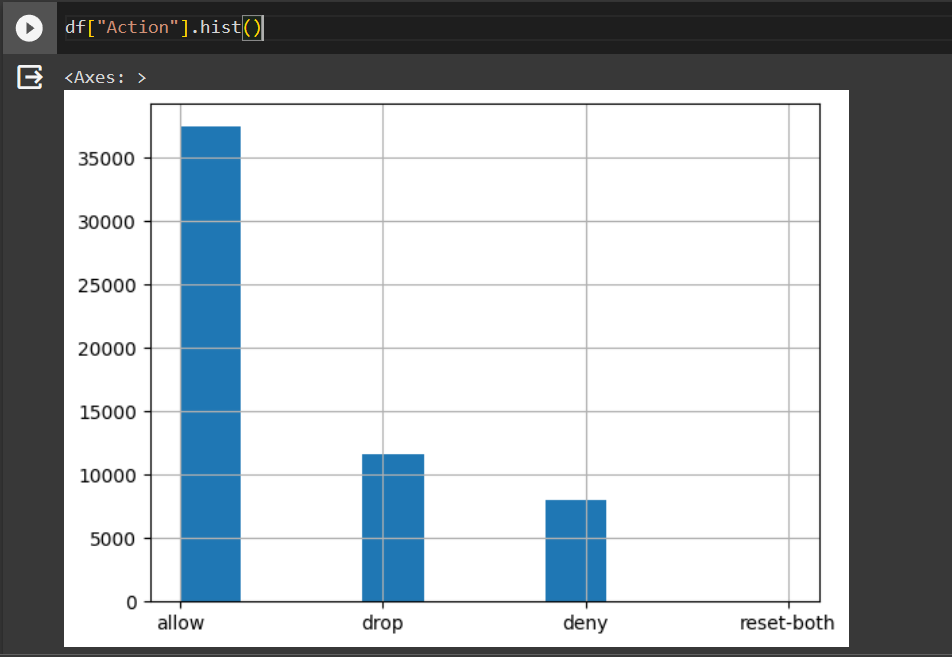
- Percentiles (25%, 50%, 75%): Đây là các phân vị (percentiles) của dữ liệu. Các phân vị này có thể giúp nhận ra đặc điểm phân bố của dữ liệu.

* Kiểm tra giá trị thiếu trong dữ liệu

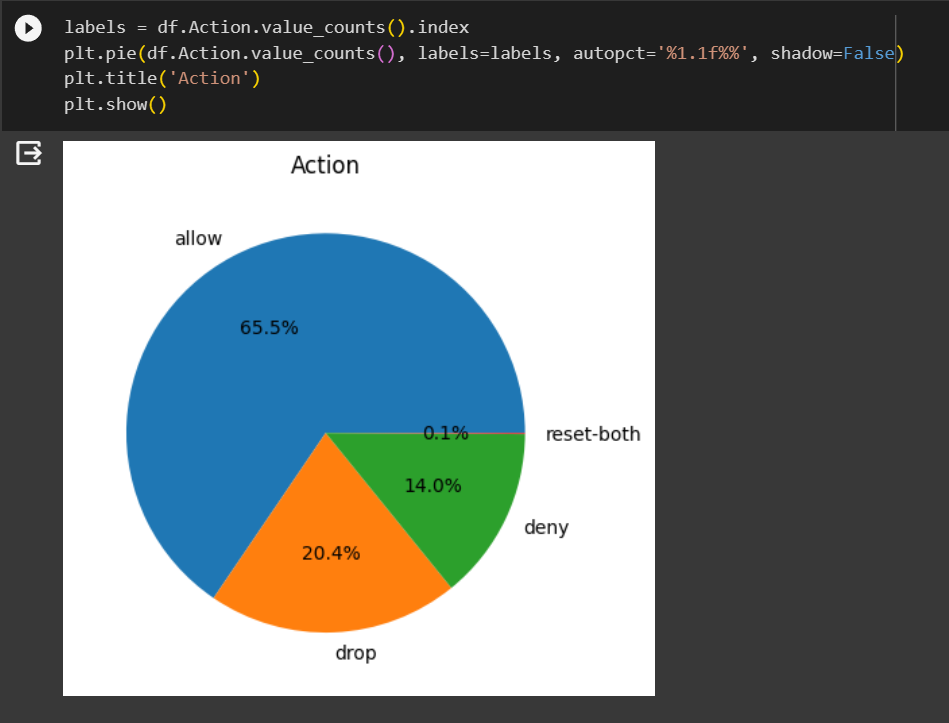


* Không có dữ liệu thiếu nào nên giá trị trả về là False.

* biểu đồ phân phối



* Biểu đồ tròn thống kê sự phân bổ giá trị



* Theo 2 biểu đồ ta thấy sự phân bổ của 4 giá trị “Allow”, “Drop”, “Deny”, “Reset-Both”:

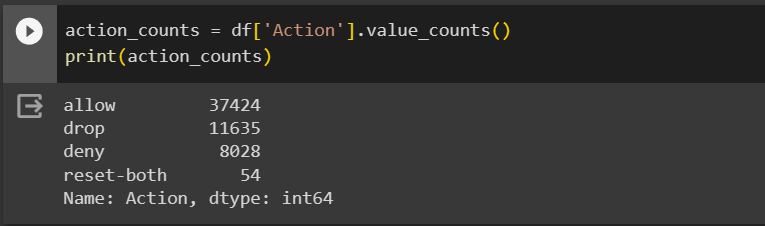
- Allow: chiếm nhiều giá trị nhất, hơn 35 ngàn dòng (65.5%) .

- Drop: chiếm hơn 10 ngàn dòng (20.4%).

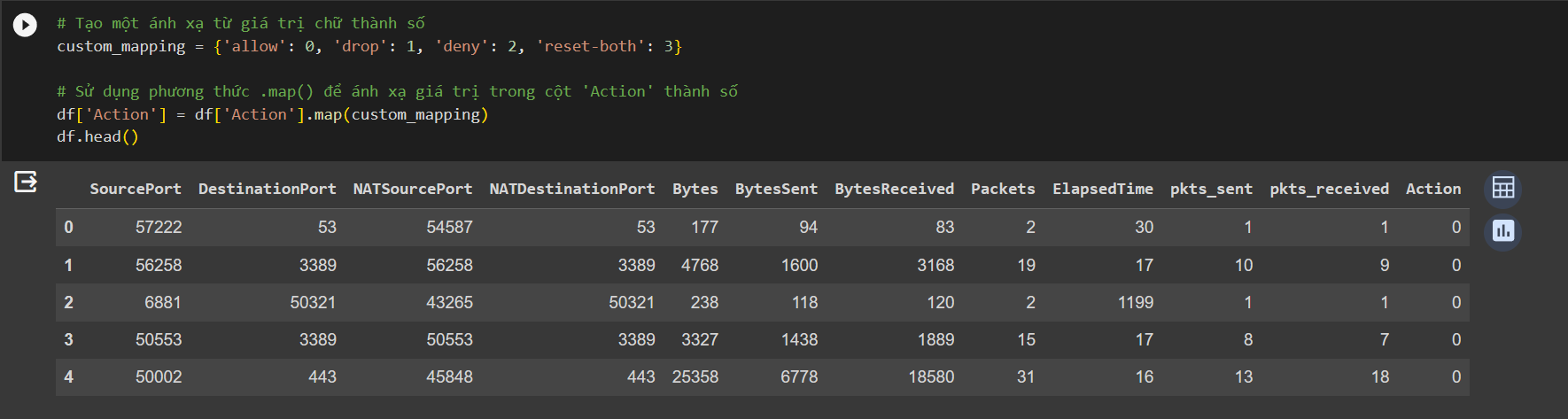
- Deny: chiếm gần 10 ngàn dòng (14.0%).

- Reset-Both: chiếm ít dòng nhất (0.1%).

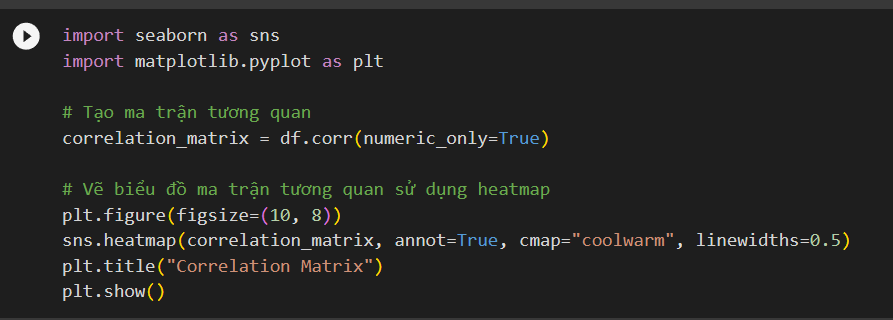
* Giá trị chính xác của từng giá trị

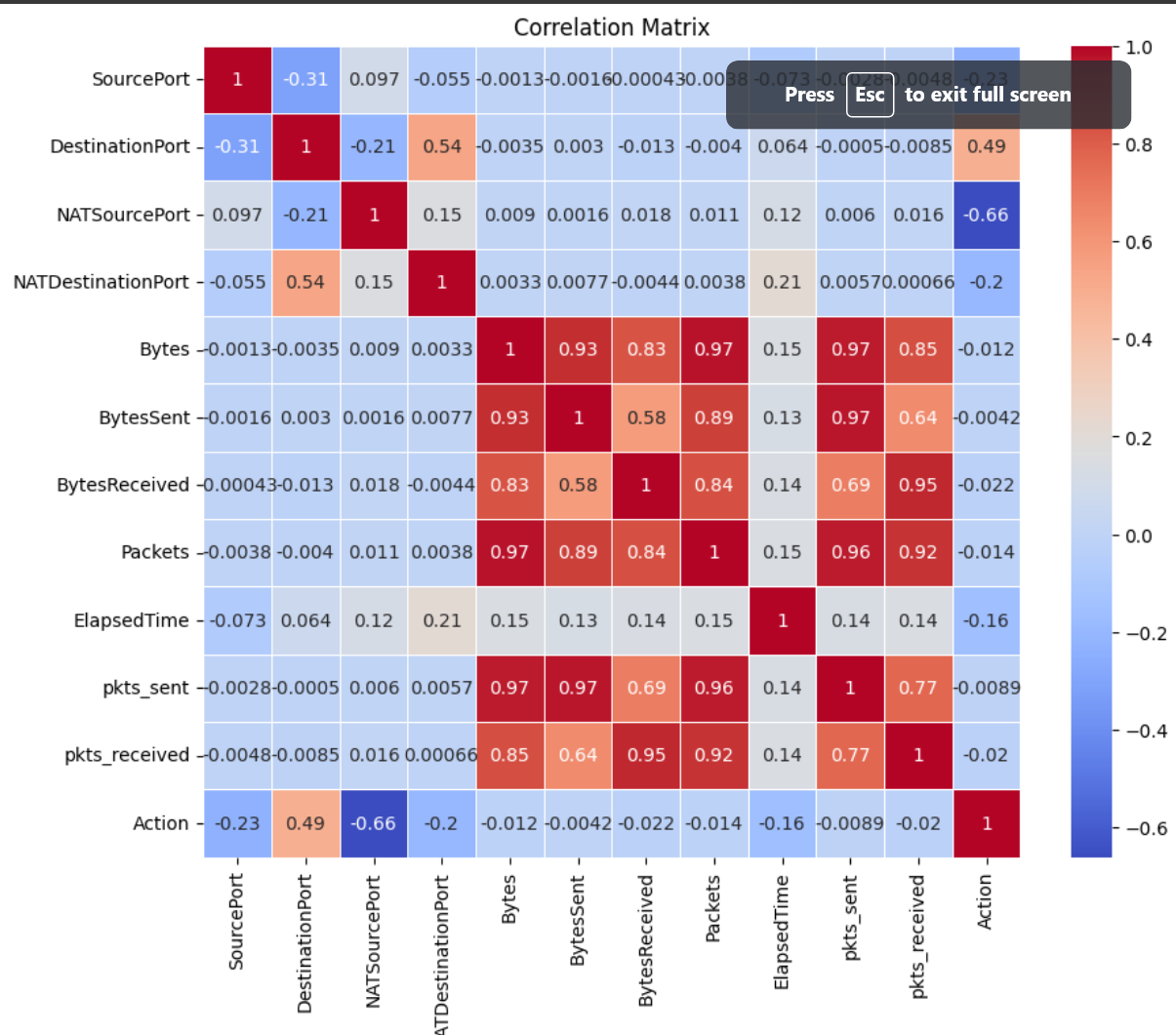


* Ánh xạ giá trị của cột Action thành số

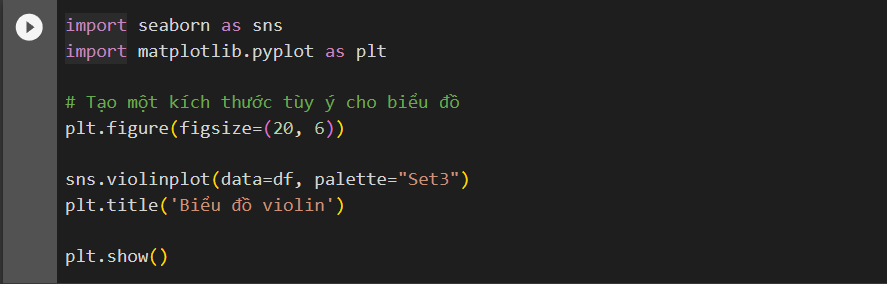


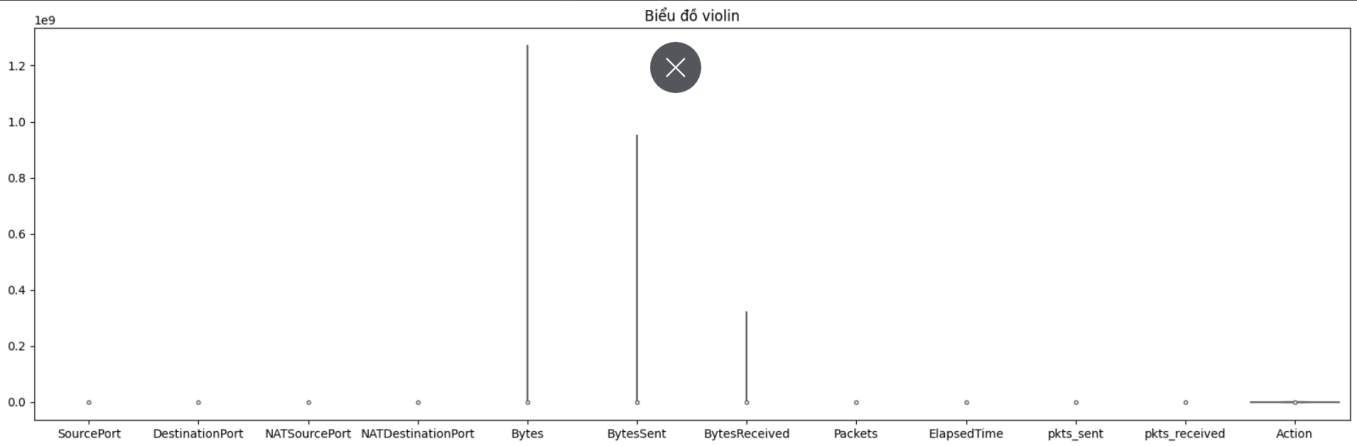
* Ta ánh xạ để tìm độ tương quan cho từng cột so với cột “Action”
* Ma trận độ tương quan của dữ liệu



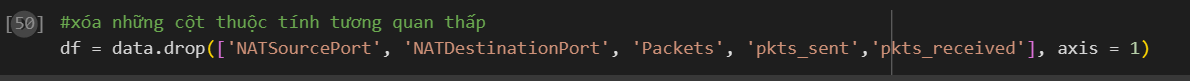


* Qua ma trận, ta nhận thấy thuộc tính “Source Port”, “Destination Port” và “Elapsed Time” có độ tương quan cao hơn so với những cái khác.
* Biểu đồ Violin kiểm tra, so sánh sự phân bố giữa các thuộc tính

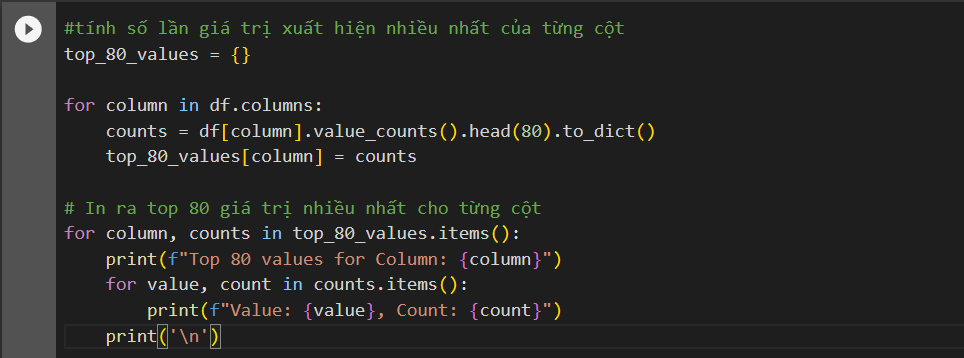




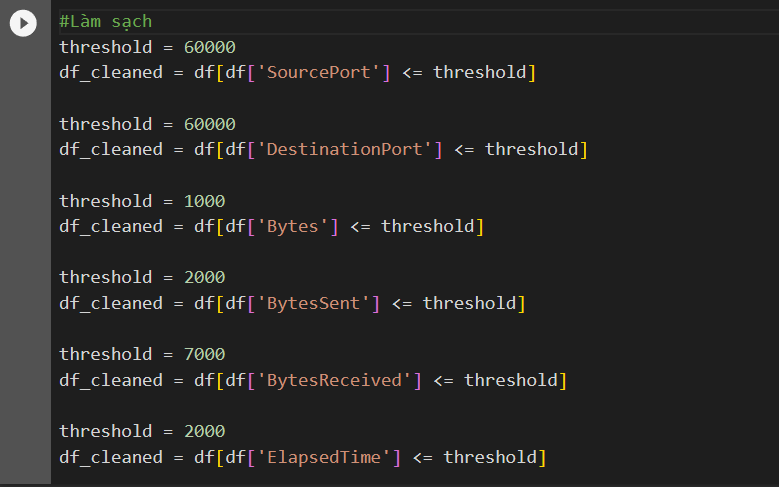
* Qua biểu đồ, ta thấy cột thuộc tính “Bytes”, “Bytes Sent” và “Bytes Received” có sự phân bố lớn, tác động lớn đến đến các giá trị trong thuộc tính “Action”.
* Qua ma trận tương quan và biểu đồ Violin, ta rút ra được các thuộc tính ảnh hưởng đến kết quả của các giá trị trong cột thuộc tính “Action”.
* Xóa đi các thuộc tính có ít độ tương quan và ảnh hưởng .



* Top 80 các giá trị xuất hiện nhiều nhất trong từng cột thuộc tính



* Loại bỏ các dữ liệu bất thường.



Với ngưỡng (threshold) đã rút ra dựa vào giá trị từng cột thuộc tính từ kết quả của thống kê dữ liệu:

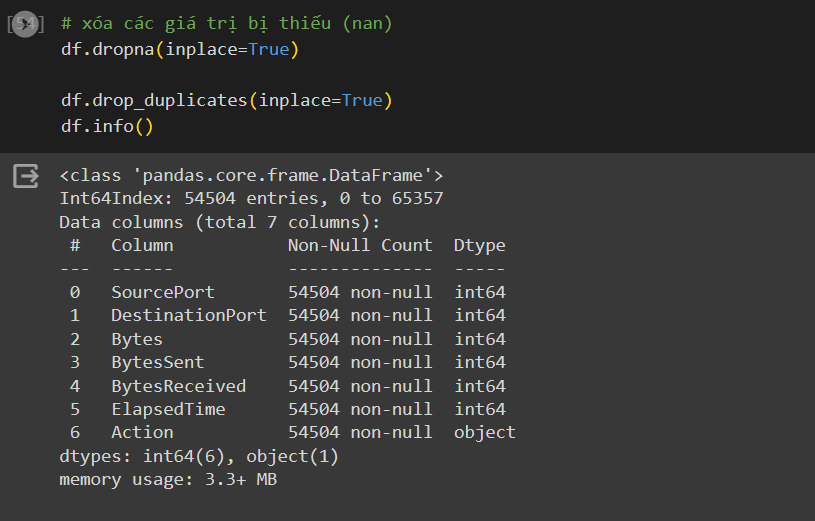
- Cột SourcePort, DestinationPort: Áp dụng ngưỡng 60000, bỏ các giá trị lớn hơn ngưỡng (vượt quá giới hạn chuẩn của cổng).

- Cột Bytes: Áp dụng ngưỡng 1000, bỏ dữ liệu lớn hơn (có thể là dữ liệu bất thường).

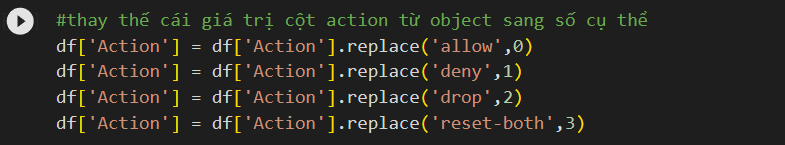
- Cột BytesSent, BytesReceived: Áp dụng ngưỡng 2000, bỏ dữ liệu lớn hơn.

- Cột ElapsedTime: Áp dụng ngưỡng 2000 , bỏ các giao dịch kéo dài hơn ngưỡng.

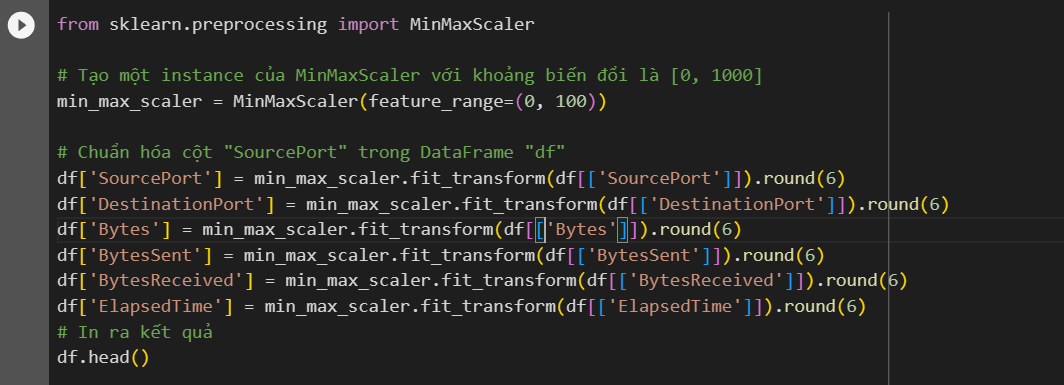
* Xóa các dữ liệu bị thiếu sau khi thực hiện bước trên.



* CHUẨN HÓA DỮ LIỆU
* Chuyển đổi dữ liệu của cột “Action” đang là kiểu Object sang dạng int64 bằng code bên dưới.



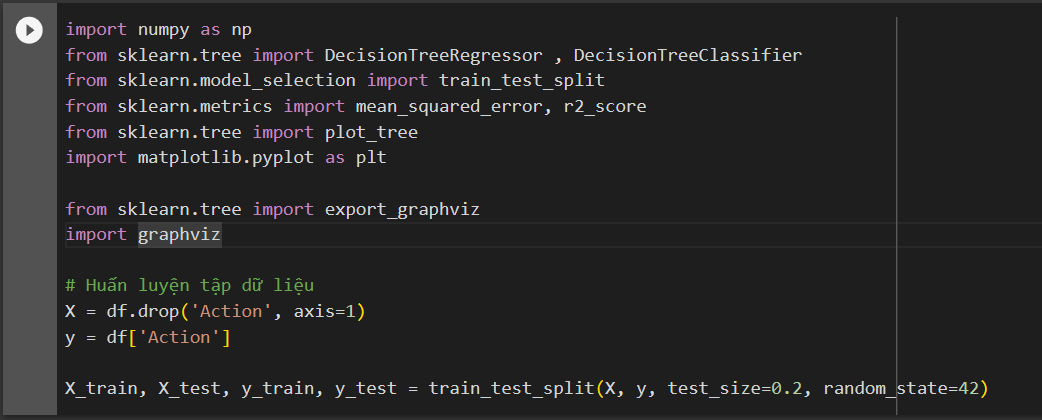
* Mục đích là để đồng bộ 1 kiểu dữ liệu int64 cho việc huấn luyện mô hình.
* Chuẩn hóa bằng phương pháp MinMaxScaler



* Mục đích để ta biến đổi dữ liệu sao cho min, max trong khoản từ 0 đến 100, đồng thời hiển thị kết quả dưới dạng 6 số thập phân thay thế vào các cột dữ liệu cũ.



* HUẤN LUYỆN MÔ HÌNH
* Khai báo thư viện và huấn luyện tập dữ liệu



* Ta chia tập dữ liệu thành tập huấn luyện và đánh giá mô hình

Trong đó:

- **X** là các biến độc lập (các cột trừ cột 'Action')

- **y** là biến phụ thuộc cần phân lớp (cột 'Action')

- **train\_test\_split()** tách tập dữ liệu thành 2 tập:

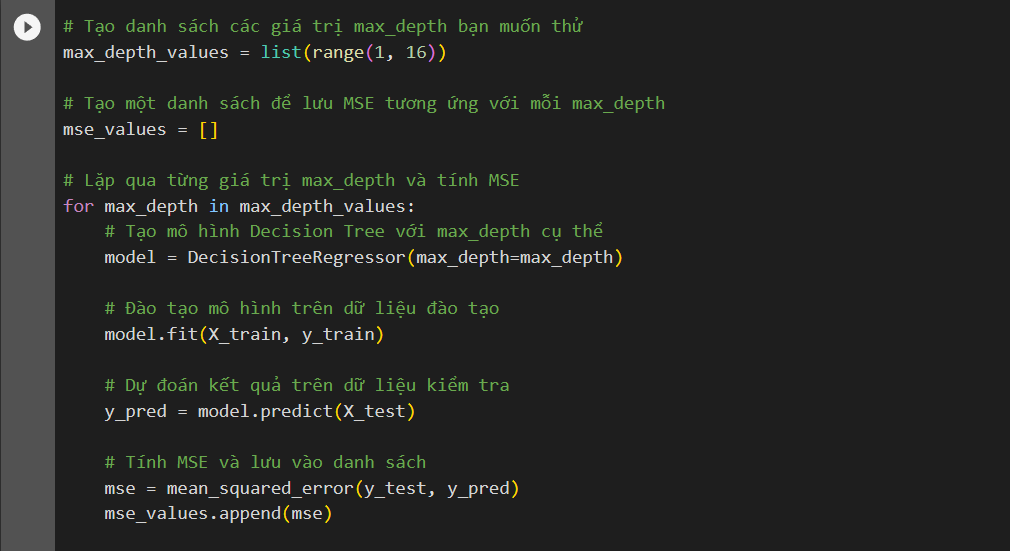
- Tập huấn luyện (**X\_train, y\_train**) để xây dựng mô hình

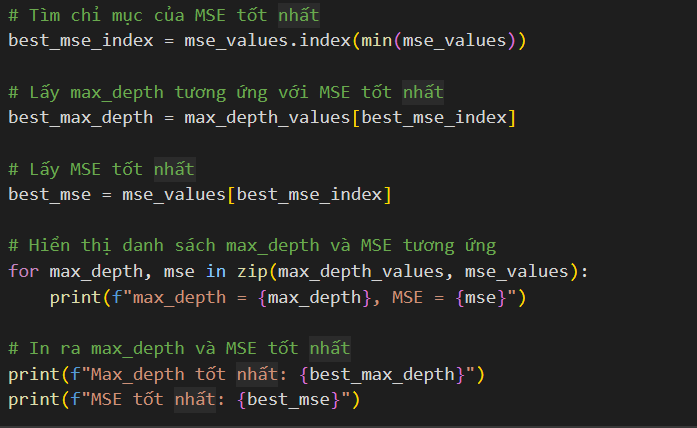
- Tập kiểm tra (**X\_test, y\_test**) để đánh giá mô hình

- **test\_size=0.2** chọn 20% dữ liệu làm tập kiểm tra

- **random\_state=42** cho kết quả lặp lại được

1. **Thuật toán Decision Tree**





- Đoạn mã này thực hiện việc chọn tham số tối ưu max\_depth cho mô hình DecisionTreeRegressor.

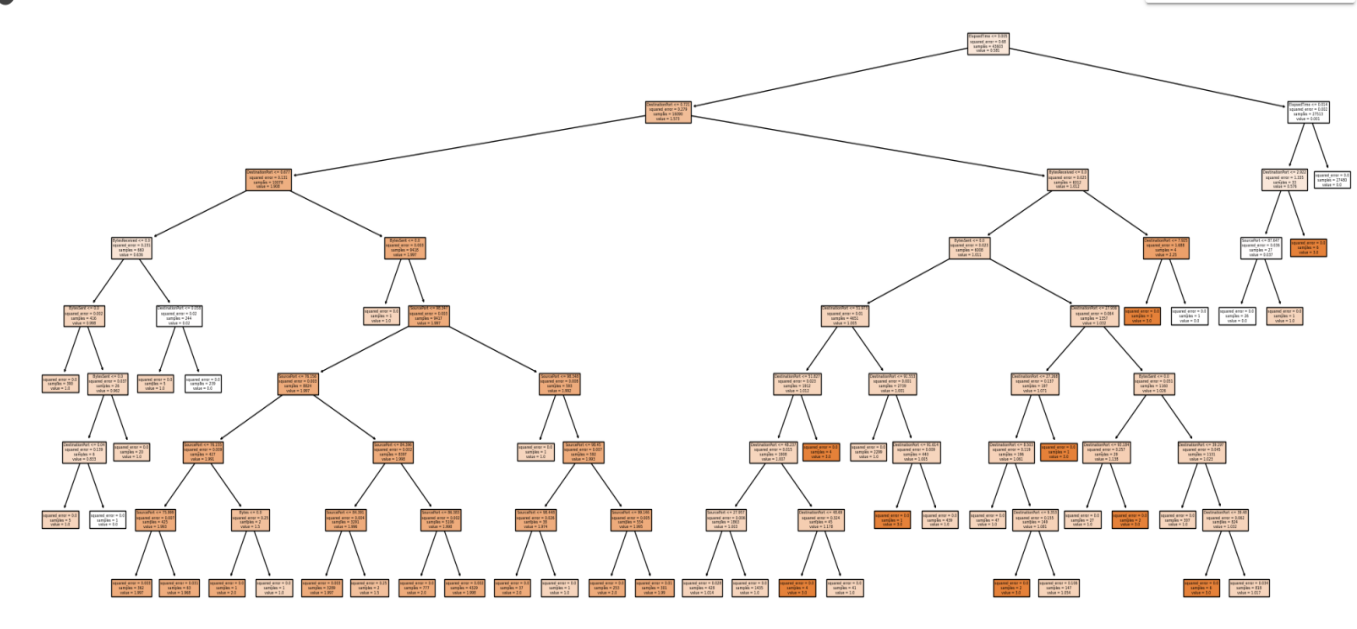
- Ta tạo một danh sách chứa các giá trị max\_depth cần thử nghiệm, từ 1 đến 15. Sau đó tạo một danh sách rỗng để lưu MSE tương ứng với mỗi giá trị max\_depth.

- Tiếp theo ta sử dụng vòng lặp để xây dựng và đào tạo mô hình với từng max\_depth cụ thể. Dự đoán kết quả trên dữ liệu test và tính toán MSE để lưu vào danh sách đã tạo. Sau khi kết thúc vòng lặp, chúng ta tìm vị trí MSE nhỏ nhất trong danh sách, lấy giá trị max\_depth tương ứng.

- Cuối cùng hiển thị danh sách max\_depth và MSE, cũng như giá trị max\_depth và MSE tối ưu nhất được chọn.

* Qua đó chúng ta xác định được giá trị max\_depth phù hợp nhất cho mô hình là 8.

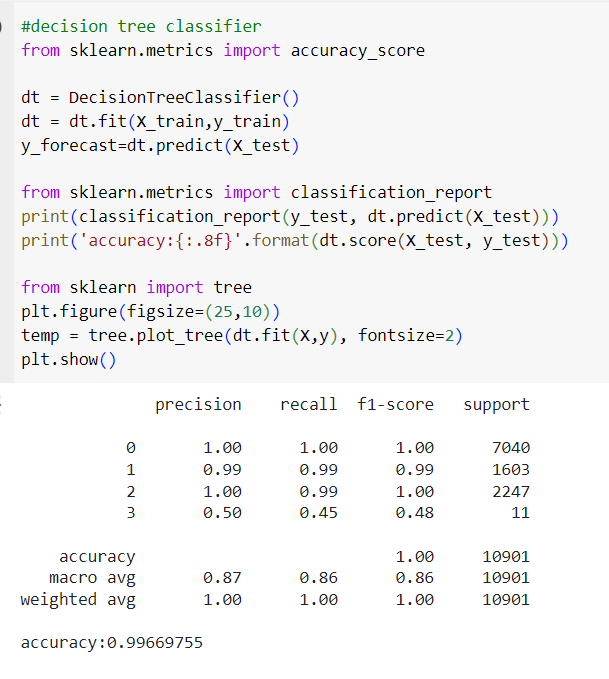
Sau khi có max\_depth là 8 ta tiến hành vẽ cây:



Dựa vào cây quyết định trên ta có thể thấy được sự ảnh hưởng giữa các thuộc tính với nhau như:

* Thuộc tính Elapsed Time ảnh hưởng lớn đến quyết định phân chia của cây nên được đặt làm gốc với 3 khoảng phân chia:
  + Elapsed Time <= 0.005: Có tới 16090 giá trị
  + 0.095<Elapsed Time <= 0.014: Có 33 giá trị.
  + Elapsed Time > 0.014: Có 27480 giá trị.

Tiếp theo ta đi đánh giá mô hình:

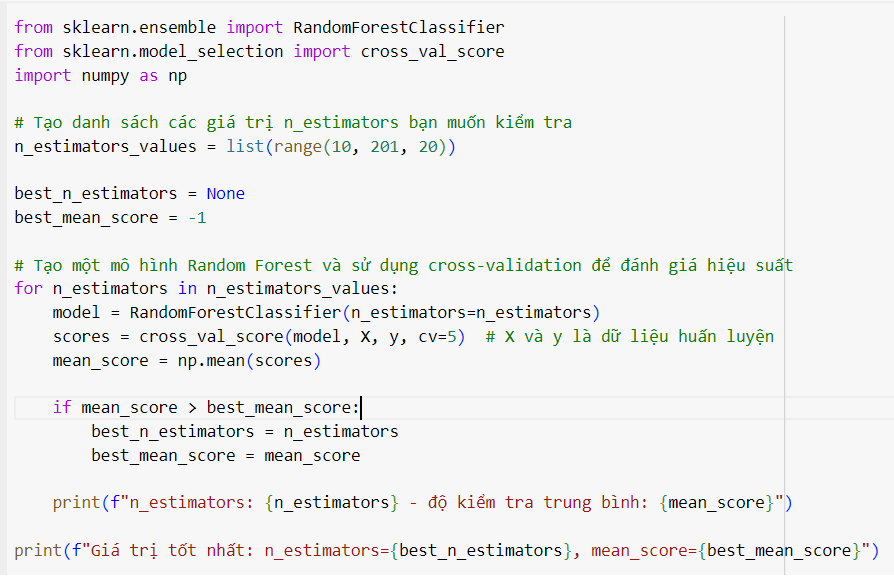


Từ hình ta có thể thấy Độ chính xác là tỷ lệ dự đoán đúng trên tổng số dự đoán do mô hình đưa ra. Trong trường hợp này, độ chính xác được báo cáo là 0,9967, có nghĩa là độ chính xác tổng thể của mô hình là khoảng 99,67%.

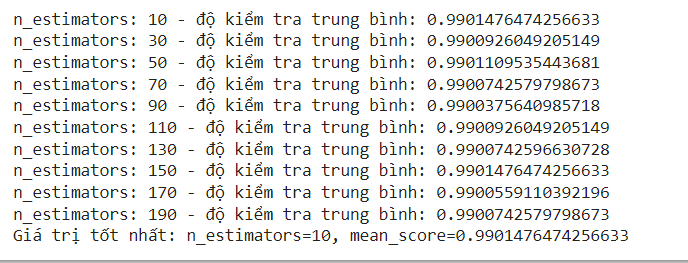
Nhìn chung, báo cáo phân loại này cho thấy mô hình đang hoạt động rất tốt, với độ chính xác, khả năng thu hồi và điểm F1 cao đối với hầu hết các lớp. Độ chính xác cũng khá cao.

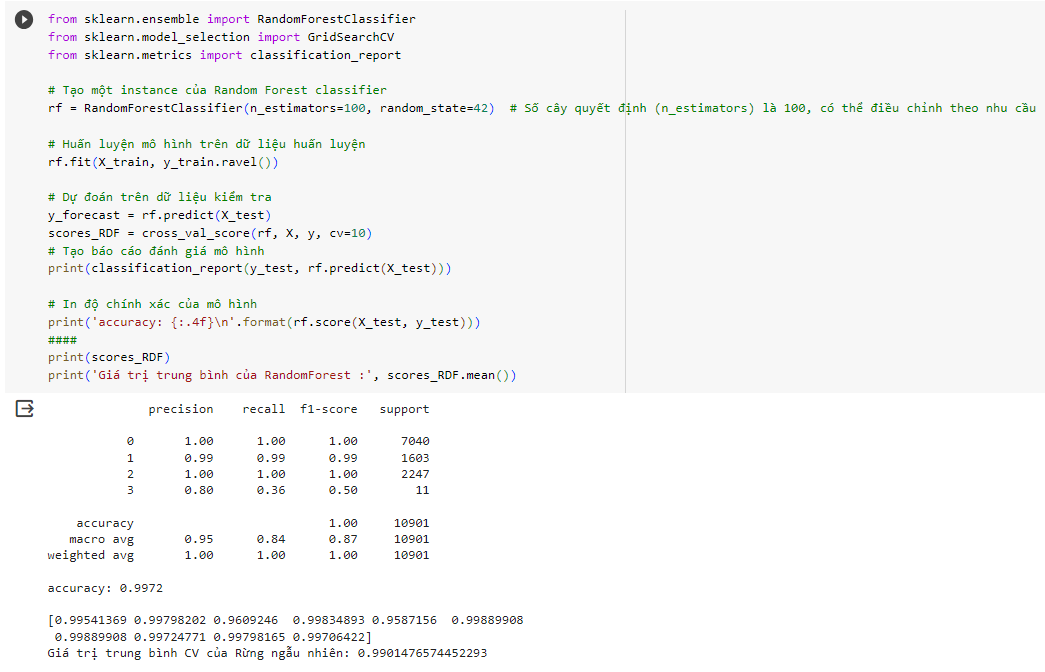
## Thuật toán Random Forest

Sau đây là code của random forest

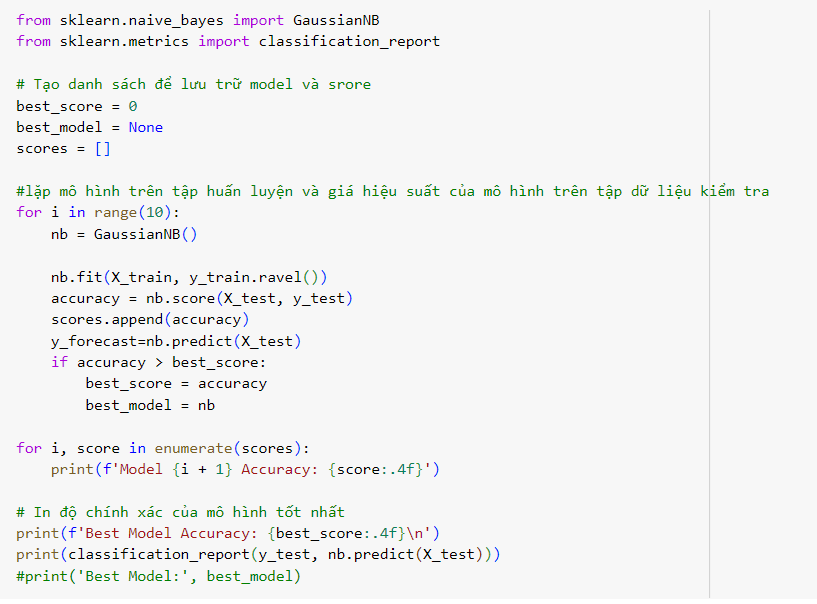


Sau khi chạy thuật toán trên ta thu được giá trị n\_estimators tốt nhất

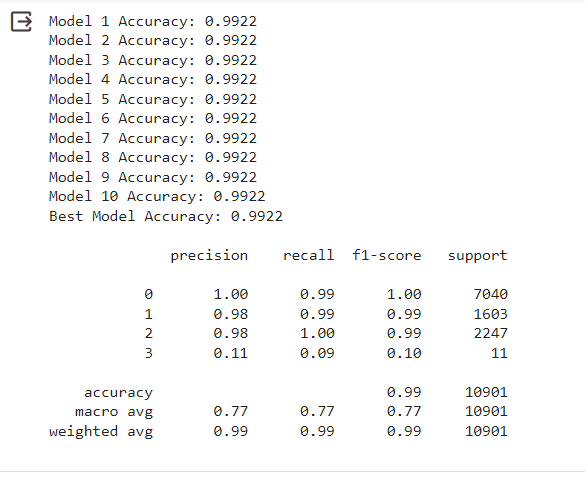




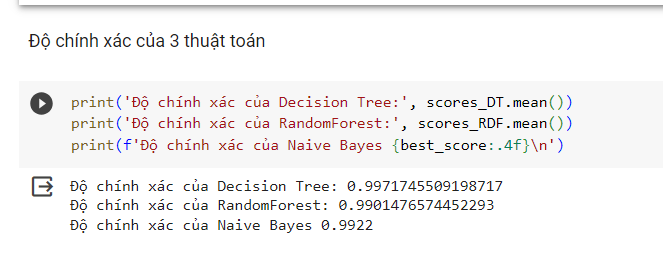
## Thuật toán Naive Bayes

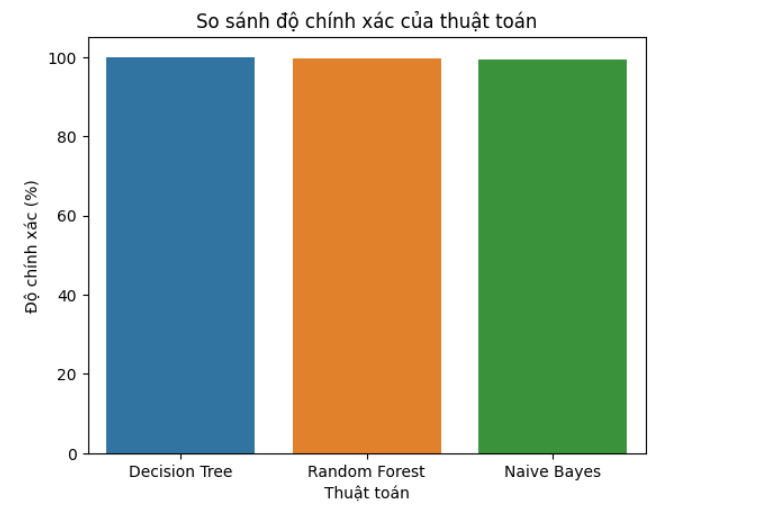


Thu được:



Sau đó ta xét độ chính xác của 3 thuật toán





# Chương IV: Kết quả đạt được

# Chương V: DEMO - Hướng phát triển