**BỘ THÔNG TIN & TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN 2**

****

**ĐỒ ÁN CUỐI KỲ**

**IoT VÀ ỨNG DỤNG**

**ĐỀ TÀI**

**VIẾT TIỆN ÍCH MỞ RỘNG (EXTENSION)**

**CÀI ĐẶT TRÊN TRÌNH DUYỆT WEB**

**PHÁT HIỆN URL ĐÁNG NGỜ HAY HỢP PHÁP**

**DỰA TRÊN MÔ HÌNH HỌC MÁY**

**THẠC SĨ, GIẢNG VIÊN HƯỚNG DẪN: ĐÀM MINH LỊNH**

**MÃ MÔN HỌC: INT1405**

**NHÓM 21**

**NGUYỄN DƯƠNG PHI – N20DCCN125**

**NGUYỄN VŨ QUANG – N20DCCN130**

**TP. HỒ CHÍ MINH, THÁNG 01 – 2024**

**– LỜI CẢM ƠN –**

– Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành và đến Học viện Công nghệ Bưu chính Viễn thông cơ sở tại Thành phố Hồ Chí Minh đã đưa bộ môn IoT và ứng dụng vào chương trình giảng dạy.

– Tiếp theo chúng em xin chân thành cảm ơn thầy Đàm Minh Lịnh trong thời gian qua đã dạy, hướng dẫn, giúp đỡ chúng em trong quá trình học tập. Giúp chúng em nắm vững kiến thức của môn học IoT và ứng dụng.

– Chúc thầy luôn luôn mạnh khỏe, luôn vui tươi, dồi dào sức sống và có nhiều thành công trong công việc giảng dạy!

– Cuối cùng chúng em xin cảm ơn gia đình, người thân và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên chúng em trong suốt quá trình học tập và thực hiện đồ án cuối kỳ này.

**Nhóm sinh viên thực hiện.**

**NHÓM 21**

**CHƯƠNG I. TỔNG QUAN VỀ ĐỀ TÀI**

Trong thời đại công nghệ số phát triển mạnh mẽ, internet đã trở thành một phần không thể thiếu trong cuộc sống của con người. Tuy nhiên, bên cạnh những tiện ích mà internet mang lại, cũng có không ít những mối đe dọa về an toàn thông tin. Một trong những mối đe dọa phổ biến nhất là các trang web độc hại, có thể gây ra các thiệt hại như đánh cắp thông tin cá nhân, lừa đảo tài chính,...

Để giúp người dùng nhận diện và tránh truy cập vào các trang web độc hại, một giải pháp hiệu quả là sử dụng các tiện ích mở rộng (extension) của trình duyệt web. Các tiện ích mở rộng này thường sử dụng các thuật toán học máy để phân tích URL và xác định xem URL đó có đáng ngờ hay không.

Tiện ích mở rộng được thiết kế để tích hợp trực tiếp vào trình duyệt web, tận dụng khả năng của mô hình học máy để phân loại URL là đáng ngờ hay hợp pháp. Nhờ vào khả năng học và điều chỉnh từ dữ liệu người dùng và các nguồn thông tin đáng tin cậy, tiện ích mở rộng này không chỉ giúp người dùng phòng tránh các trang web độc hại mà còn nâng cao khả năng nhận biết đối với các trang web mới và nguy cơ tiềm ẩn.

Với sự kết hợp giữa công nghệ tiên tiến và tính ứng dụng, đề tài này mang lại giải pháp hiệu quả để cải thiện an ninh trực tuyến, giảm rủi ro từ các đường link độc hại và bảo vệ người dùng khi sử dụng trình duyệt web.

**CHƯƠNG II. XÂY DỰNG CHƯƠNG TRÌNH**

## 1. XÂY DỰNG FILE NOTEBOOK: (DetectingPhishingURL.ipynb)

### 1.1. GOOGLE COLAB:

Google Colaboratory, hay còn gọi là Colab, là một dịch vụ cung cấp môi trường làm việc trực tuyến (online environment) để thực hiện các dự án liên quan đến khoa học dữ liệu và máy học. Được phát triển bởi Google Research, Colab đưa ra một cách tiếp cận đơn giản và hiệu quả cho việc phân tích dữ liệu và xây dựng mô hình máy học mà không yêu cầu cài đặt trực tiếp trên máy tính cá nhân.



Một số đặc điểm nổi bật của Google Colab bao gồm:

– Colab cung cấp một môi trường làm việc miễn phí và dễ sử dụng trực tuyến, không yêu cầu bất kỳ cài đặt phức tạp nào. Người dùng chỉ cần có tài khoản Google để bắt đầu sử dụng.

– Colab tích hợp chặt chẽ với Google Drive, cho phép người dùng lưu trữ, chia sẻ và quản lý các notebook (văn bản mã nguồn) của mình một cách thuận tiện.

– Một trong những ưu điểm lớn nhất của Colab là khả năng sử dụng miễn phí các tài nguyên GPU (Graphics Processing Unit) và TPU (Tensor Processing Unit) từ Google, giúp tăng tốc quá trình huấn luyện mô hình máy học.

– Colab cho phép người dùng chia sẻ notebook của mình với bất kỳ ai thông qua liên kết duy nhất hoặc tích hợp vào GitHub, tạo điều kiện thuận lợi cho làm việc nhóm và trình bày công việc.

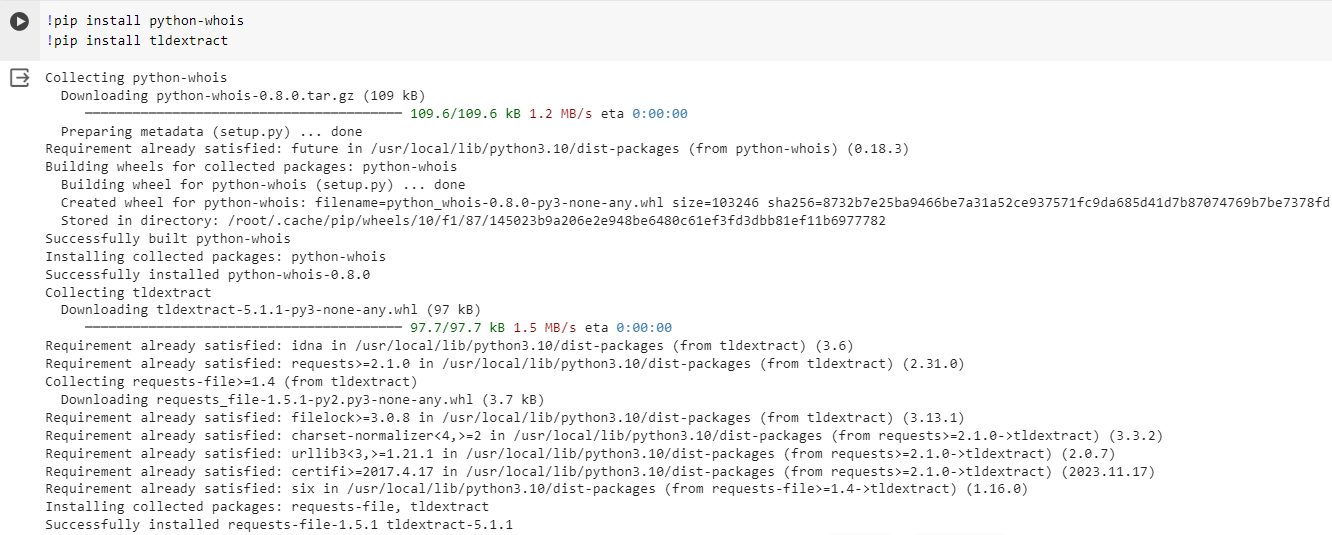
– Colab hỗ trợ nhiều ngôn ngữ lập trình như Python, R, và Scala, giúp người dùng linh hoạt trong việc sử dụng ngôn ngữ ưa thích của họ.

Với những tính năng và lợi ích nổi bật này, Google Colaboratory đã trở thành một công cụ quan trọng trong cộng đồng khoa học dữ liệu và máy học, hỗ trợ người dùng thực hiện các dự án phức tạp mà không phải lo lắng về cấu hình phần cứng hay môi trường lập trình.

### 1.2. XÂY DỰNG:

# Cài thư viện lấy thông tin về tên miền từ dịch vụ whois.

# Cài thư viện phân tích các thành phần của tên miền, gồm phần tên miền cấp cao nhất (TLD), tên miền chính (domain), và tên miền con (subdomain).

****

**# pandas:** sử dụng cho xử lý và phân tích dữ liệu.

**# numpy:** sử dụng để thực hiện phép toán số học & xử lý mảng nhiều chiều.

**# sys:** cung cấp các chức năng và biến liên quan đến hệ thống.

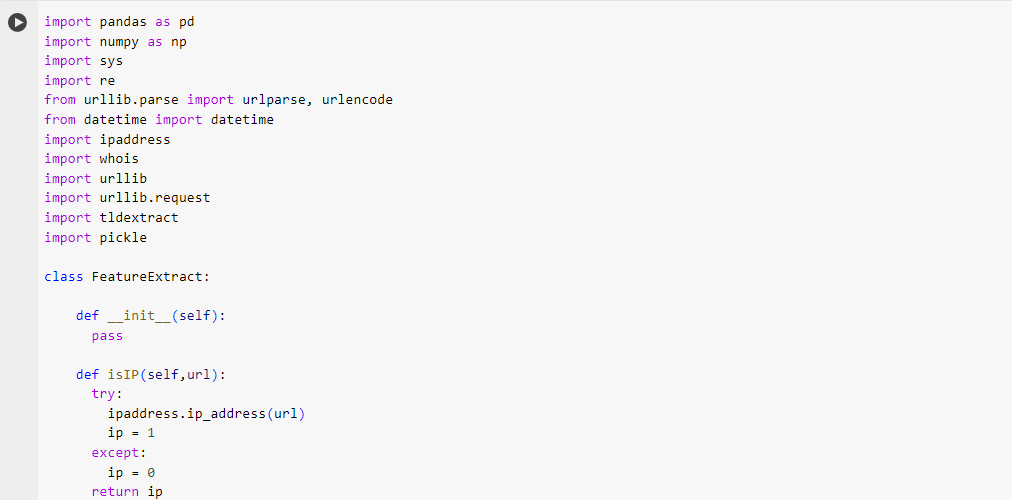
**# urllib.parse:** module của thư viện urllib, được sử dụng để phân tích các thành phần của URL như scheme, netloc, path, params, query, và fragment.

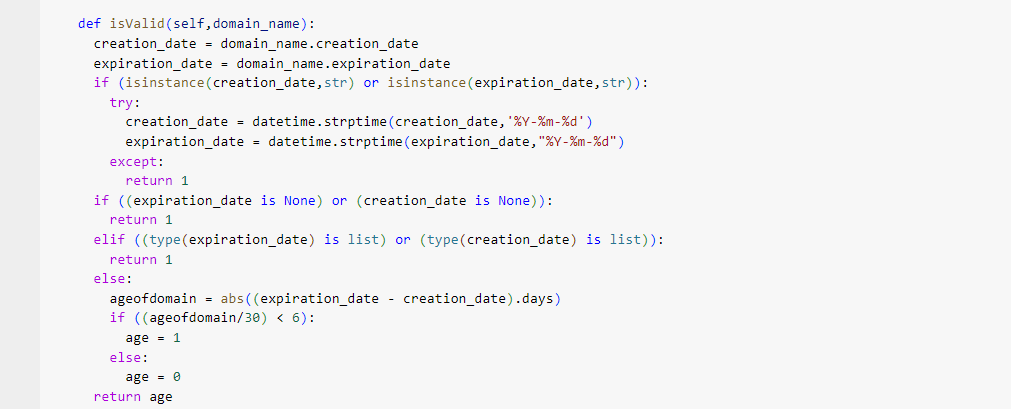
**# datetime:** cung cấp lớp & chức năng liên quan đến thời gian & ngày tháng.

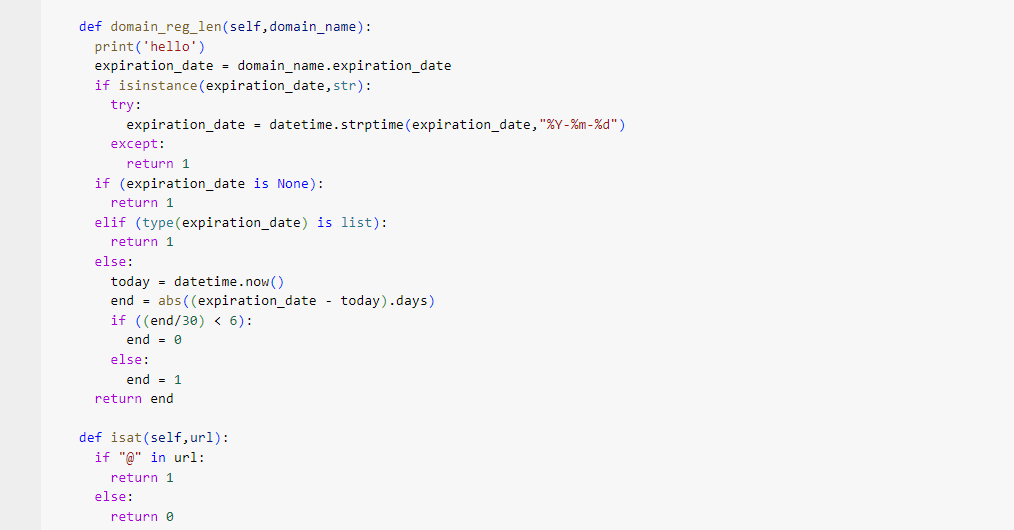
**# ipaddress:** cung cấp các công cụ để xử lý địa chỉ IP và mạng.

**# urllib:** thực hiện các hoạt động liên quan đến URL, như mở các đối tượng URL, thực hiện các yêu cầu HTTP, và xử lý URL.

**# pickle:** sử dụng để lưu trữ và khôi phục mô hình đã được huấn luyện hoặc các đối tượng khác.

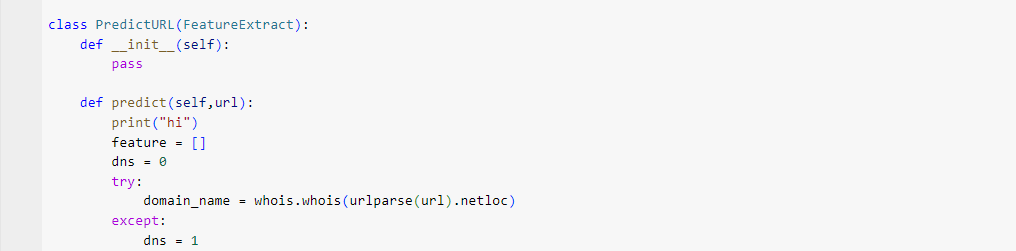
****

****

****

****

****

****

****

**Lớp FeatureExtract:**

–Nếu địa chỉ IP được sử dụng thay thế cho tên miền trong URL, chẳng hạn như “<http://125.98.3.123/fake.html>”, người dùng có thể chắc chắn rằng ai đó đang cố lấy cắp thông tin cá nhân của họ. Đôi khi, địa chỉ IP thậm chí còn được chuyển đổi thành mã thập lục phân như trong liên kết sau “<http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html>”.

**isIP():** Phương thức kiểm tra xem có địa chỉ IP tồn tại trong URL không.

Nếu có trả về 1 (đáng ngờ), không có trả về 0 (hợp pháp).

**isValid():** Phương thức kiểm tra tính hợp lệ của URL.

Nếu tuổi tên miền < 6 tháng, trả về 1 (đáng ngờ), ngược lại trả về 0 (hợp pháp).

– Dựa trên thực tế là một trang web đáng ngờ tồn tại trong một khoảng thời gian ngắn, các miền đáng tin cậy thường được trả trước vài năm. Trong tập dữ liệu dataset, nhận thấy rằng các miền đáng ngờ lâu nhất chỉ được sử dụng trong một năm.

**domain\_reg\_len():** So sánh thời gian hết hạn với thời điểm hiện tại.

Nếu tên miền đã hết hạn, trả về 1 (đáng ngờ), ngược lại trả về 0 (hợp pháp).

– Việc sử dụng ký hiệu “@” trong URL sẽ khiến trình duyệt bỏ qua mọi thứ trước ký hiệu “@” và địa chỉ thực thường theo sau ký hiệu “@”.

**isat():** Kiểm tra xem URL có chứa '@' hay không.

Nếu có '@' trả về 1 (đáng ngờ), ngược lại trả về 0 (hợp pháp).

– Sự tồn tại của “//” trong đường dẫn URL có nghĩa là người dùng sẽ được chuyển hướng đến một trang web khác. Một ví dụ về URL như vậy là:

“[http://www.legitimate.com//http://www.phishing.com](http://www.legitimate.com/http:/www.phishing.com)”. Kiểm tra vị trí xuất hiện “//” thấy rằng nếu URL bắt đầu bằng “HTTP”, điều đó có nghĩa là “//” sẽ xuất hiện ở vị trí thứ 6. Tuy nhiên, nếu URL sử dụng “HTTPS” thì “//” sẽ xuất hiện ở vị trí thứ 7.

**isRedirect():** Kiểm tra URL chứa “//” liền kề trong ngữ cảnh của “http://” hoặc “https://” không.

Nếu có trả về 1 (đáng ngờ), ngược lại trả về 0 (hợp pháp).

– Biểu tượng “–” hiếm khi được sử dụng trong các URL hợp pháp. Những kẻ đáng ngờ có xu hướng thêm tiền tố hoặc hậu tố được phân tách bằng “–” vào tên miền để người dùng cảm thấy rằng họ đang truy cập một trang web hợp pháp. Ví dụ: “<http://www.Confirme–paypal.com/>”.

**haveDash():** Kiểm tra xem có chứa '-' trong phần netloc của URL không.

Nếu có trả vê 1 (đáng ngờ), nếu không trả về 0 (hợp pháp).

– Giả sử ta có liên kết sau: “<http://www.hud.ac.uk/students/>”. Một tên miền có thể bao gồm các tên miền cấp cao nhất theo mã quốc gia (ccTLD), trong ví dụ là “uk”. Phần “ac” là viết tắt của “academic”, phần “ac.uk” được gọi là tên miền cấp hai (SLD) và “hud” là tên thật của miền. Để tạo quy tắc trích xuất tính năng này, trước tiên phải bỏ qua (www.) khỏi URL, trên thực tế, bản thân nó là một tên miền phụ. Sau đó phải xóa (ccTLD) nếu nó tồn tại. Cuối cùng, đếm các dấu chấm còn lại.

**no\_sub\_domain():** Kiếm tra xem URL có chưa nhiều hơn một tên miền phụ hay không.

Số lượng tên miền là 1, trả về 0 (hợp pháp), ngược lại trả về 1 (đáng ngờ).

– Những kẻ lừa đảo có thể thêm mã thông báo “HTTPS” vào phần tên miền của URL để lừa người dùng. Ví dụ: <http://https-www-paypal-it-webapps-mpp-home.soft-hair.com/>.

**httpDomain():** Kiểm tra xem có chứa “http” trong phần netloc của URL hay không.

Nếu có trả về 1 (đáng ngờ), nếu không trả về 0 (hợp pháp).

– Những kẻ đáng ngờ có thể sử dụng URL dài để ẩn phần đáng ngờ trên thanh địa chỉ. Ví dụ:

“[http://federmacedoadv.com.br/3f/aze/ab51ee1e51502f46b773a5e/?cmd=\_home&amp;dispatch=11004d58f5b74f8dc1e7c2e8d811004585b74dc1e7c2e8dd4105e 8@phishing.website.html](http://federmacedoadv.com.br/3f/aze/ab51ee1e51502f46b773a5e/?cmd=_home&amp;dispatch=11004d58f5b74f8dc1e7c2e8d811004585b74dc1e7c2e8dd4105e%208@phishing.website.html)”

Để đảm bảo tính chính xác, chúng em đã tính toán độ dài URL trong tập dữ liệu và tạo ra độ dài URL trung bình. Kết quả cho thấy nếu độ dài của URL lớn hơn hoặc bằng 54 ký tự thì URL đó được phân loại là đáng ngờ. Bằng cách xem xét tập dữ liệu, có thể tìm thấy 1220 URL có độ dài từ 54 trở lên, chiếm 48,8% tổng kích thước tập dữ liệu.

**LongURL():** Kiểm tra độ dài của URL.

Nếu độ dài < 54 kí tự , trả về 0 (hợp pháp), ngược lại trả về 1 (đáng ngờ).

– Rút gọn URL là một phương pháp trên “World Wide Web”, trong đó URL có thể được làm cho có độ dài nhỏ hơn đáng kể mà vẫn dẫn đến trang web được yêu cầu. Điều này được thực hiện bằng cách “Chuyển hướng HTTP” trên một tên miền ngắn, liên kết đến trang web có URL dài. Ví dụ: URL “<http://portal.hud.ac.uk/>” có thể được rút ngắn thành “<bit.ly/19DXSk4>”.

**tinyURL():** Kiểm tra xem URL có thuộc về các dịch vụ rút gọn URL hay không.

Nếu có trả về 1 (đáng ngờ), ngược lại trả về 0 (hợp pháp).

**Lớp PredictURL:**

**predict() :** Dự đoán xem URL có phải là PHISHING hay LEGITIMATE.

**:** Nhận vào một URL làm đối số.

**\_\_getstate\_\_():** Xác định những thuộc tính sẽ được lưu trữ trong file pickle

**\_\_setstate\_\_():** Khôi phục các thuộc tính từ dữ liệu được lưu trữ trong file pickle.

**classify():** Phân loại URL dựa trên các đặc trưng sử dụng mô hình đã đào tạo trước đó.

**Import thư viện và Module Machine Learning:**

**train\_test\_split:** cung cấp công cụ cho việc chia dữ liệu thành tập huấn luyện & kiểm tra.

**DecisionTreeClassifier:** Mô hình cây quyết định trong scikit-learn, sử dụng cho việc phân loại.

**RandomForestClassifier:** Mô hình Random Forest trong scikit-learn, một phương pháp học máy ensemble.

**LogisticRegression:** Mô hình hồi quy logistic, sử dụng cho phân loại nhị phân.

**MLPClassifier:** Mô hình mạng nơron đa tầng (Multilayer Perceptron) cho phân loại.

**BernoulliNB:** Mô hình Naive Bayes cho phân loại với biến ngẫu nhiên Bernoulli.

**XGBClassifier:** Mô hình XGBoost cho phân loại.

**SVC:** Mô hình Support Vector Classifier trong scikit-learn.

**Import thư viện và Module đánh giá hiệu suất:**

**seaborn:** Thư viện để vẽ đồ thị thống kê và biểu đồ.

**matplotlib.pyplot:** Thư viện vẽ đồ thị và biểu đồ.

**%matplotlib inline:** Magic command để hiển thị đồ thị trực tiếp trong notebook.

**timeit:** Thư viện để đo thời gian thực thi mã.

**math.sqrt:** Hàm tính căn bậc hai.

**Các hàm và công cụ đánh giá hiệu suất:**

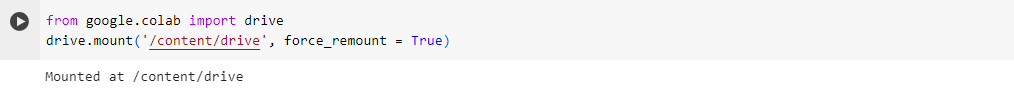
**sklearn.metrics:** Các hàm và công cụ để tính các độ đo như confusion-matrix, precision-recall-curve, ROC-curve, accuracy, cohen's kappa, …

**sklearn.model\_selection.cross\_val\_score:** Hàm để thực hiện cross-validation.

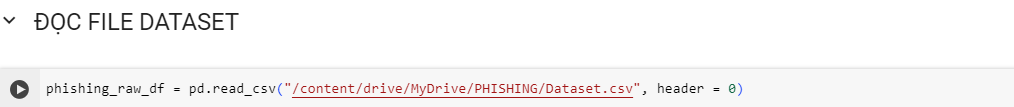
**sklearn.model\_selection.GridSearchCV:** Hàm thực hiện tìm kiếm siêu tham số trên lưới.

****

# Kết nối Google Colab với Google Drive.

****

# Sử dụng thư viện pandas để đọc tệp tin dữ liệu “Dataset.csv”.

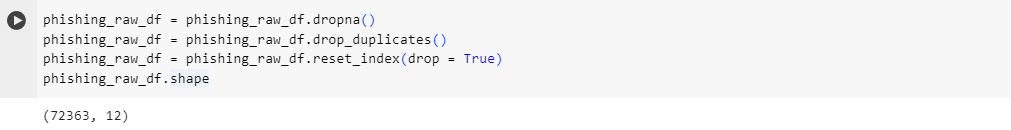
****

**TIỀN XỬ LÝ DỮ LIỆU:**

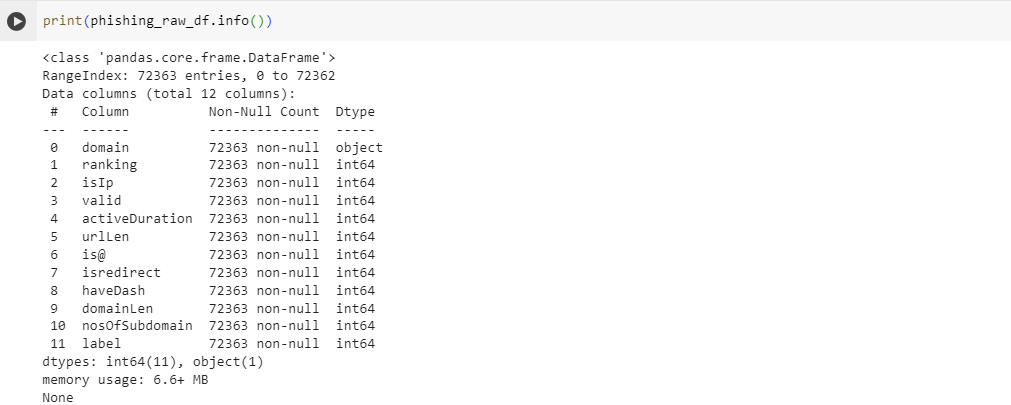
# Loại bỏ các dòng có giá trị NaN (thiếu dữ liệu) & các dòng trùng lặp.

# Đặt lại chỉ số của DataFrame sau khi xóa dòng hoặc cột.

# In ra kích thước mới của DataFrame (hàng, cột)



# Hiển thị thông tin về DataFrame.

****

# Đếm số lượng các giá trị trong cột 'label' của DataFrame

# In ra số lượng URL thuộc lớp "Legitimate" và "Phishing".

****# Vẽ biểu đồ.

# Thực hiện 'undersampling' để xử lý mất cân bằng giữa 2 lớp.

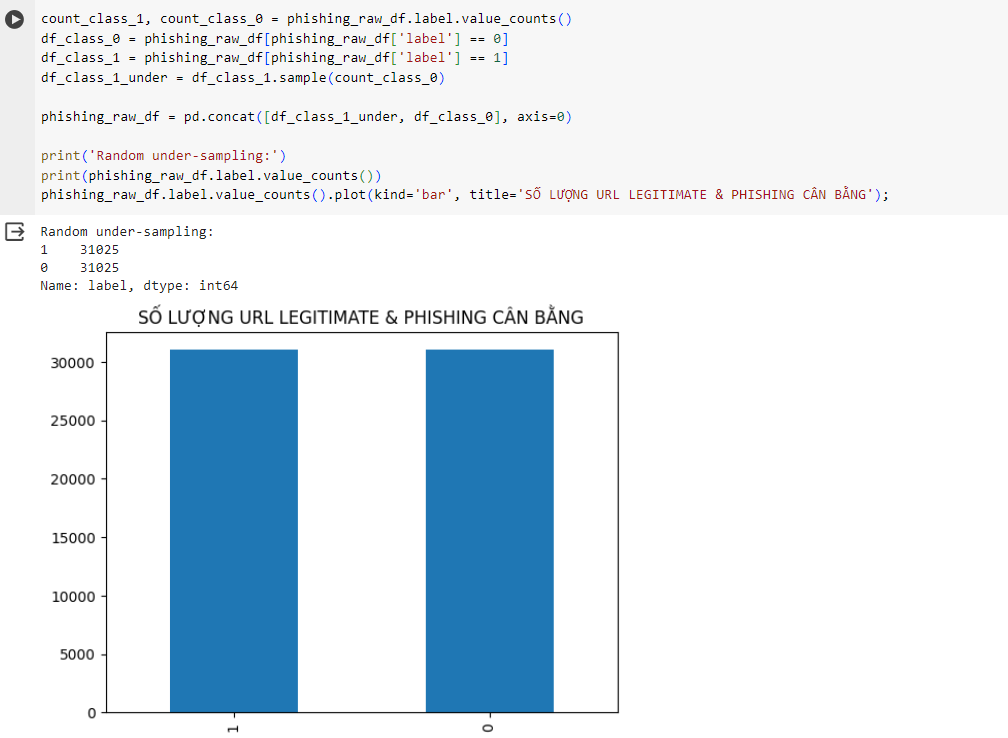
# Đếm số lượng các mẫu trong mỗi lớp của tập dữ liệu.

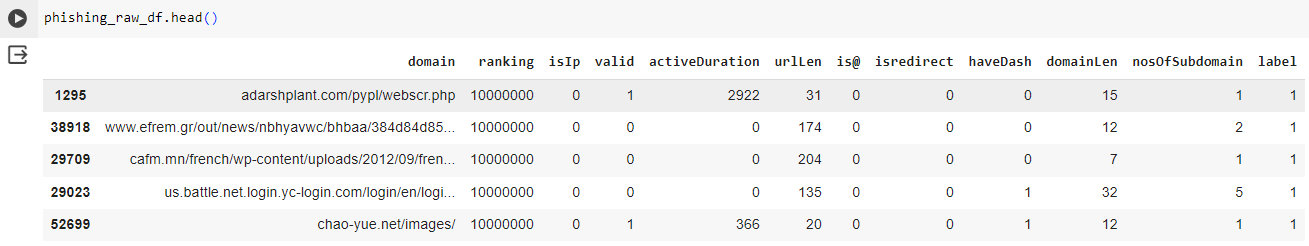
# Tạo 2 DataFrame mới chứa các mẫu ứng với lớp 0 và lớp 1 từ DataFrame gốc.

# Label 1 > 0. Thực hiện undersampling trên lớp 1

# (Lấy ngẫu nhiên một số mẫu từ lớp 1 để giảm xuống cùng số mẫu lớp 0).

# Tạo ra một DataFrame mới với số lượng mẫu bằng nhau cho cả hai lớp.

****

****

**CHUYỂN ĐỔI DỮ LIỆU:**

# Gọi các phương thức của lớp FeatureExtract để trích xuất các đặc trưng từ URL.

****

# Tạo cột mới có tên "Domain\_http".

# Giá trị được xác định bằng cách áp dụng hàm feature.httpDomain(x).

# Tạo cột mới có tên "LongURL" để chỉ độ dài của URL.

# Giá trị được xác định bằng cách áp dụng hàm feature.LongURL(x).

# Tạo cột mới có tên "TinyURL" để xem URL có thuộc các dịch vụ rút gọn.

# Giá trị của cột được xác định bằng cách áp dụng hàm feature.tinyURL(x)

# Sửa đổi giá trị trong cột "nosOfSubdomain" của DataFrame.

# Nếu số lượng subdomain lớn hơn 1, dự đoán là PHISHING.

# Sửa đổi giá trị trong cột "activeDuration" của DataFrame.

# Nếu giá trị lớn hơn 365 (1 năm), giả định là LEGITIMATE.

# Sửa đổi giá trị trong cột "ranking" của DataFrame.

# Nếu giá trị lớn hơn 100000, giả định là trang web PHISHING.

# Bỏ các cột không cần thiết ('urlLen', 'domainLen', 'domain').

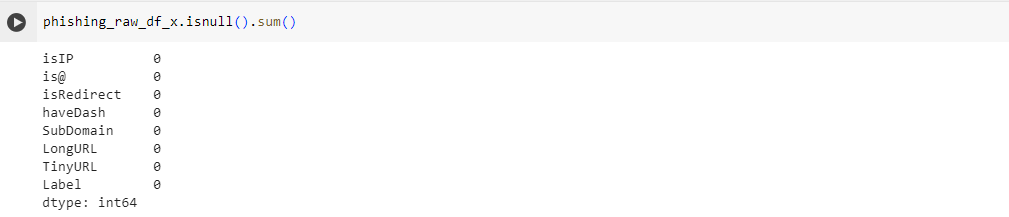
# Tạo một bản sao DataFrame và loại bỏ một số cột không cần thiết.

# Trộn ngẫu nhiên các dòng dữ liệu để đảm bảo phân phối đều khi chia tập train và test.

# Lưu DataFrame đã được tiền xử lý vào file 'Phishing\_Feature\_Engg.csv'

****

# Đếm số giá trị thiếu (NaN) trong cột DataFrame

****

**THỐNG KÊ DỮ LIỆU:**

# Tạo ra một tóm tắt thống kê của các giá trị trong DataFrame.

# mean: Trung bình cộng của các giá trị trong cột.

# std: Độ lệch chuẩn của các giá trị trong cột.

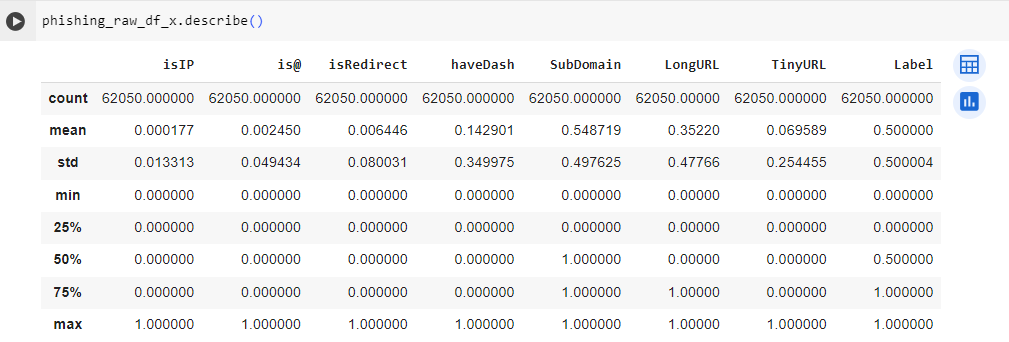
# min: Giá trị nhỏ nhất trong cột.

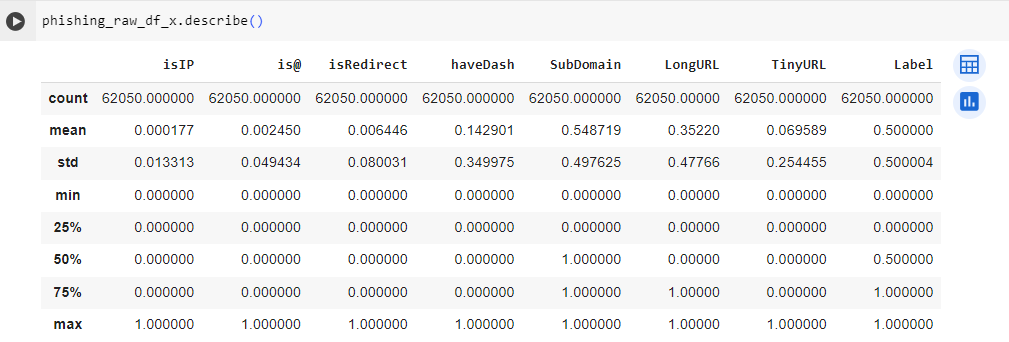
# 25%: Giá trị thứ 25 (phân vị thứ 1/4) trong cột.

# 50%: Giá trị thứ 50 (phân vị thứ 2/4) trong cột.

# 75%: Giá trị thứ 75 (phân vị thứ 3/4) trong cột.

# max: Giá trị lớn nhất trong cột.

****

****

# Tạo biểu đồ HeatMap.

# Hiển thị giá trị tương quan trên từng ô của HeatMap.

****

****

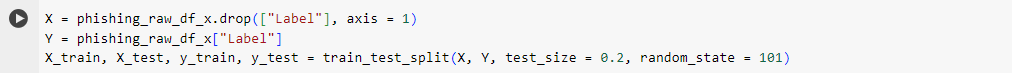
# Loại bỏ cột "Label" khỏi DataFrame và lưu phần còn lại vào biến X.

# Lưu trữ cột "Label" vào biến Y.

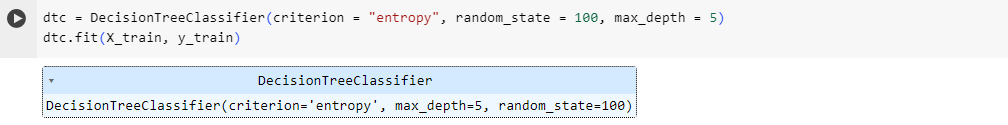
# Chia tập dữ liệu làm 2 phần:

# X\_train & y\_train là dữ liệu huấn luyện, chiếm 80% dữ liệu ban đầu.

# X\_test & y\_test là dữ liệu kiểm thử, chiếm 20% dữ liệu ban đầu.

****

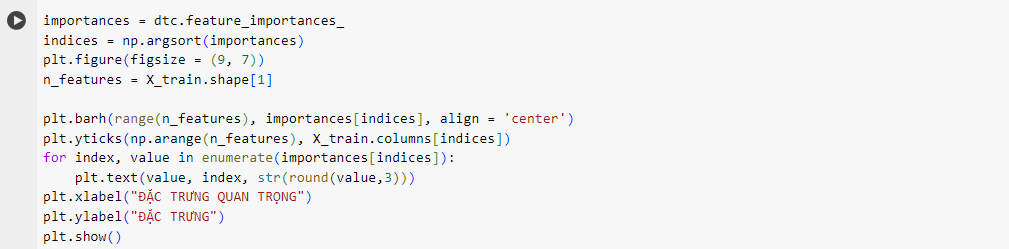
# Tạo bộ phân loại cây quyết định trên dữ liệu huấn luyện.

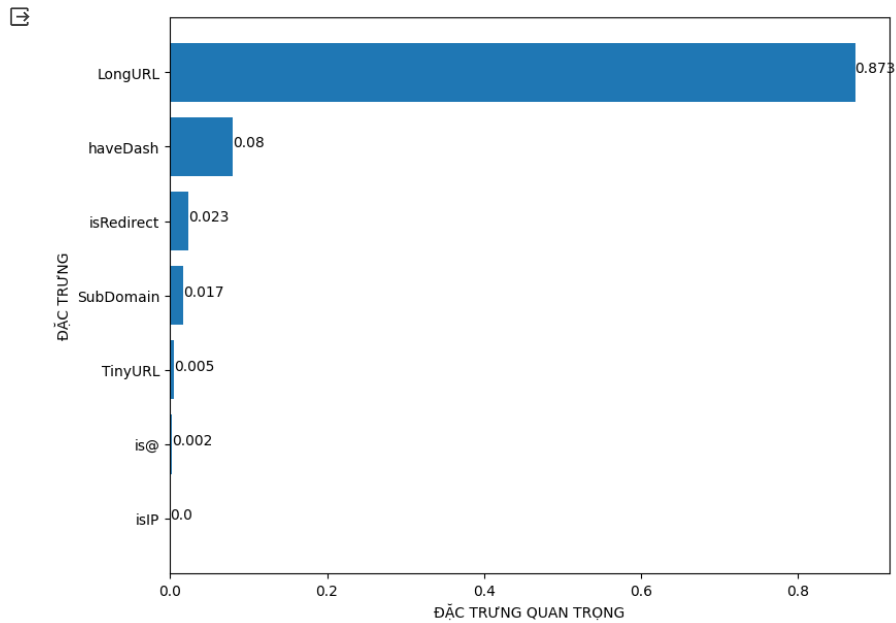
****

# Lấy tầm quan trọng của các đặc trưng từ mô hình cây quyết định đã được huấn luyện. Sắp xếp các đặc trưng theo tầm quan trọng giảm dần.

# Lấy số lượng đặc trưng trong tập huấn luyện.

# Tạo biểu đồ.

****

****

**PHÁT TRIỂN MÔ HÌNH:**

**⮚ XGBOOST**

# Huấn luyện và đánh giá mô hình XGBoost trên dữ liệu đã được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test)

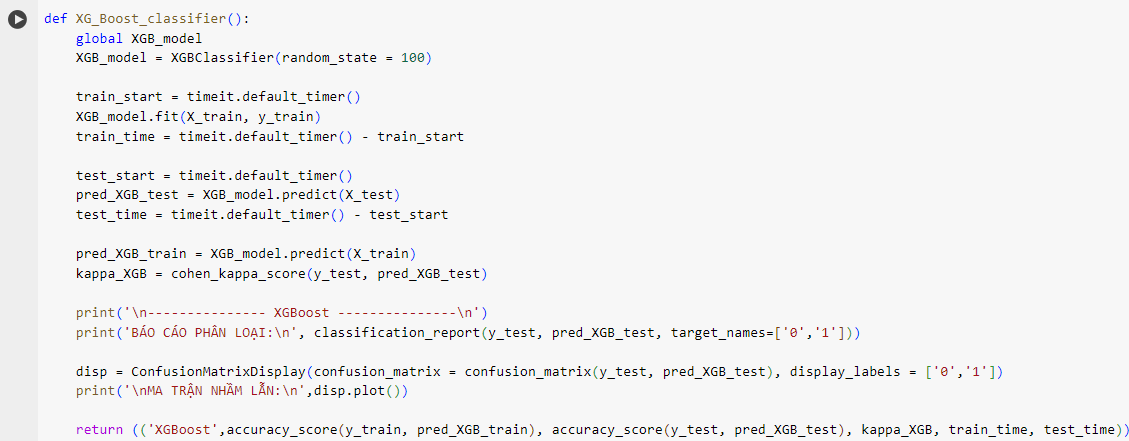
# Tạo đối tượng XGBoost.

# Huấn luyện mô hình XGBoost dựa trên tập huấn luyện và nhãn tương ứng.

# Sử dụng mô hình đã huấn luyện để đưa ra dự đoán trên tập kiểm tra.

# ĐƯA RA DƯ ĐOÁN TRÊN TẬP HUẤN LUYỆN. TÍNH HỆ SỐ KAPPA.

# Tạo đối tượng biểu diễn ma trận nhầm lẫn.

****

**⮚ CÂY QUYẾT ĐỊNH**

# Huấn luyện và đánh giá mô hình CÂY QUYẾT ĐỊNH trên dữ liệu đã được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test)

# Tạo đối tượng CÂY QUYẾT ĐỊNH.

# Huấn luyện mô hình CÂY QUYẾT ĐỊNH dựa trên tập huấn luyện & nhãn tương ứng.

# Sử dụng mô hình đã huấn luyện để đưa ra dự đoán trên tập kiểm tra.

# ĐƯA RA DƯ ĐOÁN TRÊN TẬP HUẤN LUYỆN. TÍNH HỆ SỐ KAPPA.

# Tạo đối tượng biểu diễn ma trận nhầm lẫn.

****

**⮚ MẠNG NƠRON NHÂN TẠO**

# Huấn luyện và đánh giá mô hình MẠNG NƠRON NHÂN TẠO trên dữ liệu đã được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test)

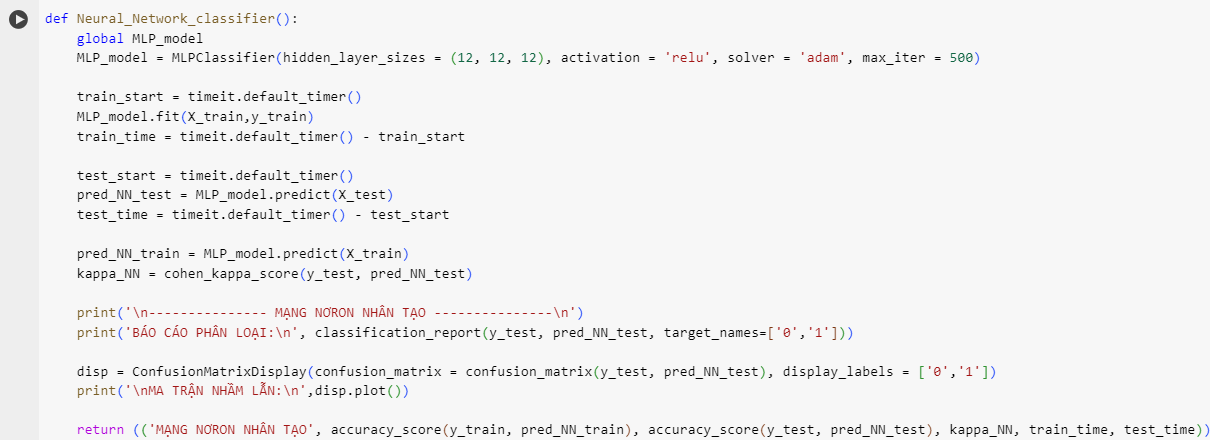
# Tạo đối tượng MẠNG NƠRON NHÂN TẠO (3 lớp ẩn, mỗi lớp chứa 12 nơron)

# Huấn luyện mô hình MẠNG NƠRON dựa trên tập huấn luyện & nhãn tương ứng.

# Sử dụng mô hình đã huấn luyện để đưa ra dự đoán trên tập kiểm tra.

# ĐƯA RA DƯ ĐOÁN TRÊN TẬP HUẤN LUYỆN. TÍNH HỆ SỐ KAPPA.

# Tạo đối tượng biểu diễn ma trận nhầm lẫn.

****

**⮚ SUPPORT VECTOR MACHINE**

# Huấn luyện và đánh giá mô hình SVM trên dữ liệu đã được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test)

# Tạo một đối tượng SVC (poly: kernel đa thức, bậc 9)

# Huấn luyện mô hình SVM dựa trên tập huấn luyện & nhãn tương ứng.

# Sử dụng mô hình đã huấn luyện để đưa ra dự đoán trên tập kiểm tra.

# ĐƯA RA DƯ ĐOÁN TRÊN TẬP HUẤN LUYỆN. TÍNH HỆ SỐ KAPPA.

# Tạo đối tượng biểu diễn ma trận nhầm lẫn.



**⮚ NAIVE BAYES**

# Huấn luyện và đánh giá mô hình NAIVE BAYES trên dữ liệu đã được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test)

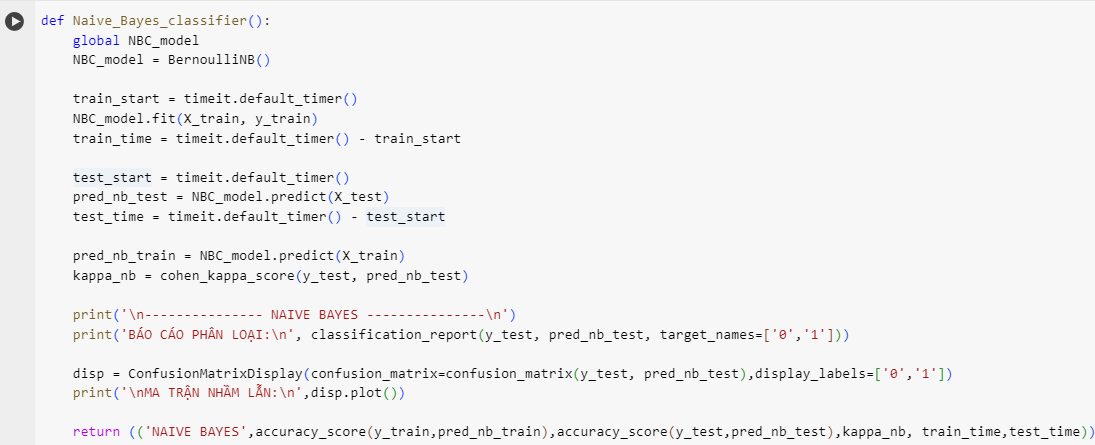
# Tạo một đối tượng BernoulliNB (Naive Bayes dạng Bernoulli)

# Huấn luyện mô hình NAIVE BAYES dựa trên tập huấn luyện & nhãn tương ứng.

# Sử dụng mô hình đã huấn luyện để đưa ra dự đoán trên tập kiểm tra.

# ĐƯA RA DƯ ĐOÁN TRÊN TẬP HUẤN LUYỆN. TÍNH HỆ SỐ KAPPA.

# Tạo đối tượng biểu diễn ma trận nhầm lẫn.



**⮚ HỒI QUY LOGISTIC**

# Huấn luyện và đánh giá mô hình HỒI QUY LOGISTIC trên dữ liệu đã được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test)

# Tạo một đối tượng LogisticRegression.

# Huấn luyện mô hình HỒI QUY LOGISTIC dựa trên tập huấn luyện & nhãn tương ứng.

# Sử dụng mô hình đã huấn luyện để đưa ra dự đoán trên tập kiểm tra.

# ĐƯA RA DƯ ĐOÁN TRÊN TẬP HUẤN LUYỆN. TÍNH HỆ SỐ KAPPA.

# Tạo đối tượng biểu diễn ma trận nhầm lẫn.

****

**⮚ RANDOM FOREST**

# Huấn luyện và đánh giá mô hình RANDOM FOREST trên dữ liệu đã được chia thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test)

# Tạo một đối tượng RandomForestClassifier.

# Huấn luyện mô hình RANDOM FOREST dựa trên tập huấn luyện & nhãn tương ứng.

# Sử dụng mô hình đã huấn luyện để đưa ra dự đoán trên tập kiểm tra.

# ĐƯA RA DƯ ĐOÁN TRÊN TẬP HUẤN LUYỆN. TÍNH HỆ SỐ KAPPA.

# Tạo đối tượng biểu diễn ma trận nhầm lẫn.

****

**ĐÁNH GIÁ SỐ LIỆU:**

# Vẽ đồ thị ROC (Receiver Operating Characteristic) cho từng mô hình.

# Tạo một figure mới để bắt đầu vẽ đồ thị ROC.

# Tạo danh sách models chứa thông tin các mô hình cần vẽ ROC curve.

# VÒNG LẶP FOR:

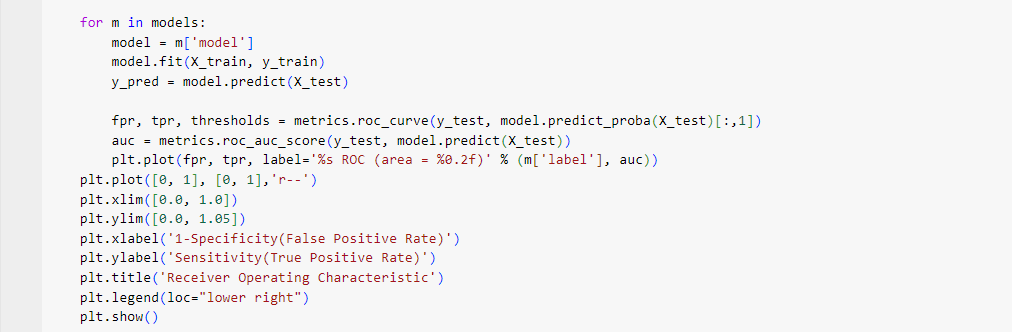
# Duyệt qua từng mô hình, huấn luyện mô hình trên tập huấn luyện, dự đoán nhãn trên tập kiểm tra.

# Tính tỷ lệ dương tính giả (FPR) và tỷ lệ đúng dương tính (TPR) để tạo đường ROC.

# Tính diện tích dưới đường ROC (AUC), đại diện cho hiệu suất tổng thể của mô hình.

# Vẽ đường ROC cho mô hình hiện tại và thêm nhãn bao gồm tên mô hình và giá trị AUC.

****

****

# Gọi và tổng hợp kết quả của các mô hình học máy:

# Khởi tạo các danh sách để lưu trữ các giá trị đánh giá và thời gian của từng mô hình.

# Lưu trữ hệ số Kappa của các mô hình.

# Lưu trữ tên các mô hình.

# Lưu trữ độ chính xác trên tập huấn luyện.

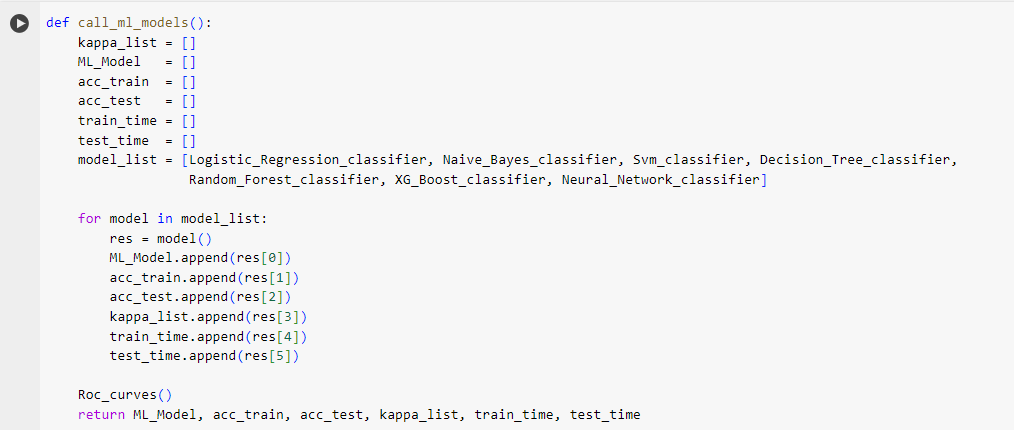
# Lưu trữ độ chính xác trên tập kiểm tra.

# Lưu trữ thời gian huấn luyện của các mô hình.

# Lưu trữ thời gian dự đoán của các mô hình.

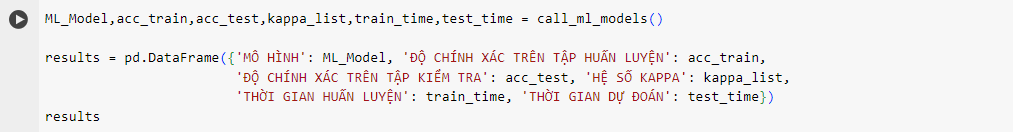
# Lặp qua các mô hình và thu thập kết quả.

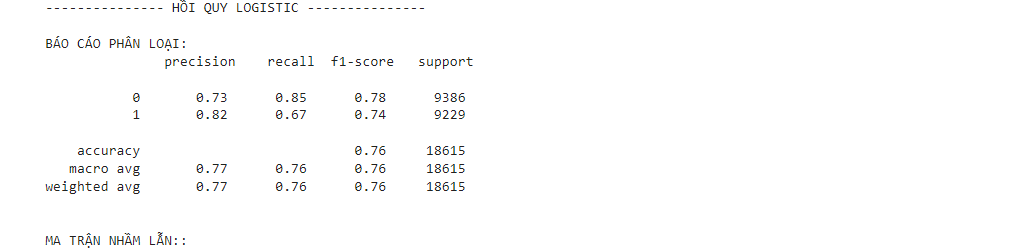
# Hiển thị đường cong ROC của tất cả các mô hình.

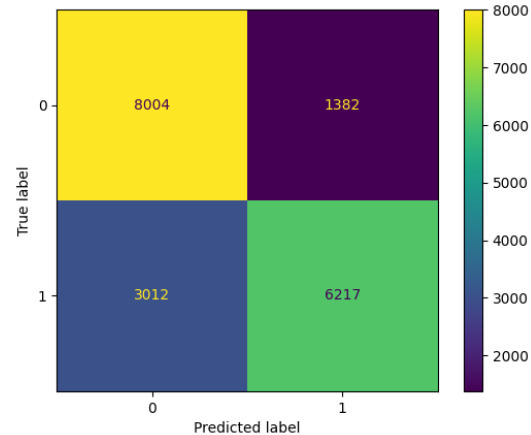
****

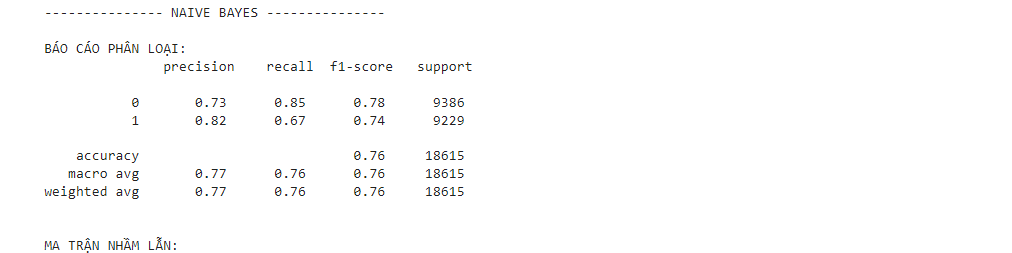
# Gọi hàm để lấy thông tin về hiệu suất của tất cả các mô hình.

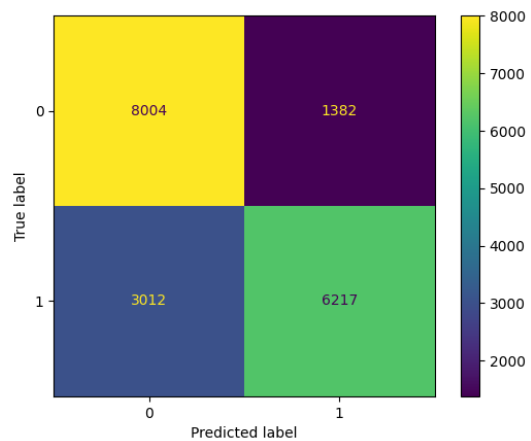
# Tạo một DataFrame với các cột ứng với thông tin hiệu suất và thời gian của mỗi mô hình.

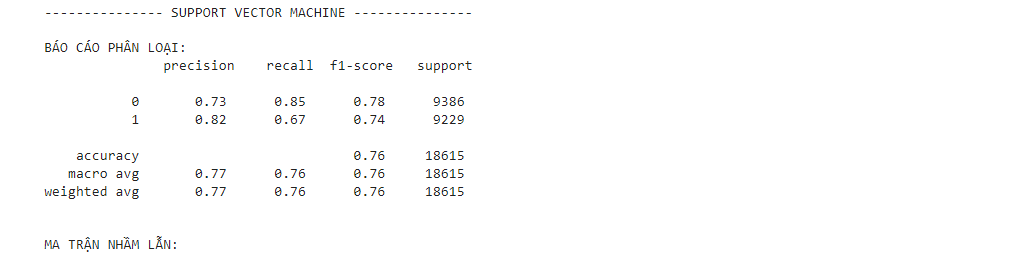
****

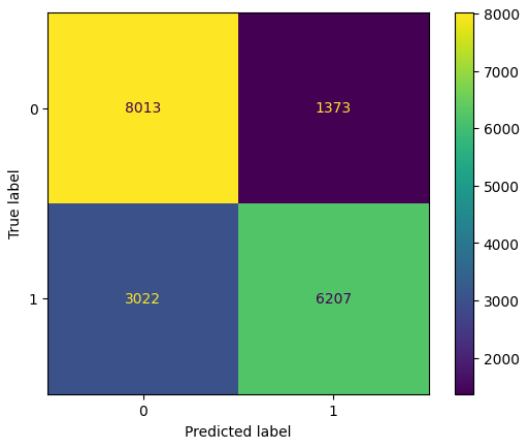
****

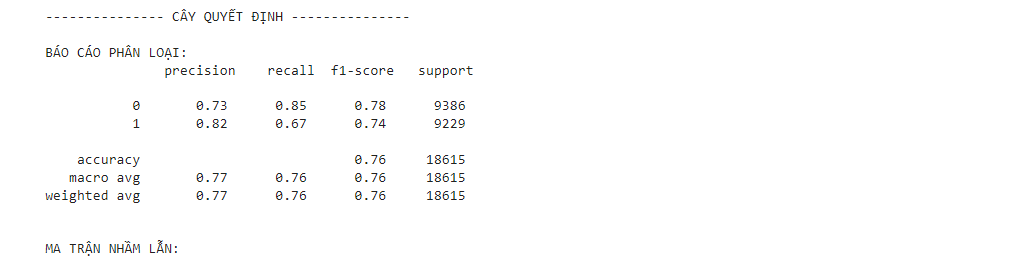
****

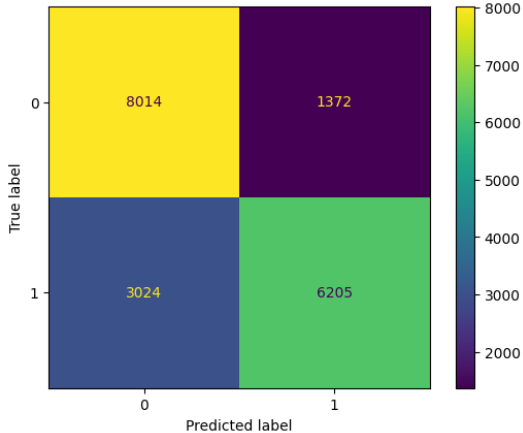
****

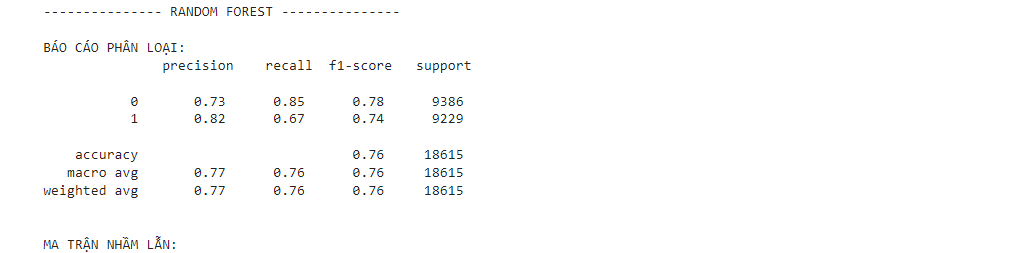
****

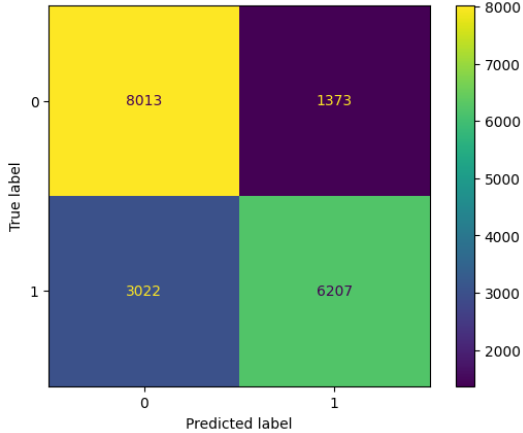
****

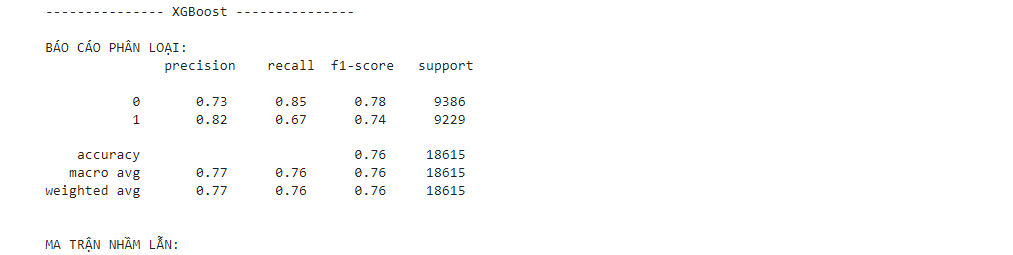
****

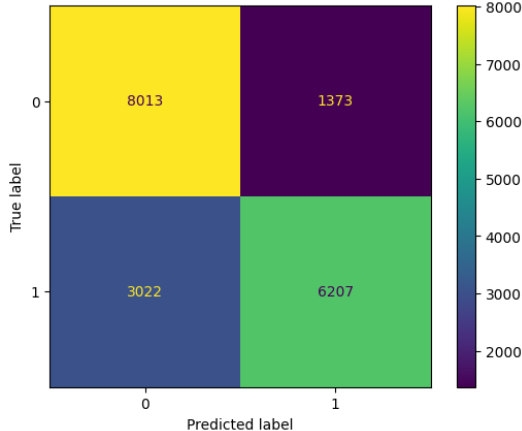
****

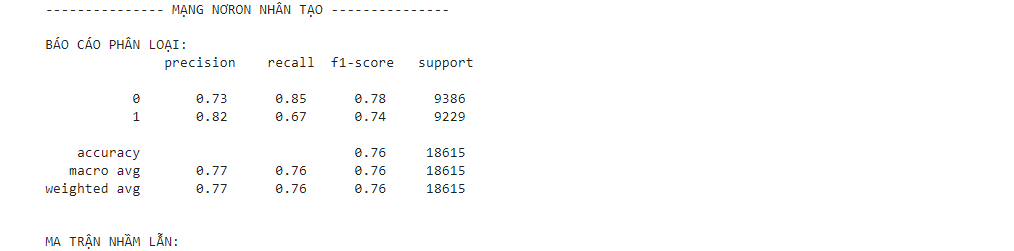
****

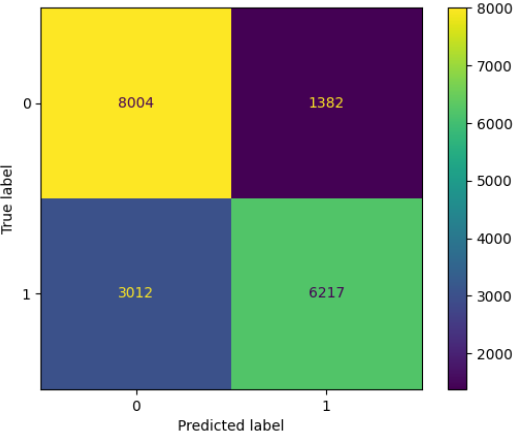
****

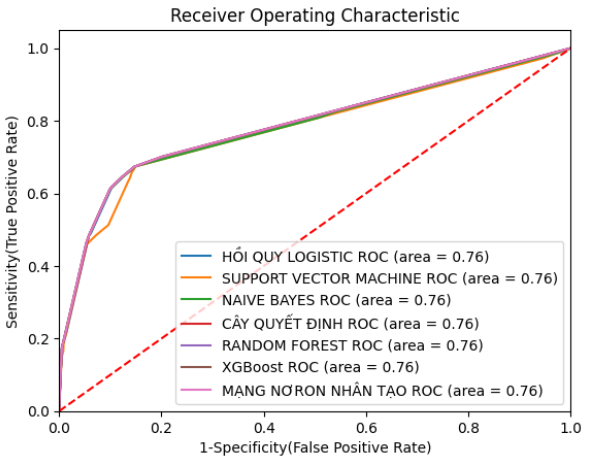
****

****

****

****

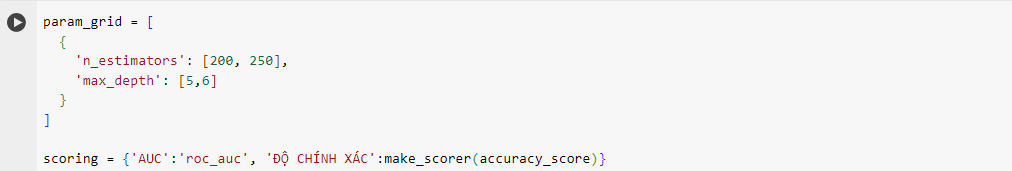


****

****

**HYPERPARAMETER TURNING (ĐIỀU CHỈNH SIÊU THAM SỐ):**

# Thiết lập không gian tham số và các chỉ số đánh giá cho việc điều chỉnh siêu tham số của mô hình XGBoost

****

# Tìm kiếm siêu tham số tối ưu cho mô hình XGBoost

# Khởi tạo đối tượng GridSearchCV để thực hiện tìm kiếm lưới tham số.

# Huấn luyện mô hình với Grid Search.

****

****

# Huấn luyện và đánh giá mô hình XGBoost với các siêu tham số tối ưu.

# Lấy mô hình XGBoost tốt nhất từ quá trình tìm kiếm siêu tham số.

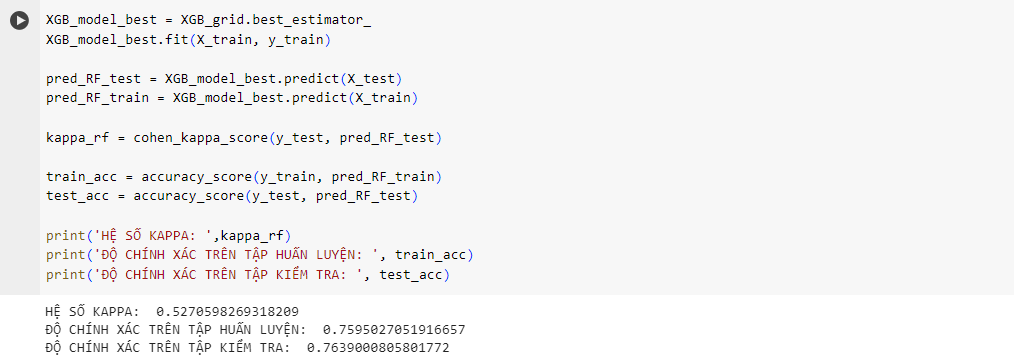
# Huấn luyện lại mô hình với tham số tối ưu trên tập huấn luyện để đảm bảo hiệu suất tốt nhất.

# Đưa ra dự đoán trên tập kiểm tra. Đưa ra dự đoán trên tập huấn luyện.

# Tính hệ số KAPPA giữa dự đoán & nhãn thực tế trên tập dữ liệu kiểm tra.

# Tính độ chính xác của mô hình trên toàn bộ tập dữ liệu huấn luyện.

# Tính độ chính xác của mô hình trên tập dữ liệu kiểm tra.

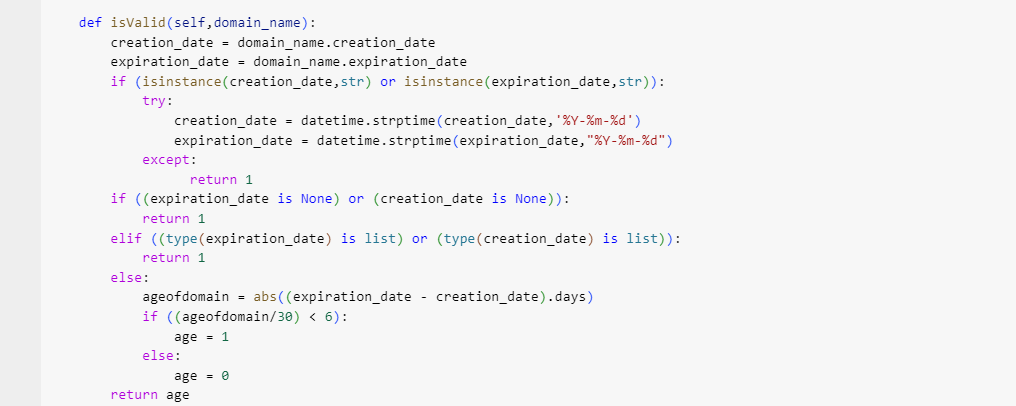
****

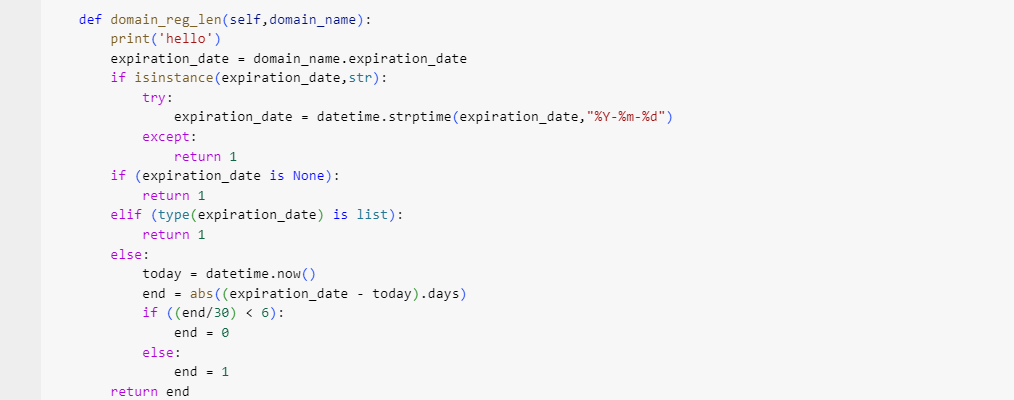
**LƯU LẠI MÔ HÌNH XGBOOST ĐÃ HUẤN LUYỆN:**

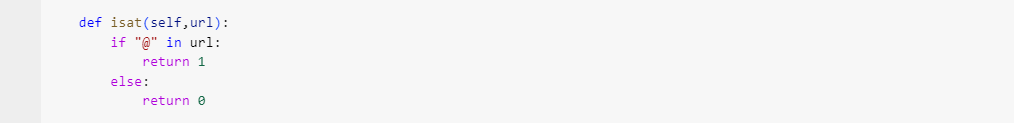
****

**TIẾN HÀNH PHÂN LOẠI BĂNG MÔ HÌNH ĐÃ LƯU:**

****

****

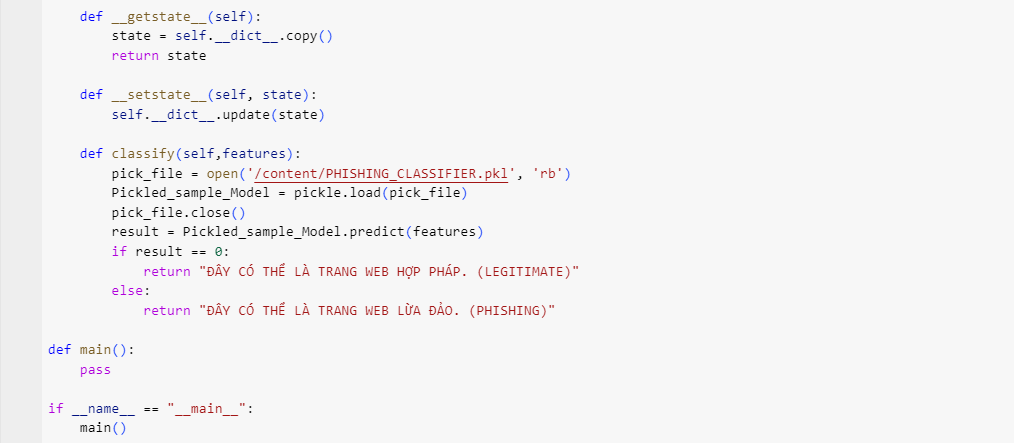
****

****

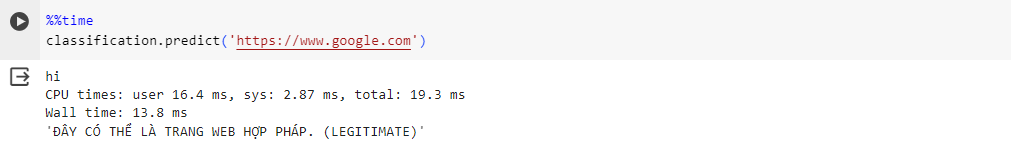
****

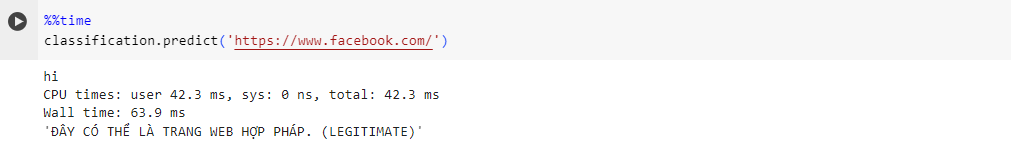
****

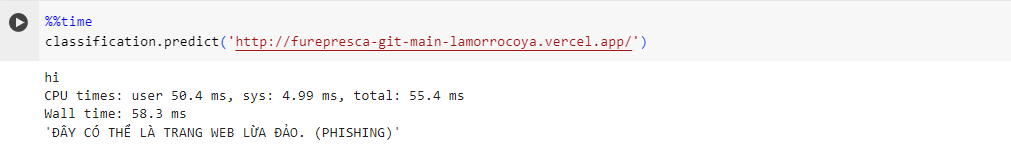
****

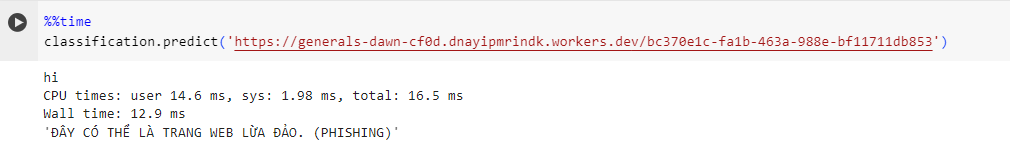
****

****

****

****

****

****

## 2. XÂY DỰNG EXTENSION:

### 2.1. PYTHON:

Python là một ngôn ngữ lập trình mạnh mẽ, linh hoạt và dễ đọc, chuyên được sử dụng trong nhiều lĩnh vực khác nhau. Được tạo ra bởi Guido van Rossum và ra mắt lần đầu tiên vào năm 1991, Python nhanh chóng trở thành một trong những ngôn ngữ phổ biến nhất trên thế giới do tính đơn giản, dễ học, và cộng đồng hỗ trợ mạnh mẽ.



Một trong những ưu điểm lớn của Python là khả năng tích hợp mạnh mẽ với nhiều thư viện và framework, đặc biệt là trong lĩnh vực Machine Learning. Thư viện nổi tiếng như NumPy, Pandas, Matplotlib, và scikit-learn đã giúp Python trở thành một ngôn ngữ chính trong nghiên cứu và phát triển các mô hình máy học.

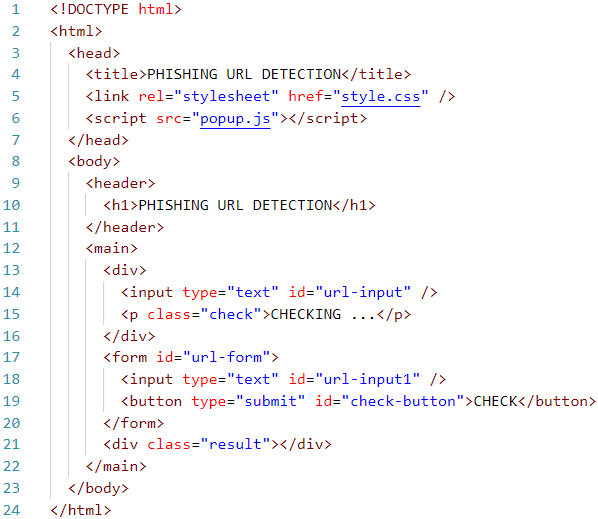
Đối với Machine Learning, Python cung cấp môi trường phát triển linh hoạt và hiệu quả. Dưới đây là một số ưu điểm quan trọng của Python trong lĩnh vực này:

* Python có một cộng đồng lớn và đa dạng, giúp người lập trình và nhà nghiên cứu có thể chia sẻ thông tin, học hỏi và giải quyết vấn đề một cách nhanh chóng.
* Các thư viện như TensorFlow, PyTorch và Keras cung cấp các công cụ mạnh mẽ để xây dựng, huấn luyện và triển khai mô hình máy học.
* Python có cú pháp đơn giản và gần gũi với ngôn ngữ tự nhiên, giúp người lập trình dễ dàng hiểu và sửa lỗi code.
* Python cho phép tích hợp nhanh chóng giữa các thư viện và frameworks khác nhau, giúp phát triển mô hình Machine Learning một cách linh hoạt và hiệu quả.
* Sự phổ biến của Python trong cả nghiên cứu và công nghiệp làm cho nó trở thành một lựa chọn hàng đầu cho các dự án Machine Learning.

Nhờ những đặc điểm nổi bật này, Python đã đóng góp quan trọng vào sự phát triển và tiến bộ của lĩnh vực Machine Learning, giúp nhiều người lập trình và nhà nghiên cứu đạt được những thành công đáng kể trong việc xử lý dữ liệu và xây dựng các mô hình dự đoán hiệu quả.

### 2.2. XÂY DỰNG:

#### 2.2.1. FILE popup.html:



**<!DOCTYPE html>**: Khai báo loại tài liệu HTML.

**<html>**: Thẻ mở của tài liệu HTML.

**<head>**: Phần đầu, chứa các thông tin meta và liên kết đến các tệp CSS và JavaScript.

**<title>PHISHING URL DETECTION</title>**: Đặt tiêu đề.

<**link .../>**: Liên kết tệp CSS để tùy chỉnh giao diện trang.

**<script src="popup.js"></script>**: Liên kết đến tệp "popup.js" để thực hiện các chức năng tương tác trên trang.

**</head>**

**<body>:** Phần thân của tài liệu HTML.

**<header>…</header>**: Phần tiêu đề của trang.

**<main>:** Phần chính của extension.

**<div>**

**<input type="text" id="url-input" />**: Ô chứa URL từ trình duyệt

**<p class="check">CHECKING ...</p>**: Hiện trong quá trình kiểm tra.

**</div>**

**<form>**

**<input type="text" id="url-input1" />**: Ô nhập dữ liệu URL.

**<button type="submit" id="check-button">CHECK</button>**: Nút submit.

**</form>**

**<div class="result"></div>:** Hiển thị kết quả của quá trình kiểm tra URL.

**</main>**

**</body>**

**</html>**

#### 2.2.2. FILE popup.js:



Sử dụng API của Chrome để truy vấn thông tin về tab đang hoạt động.

Lấy URL của tab đang được chọn và đặt giá trị vào thẻ input có id là “url-input”

Gọi hàm checkPhishing với URL và một chuỗi “check” làm đối số.

**chrome.tabs.query({ active: true, currentWindow: true }, function (tabs) {**

**var url = tabs[0].url;**

**document.getElementById("url-input").value = url;**

**checkPhishing(url, "check");**

**});**

Khi người dùng submit form

Lấy giá trị của URL từ thẻ input có id là “url-input1”

**document**

**.getElementById("url-form")**

**.addEventListener("submit", function (event) {**

**event.preventDefault();**

**var url = document.getElementById("url-input1").value;**

**checkPhishing(url, "result");**

**});**

Hàm kiểm tra URL là PHISHING hay LEGITIMATE.

Gửi yêu cầu GET đến một API máy học với URL như là một tham số truy vấn.

Xử lý phản hồi từ API.

**function checkPhishing(url, classname) {**

**fetch(`http://127.0.0.1:8000/api?url=${url}`, {**

**method: "GET",**

**headers: {**

**"Content-Type": "application/json",**

**},**

**})**

**.then((response) => response.json())**

**.then((data) => {**

**console.log(data);**

**let test = data[0];**

**console.log(test);**

**console.log(typeof test);**

**var resultDiv = document.getElementsByClassName(classname)[0];**

**if (typeof test === "string") {**

**resultDiv.innerHTML = test;**

**} else {**

**resultDiv.innerHTML = data.msg;**

**}**

**})**

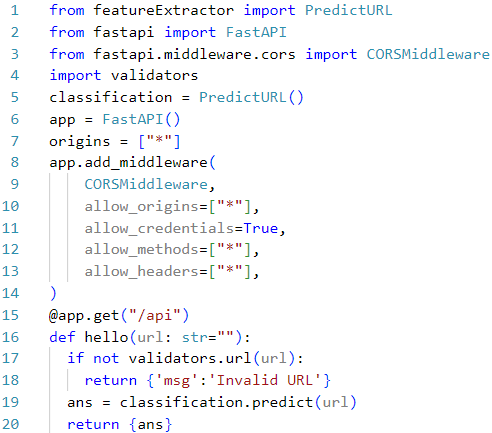
**.catch((error) => {**

**console.error("Error:", error);**

**});**

**}**

#### 2.2.3. FILE api.py:



**⮚ IMPORT THƯ VIỆN & MODULE:**

**PredictURL:** lớp từ tệp featureExtractor.py dùng để dự đoán URL là PHISHING hay LEGITIMATE gồm các chức năng trích xuất đặc trưng và dự đoán trên mô hình học máy.

**FastAPI:** framework FastAPI để tạo API web.

**CORSMiddleware:** middleware CORS cho phép truy cập API từ domain khác nhau.

**validators:** kiểm tra tính hợp lệ của URL.

**⮚ KHỞI TẠO ĐỐI TƯỢNG DỰ ĐOÁN VÀ ỨNG DỤNG FASTAPI:**

**classification = PredictURL()** : Tạo một đối tượng của lớp PredictURL.

**app = FastAPI()** : Tạo một ứng dụng FastAPI.

**⮚ CẤU HÌNH CORS MIDDLEWARE:**

**origins = ["\*"]:** cho phép truy cập từ tất cả các domain.

**app.add\_middleware(**

**CORSMiddleware,**

**allow\_origins=["\*"], :** cho phép tất cả các nguồn được chỉ định trong origins.

**allow\_credentials = True, :** cho phép trình duyệt gửi cookie và các thông tin xác thực khác từ domain khác đến API.

**allow\_methods=["\*"], :** cho phép các phương thức HTTP như GET, POST, …

**allow\_headers=["\*"], :** cho phép các header HTTP tùy chỉnh được gửi đến API.

**)**

**⮚ ĐỊNH NGHĨA API ENDPOINT:**

**@app.get("/api") :** Định nghĩa endpoint có đường dẫn “/api” cho phương thức GET.

**def hello(url: str=""):** xử lý các yêu cầu gửi đến endpoint.

**if not validators.url(url):** kiểm tra xem URL có hợp lệ hay không.

**return {'msg':'Invalid URL'} :** nếu không hợp lệ, trả về một thông báo lỗi.

**ans = classification.predict(url) :** dự đoán URL được truyền vào.

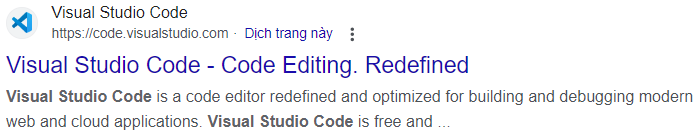
**return {ans} :** trả về kết quả dự đoán trong một đối tượng JSON.

**CHƯƠNG III. CÀI ĐẶT CHƯƠNG TRÌNH & THỰC NGHIỆM**

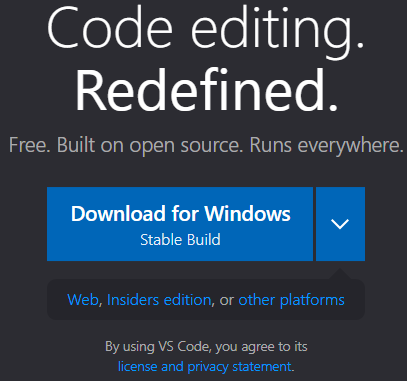
**1. CÀI ĐẶT VISUAL STUDIO CODE:**

Visual Studio Code là một trình soạn thảo mã nguồn mở và miễn phí được phát triển bởi Microsoft. Nó là một ứng dụng dùng để viết mã, chỉnh sửa và debug mã nguồn, và hỗ trợ nhiều ngôn ngữ lập trình khác nhau. Visual Studio Code là một công cụ mạnh mẽ có thể được sử dụng để phát triển các ứng dụng trên nhiều nền tảng.

**–** Truy cập trang chính thức của Visual Studio Code tại <https://code.visualstudio.com/>



**–** Tải phiên bản phù hợp với hệ điều hành (Windows, macOS, hoặc Linux).

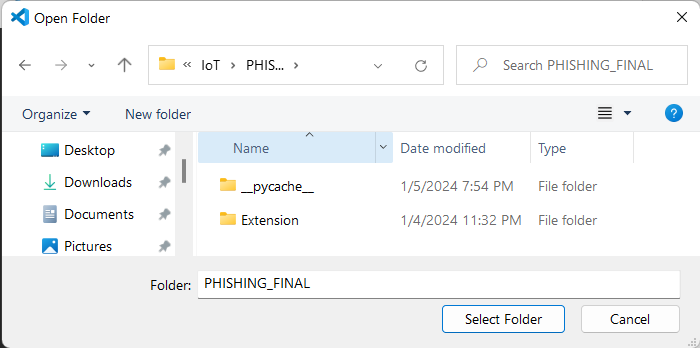


**–** Tải xuống tệp cài đặt và chạy trình cài đặt theo hướng dẫn.

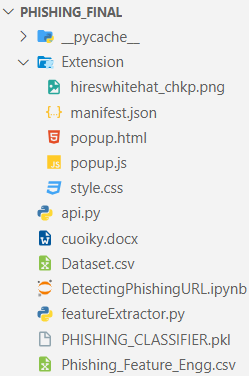


**2. CHẠY DỰ ÁN:**

**–** Trong Visual Studio Code, chọn **File → Open Folder**, sau đó chọn đến thư mục dự án và chọn **Select Folder**.

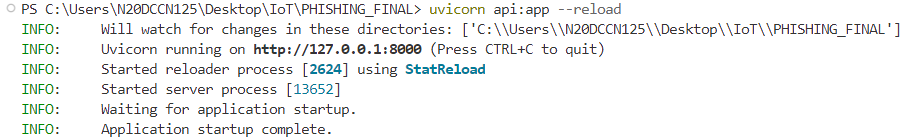


**–** Đây là cây thư mục của dự án.

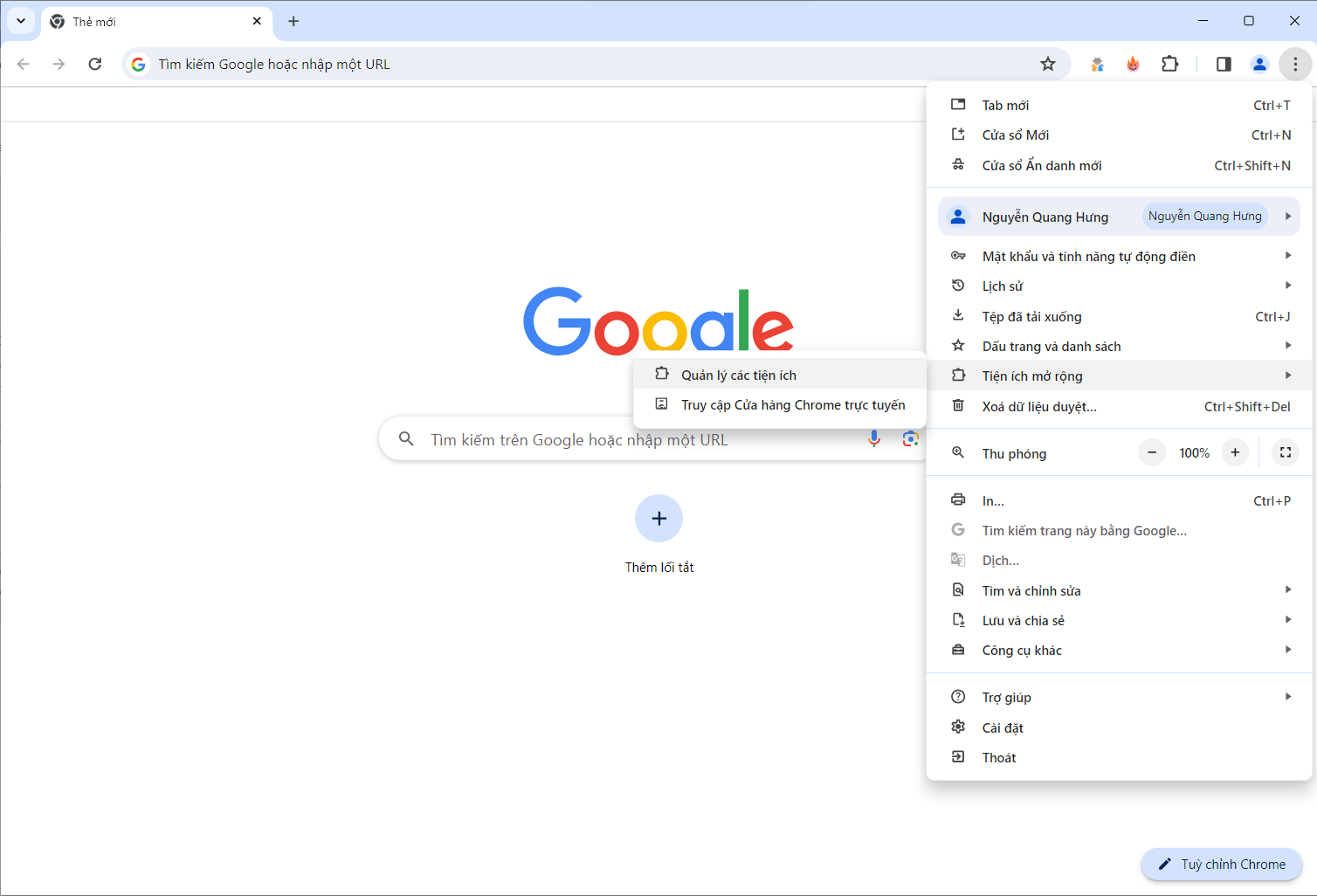


**–** Chọn **Terminal → New Terminal** để mở cửa sổ dòng lệnh tích hợp.

**–** Sau khi đã mở Terminal và đang ở trong thư mục dự án, gõ lệnh **uvicorn api:app –reload** và nhấn Enter.



– Mở trình duyệt web, chọn **→ Tiện ích mở rộng → Quản lý các tiện ích.**

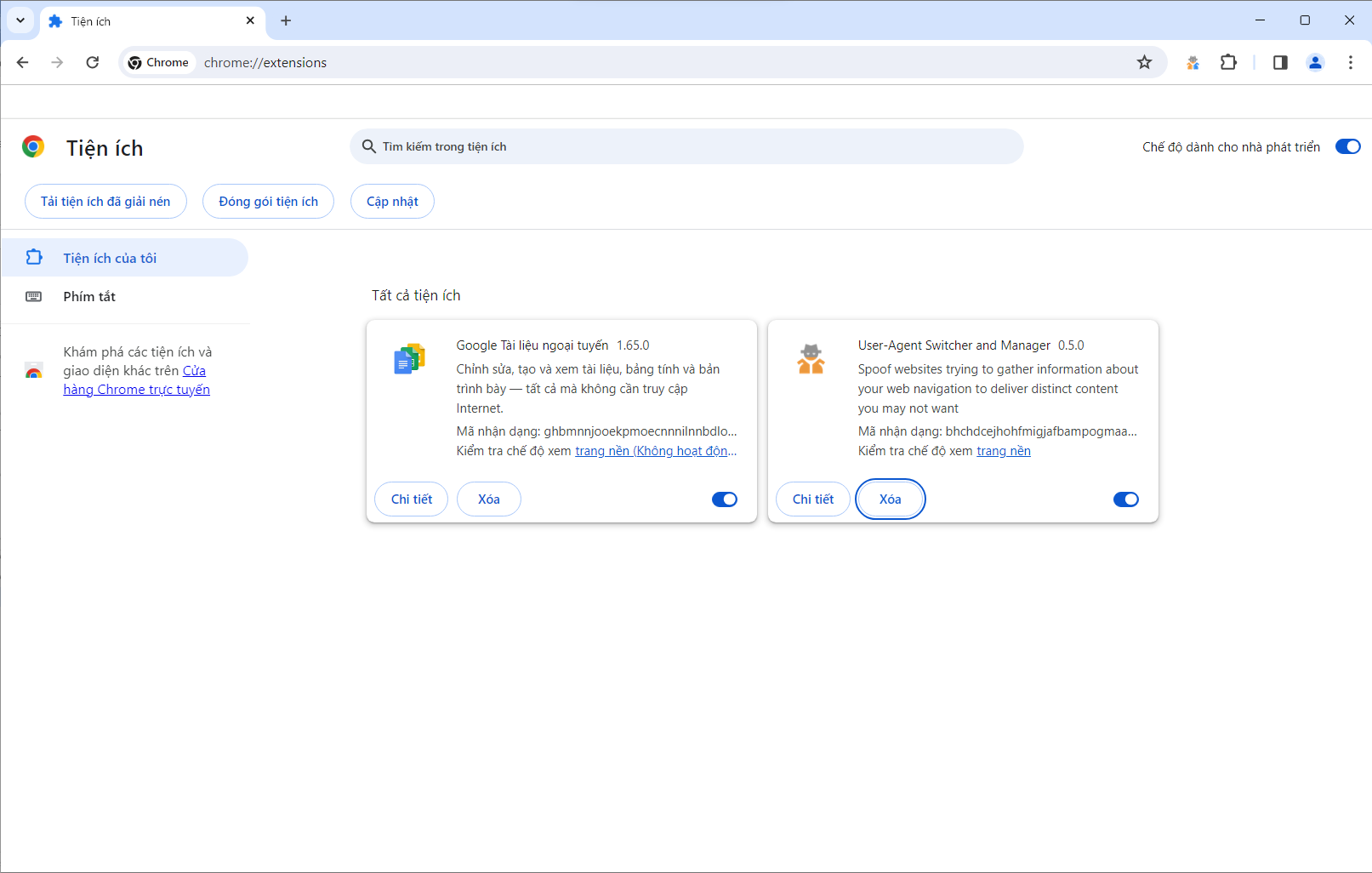


3

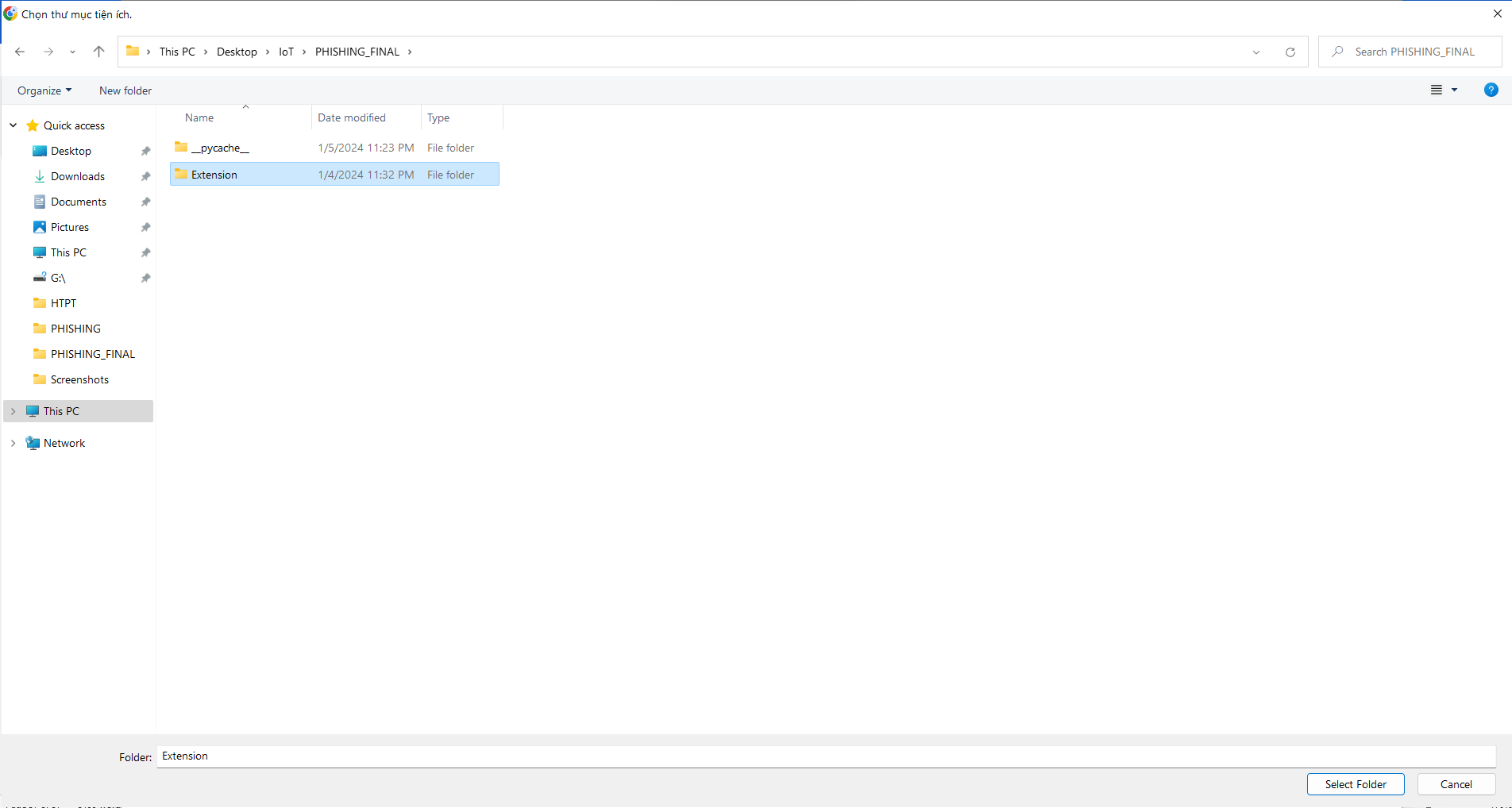
2

1

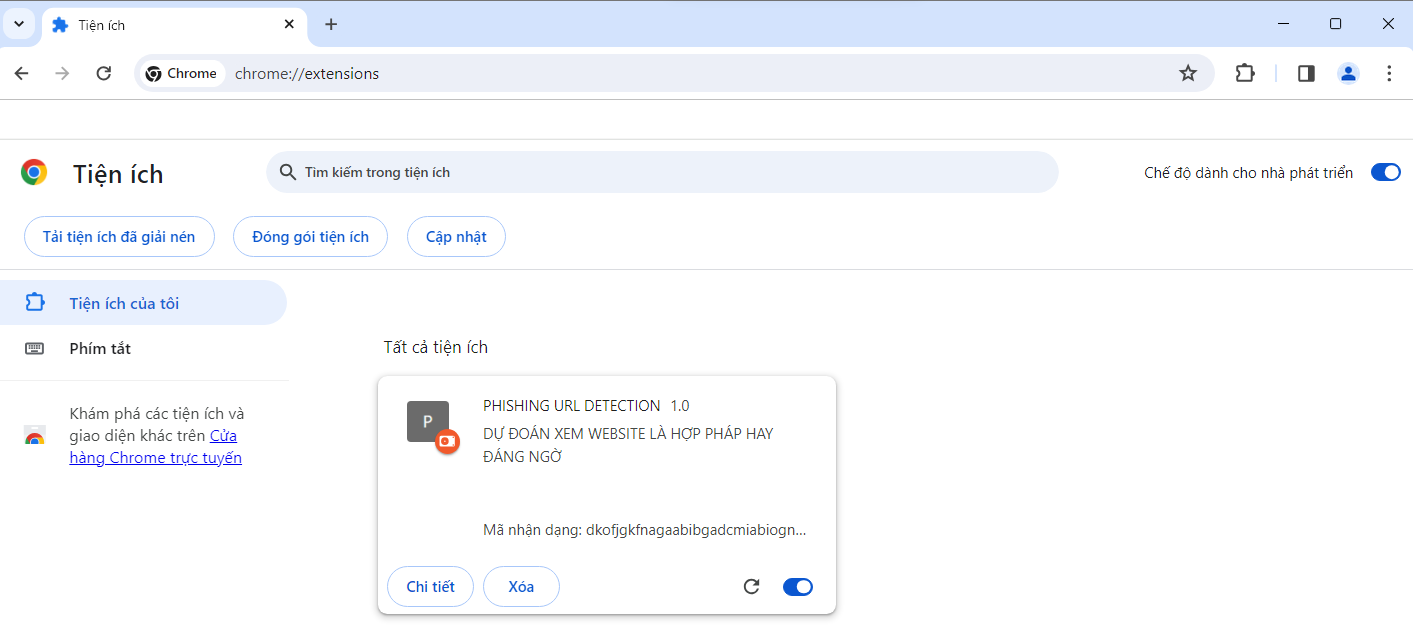
– Bật **chế độ dành cho nhà phát triển**, sau đó chọn **Tải tiện ích đã giải nén**.



– Chọn thư mục **Extension**.



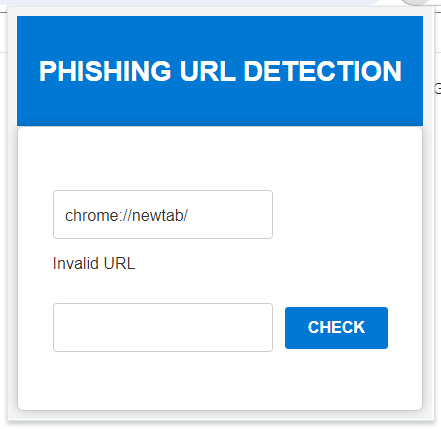
– Kiểm tra tiện ích đã được cài đặt thành công giống như hình dưới.



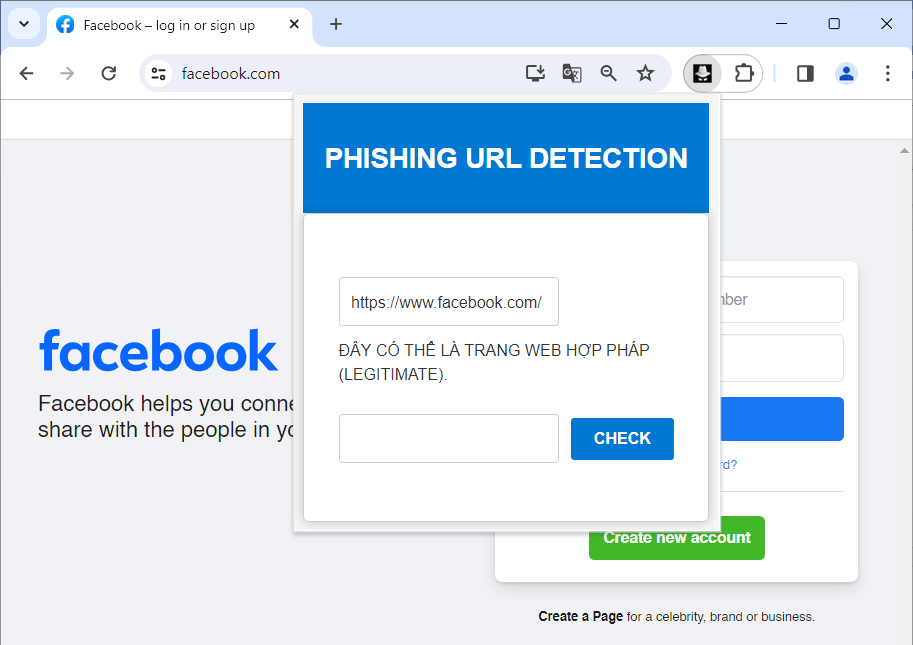
– Ghim tiện ích lên cạnh thanh URL để tiện trong quá trình sử dụng.

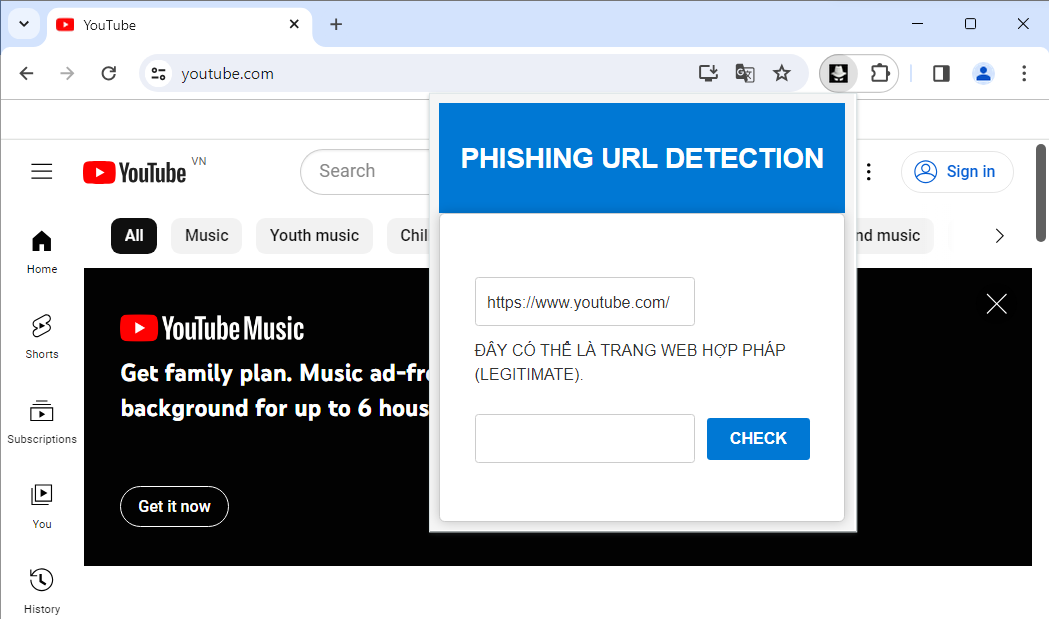


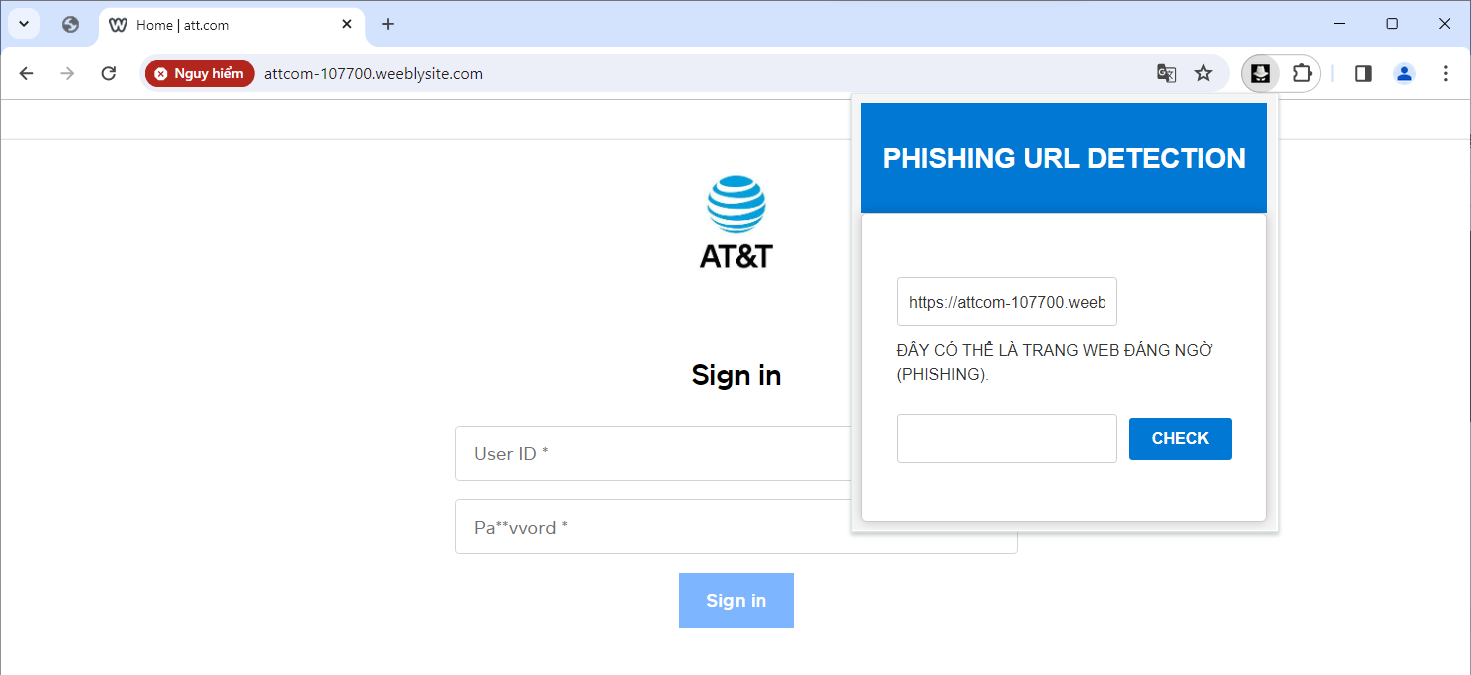
– Giao diện của tiện ích.

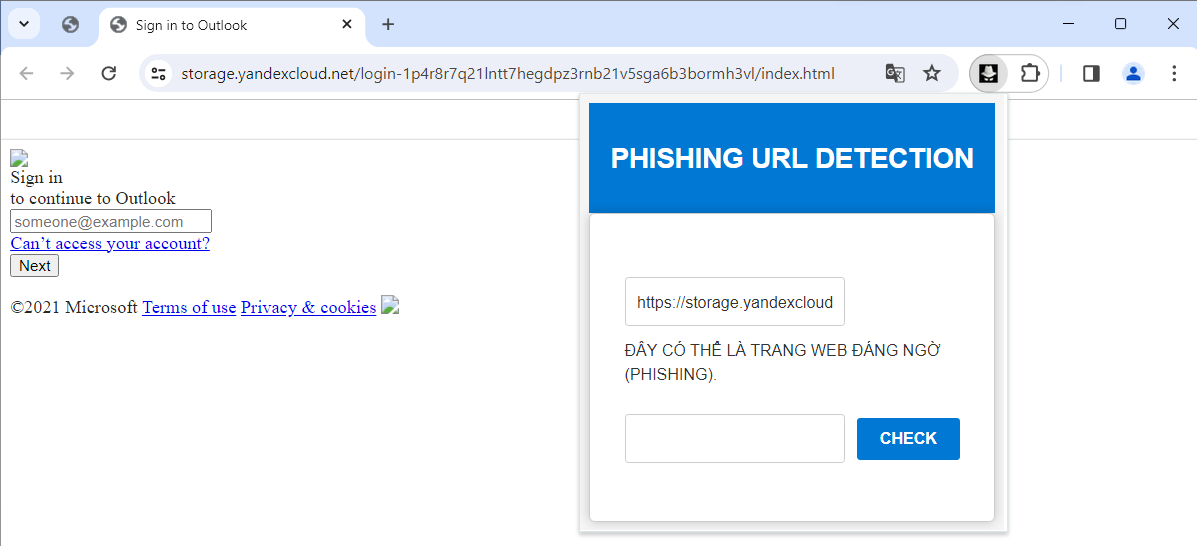


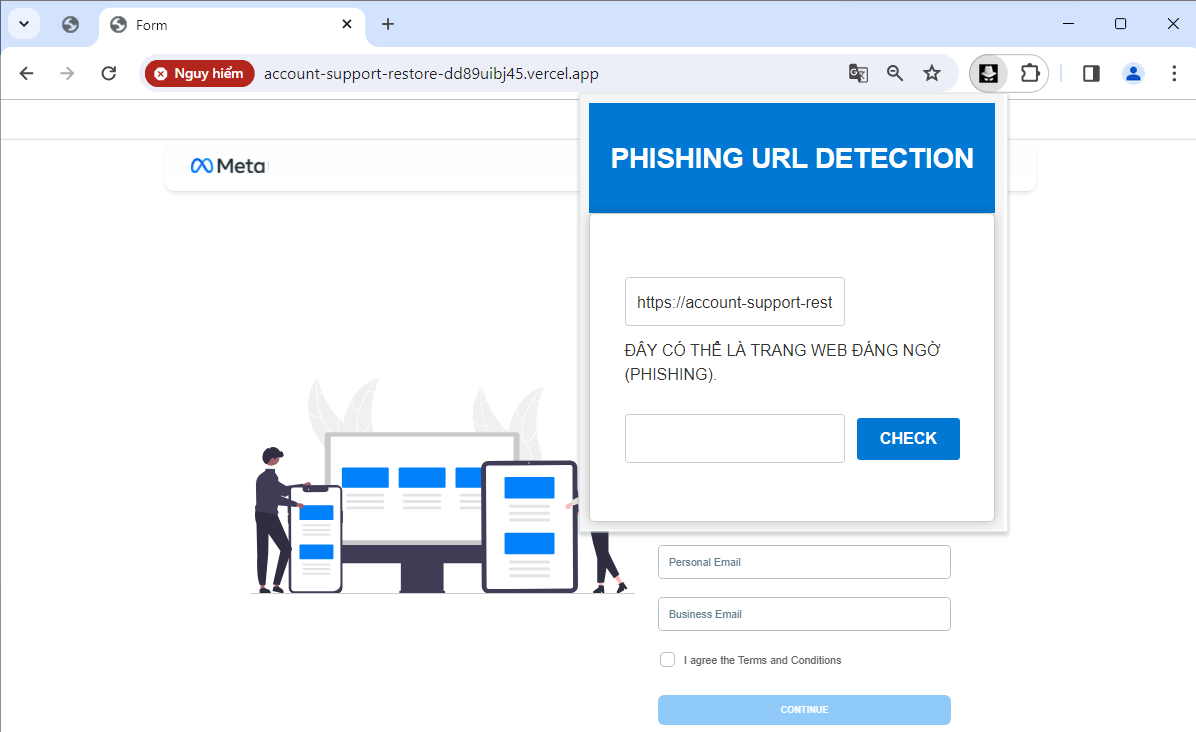
– Dự đoán trang web đang truy cập là hợp pháp (LEGITIMATE) hay đáng ngờ (PHISHING).











– Ngoài ra người dùng còn có thể tự nhập trước URL để tiện ích dự đoán.

