BÁO CÁO THỰC HÀNH LAB 2

**Môn học: Phương pháp học máy trong an toàn thông tin**

**Tên chủ đề: Machine Learning based Malware Detection**

*GVHD: Nguyễn Hữu Quyền*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Bạch Văn Xuân Thông | 20521978 | 20521978@gm.uit.edu.vn |
| 2 | Nguyễn Minh Trí | 20522049 | 20522049@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-0)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Yêu cầu 1 | 100% |
| 2 | Yêu cầu 2 | 100% |
| 3 | Yêu cầu 3 | 100% |
| 4 | Yêu cầu 4 | 80% |
| 5 | Yêu cầu 5 | 100% |
| 6 | Yêu cầu 6 | 100% |
| 7 | Yêu cầu 7 | 100% |
| 8 | Yêu cầu 8 | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

**1. Phân tích tĩnh mã độc**

**a) Tính toán hàm băm của một mẫu**

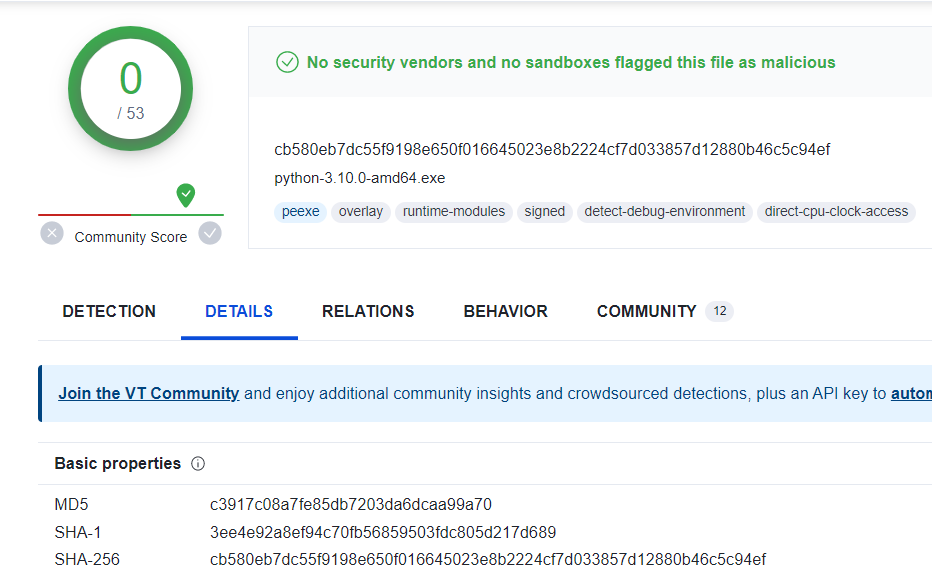


*Yêu cầu 1. Sinh viên so sánh kết quả băm với VirusTotal và website Python.*

**MD5:** c3917c08a7fe85db7203da6dcaa99a70

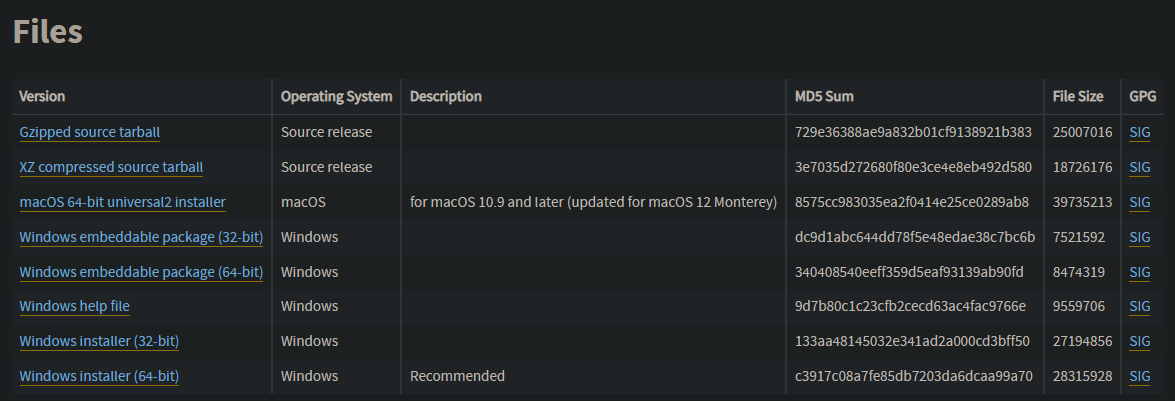
**SHA256:** cb580eb7dc55f9198e650f016645023e8b2224cf7d033857d12880b46c5c94ef

- Đối chiếu với VirusTotal:



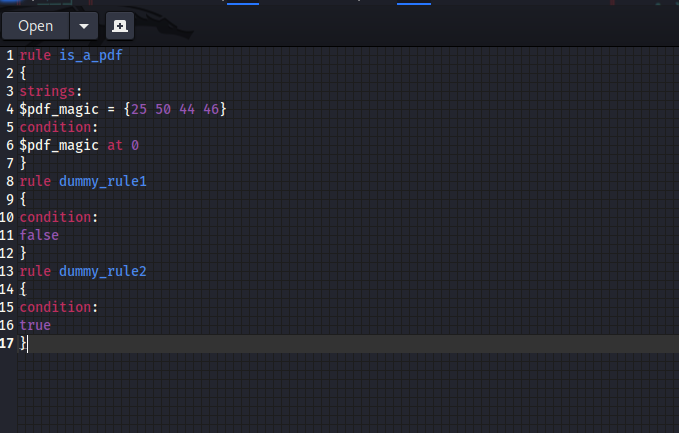
=> SHA256 hash và MD5 hash trùng nhau => file tải về toàn vẹn

- Đối chiếu với website Python:

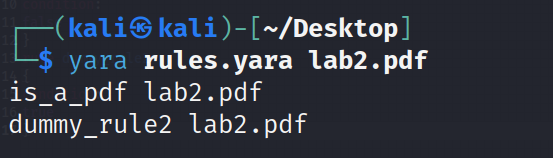
=> HASH MD5 trùng nhau => file tải về toàn vẹn

**b) YARA**

Tạo rules yara kiểm tra có phải là file pdf không

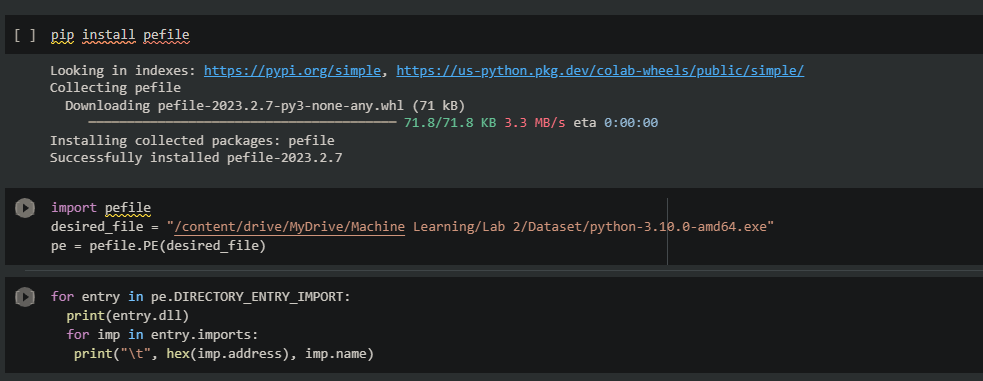


Kết quả kiểm tra như sau

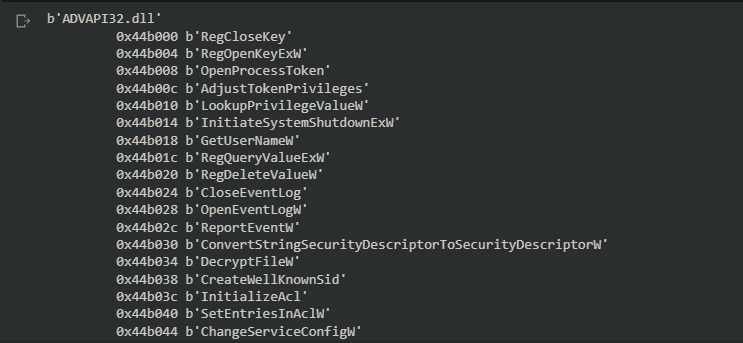


**c) Kiểm tra PE header**

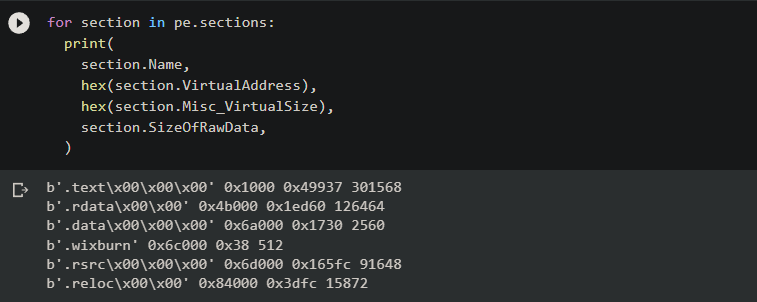
Liệt kê các import của tập tin PE



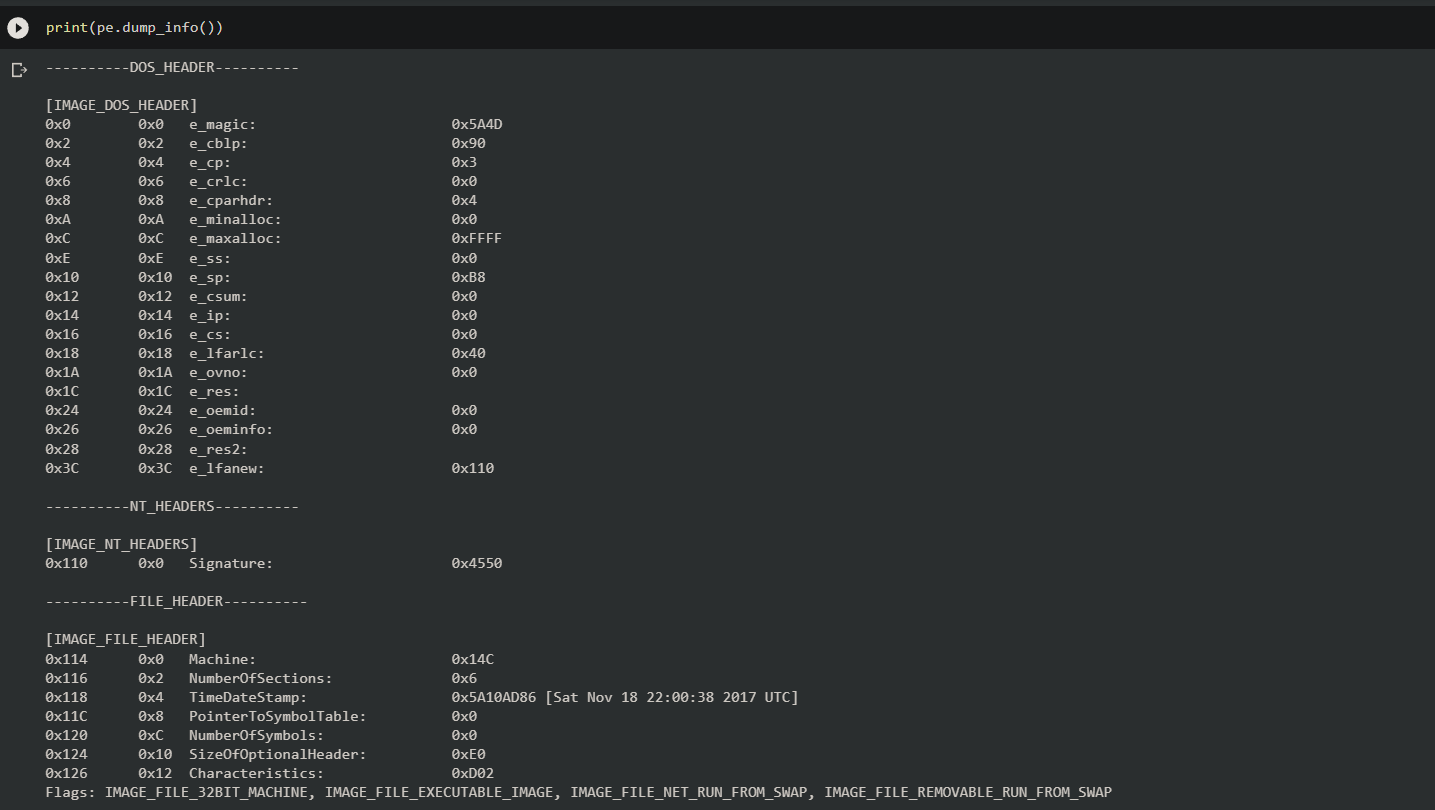
Một phần kết quả



Liệt kê các section của tập tin PE

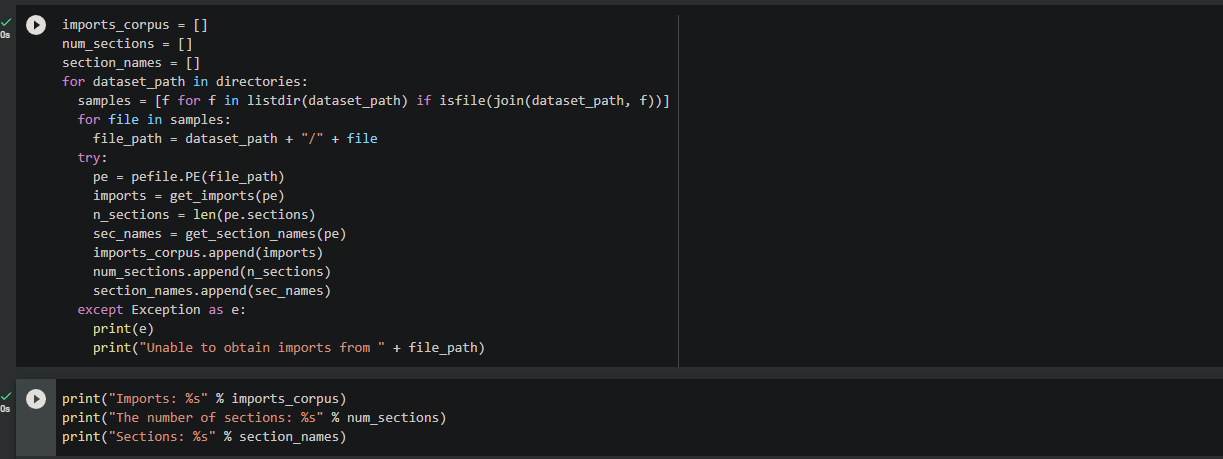
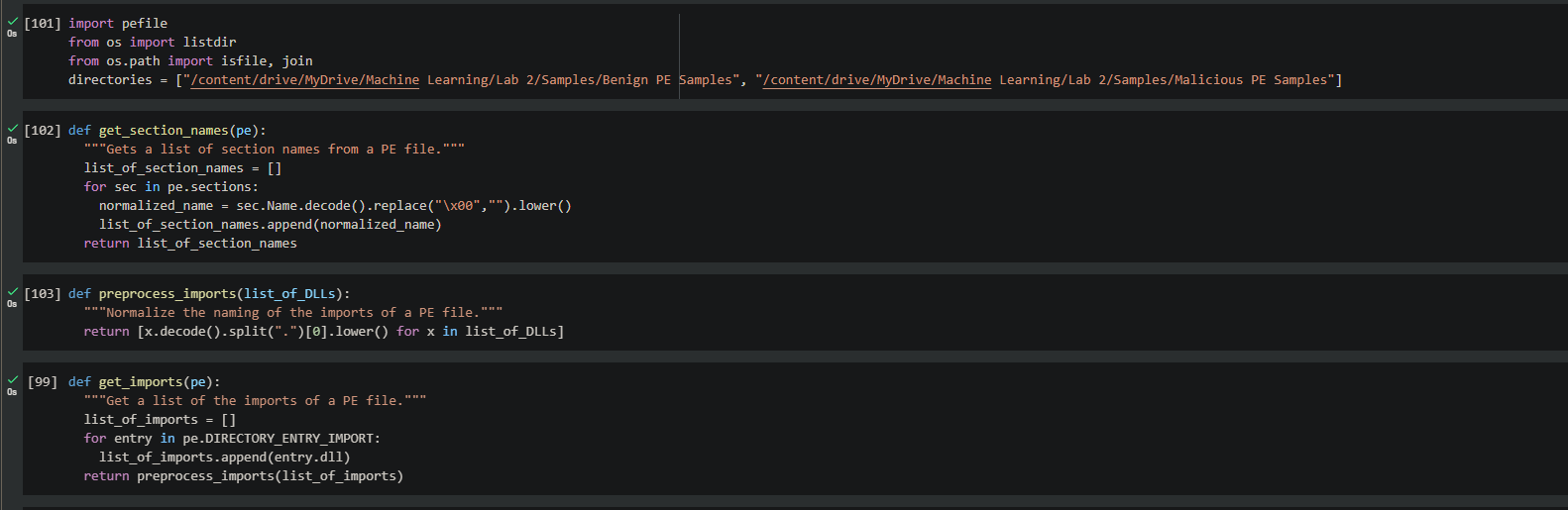


In tất cả thông tin dump từ PE

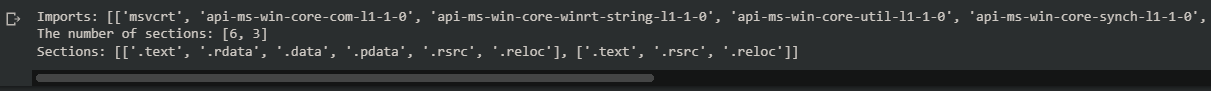


**d) Featurizing the PE header**

Nội dung chi tiết rõ hơn trong file Notebook Lab 2.ipynp



*Yêu cầu 2. Sinh viên cho biết quả của đoạn code trên*



Kết quả trả về bao gồm

- Tên các thư viện import

- Số lượng Sections

- Tên các Sections

**2. Phân tích động mã độc**

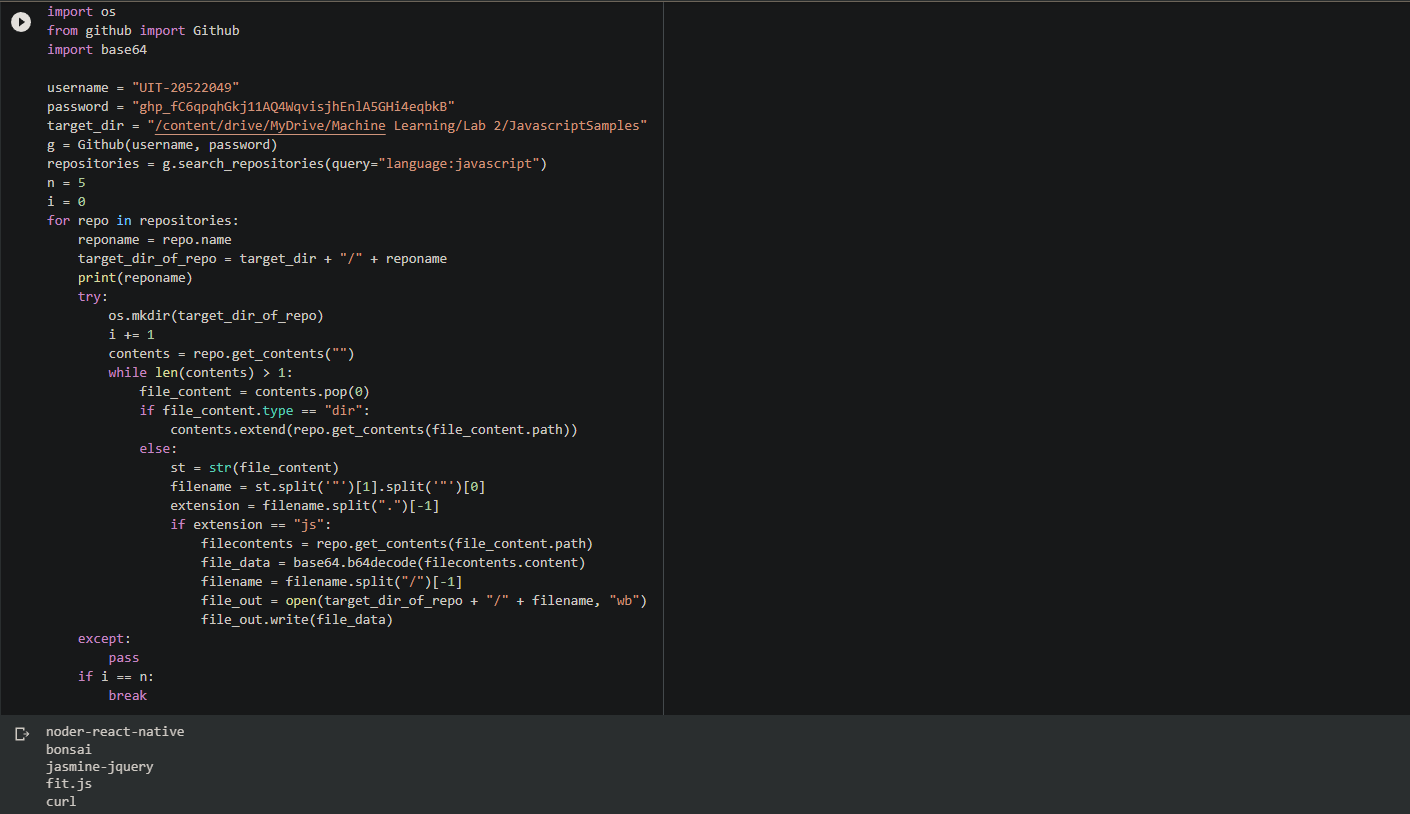
*Yêu cầu 3. Sinh viên tự tìm hiểu, cài đặt (https://cuckoo.sh/docs/introduction/index.html), thực hiện và trình bày phân tích động một tập tin PE.*

Tụi em có tìm hiểu và thấy được cài đặt link github (<https://github.com/ForeGuards/Cuckoo-Installation-Guide/blob/main/installation.txt#L186>) nhưng trong quá trình cài đặt gặp lỗi nhưng không tìm thấy cách fix lỗi trên mạng, nên em xin bỏ qua không làm câu này.

**3. Sử dụng máy học phát hiện loại tập tin**

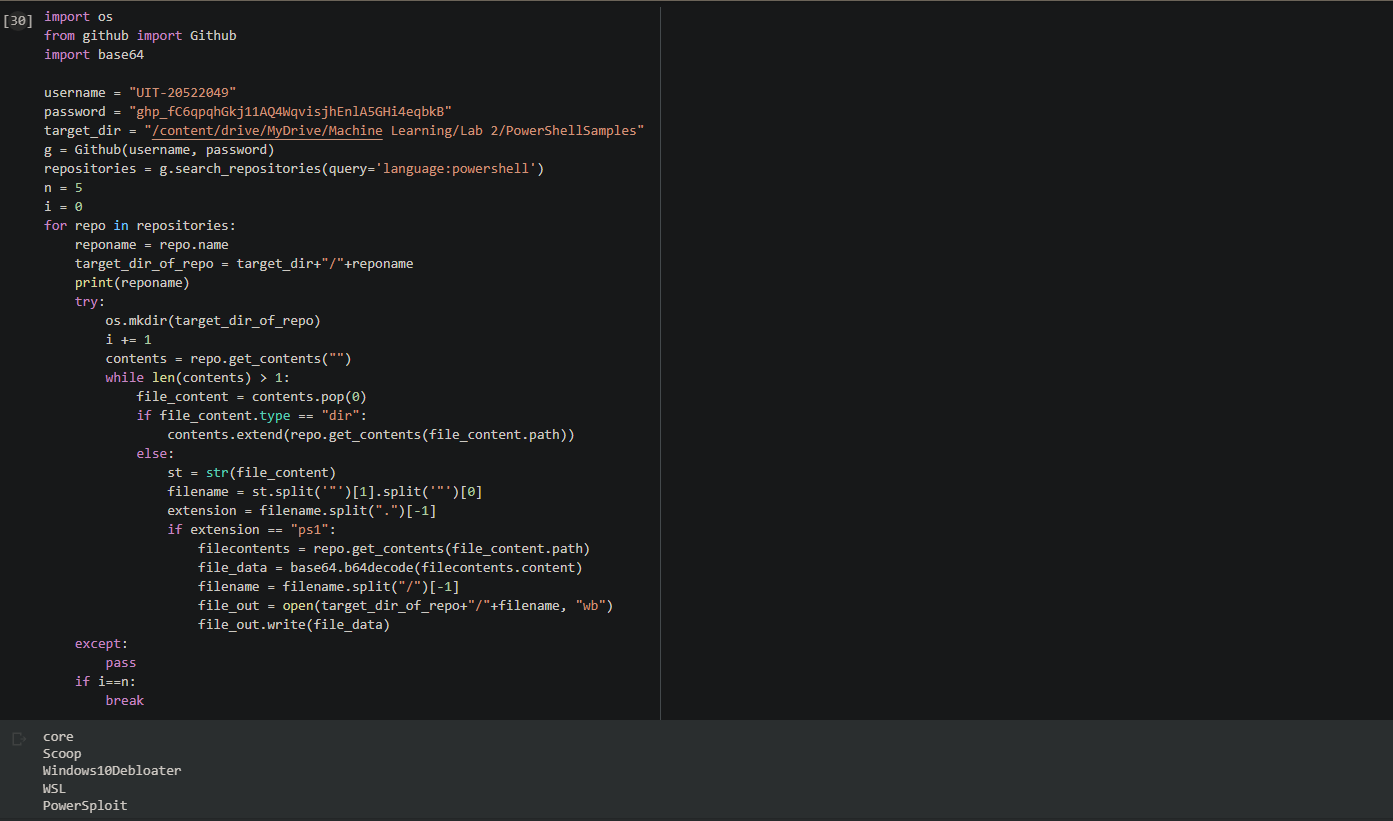
**e) Scraping GitHub cho các loại tập tin đặc biệt**

Truy vấn JavaScript

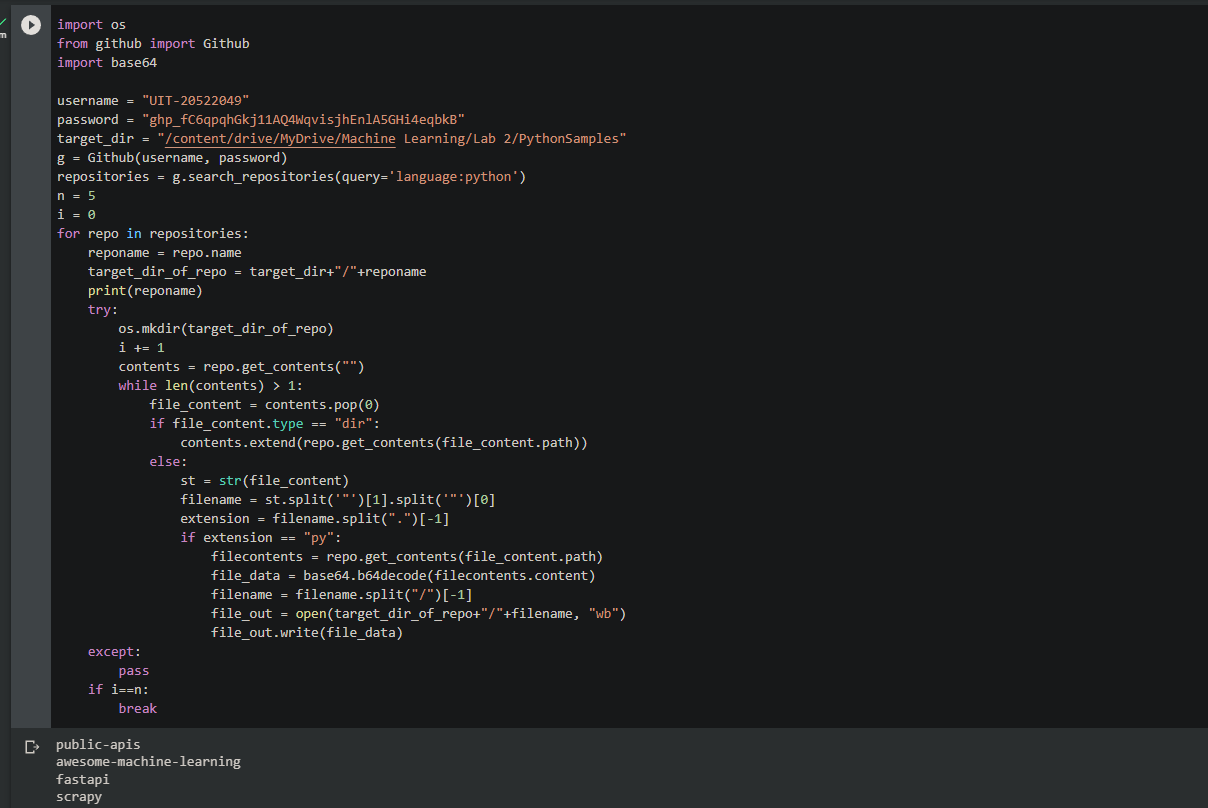


*Yêu cầu 4. Tương tự sinh viên hãy làm các câu truy vấn về Python và Powershell*

Truy vấn về Powwershell



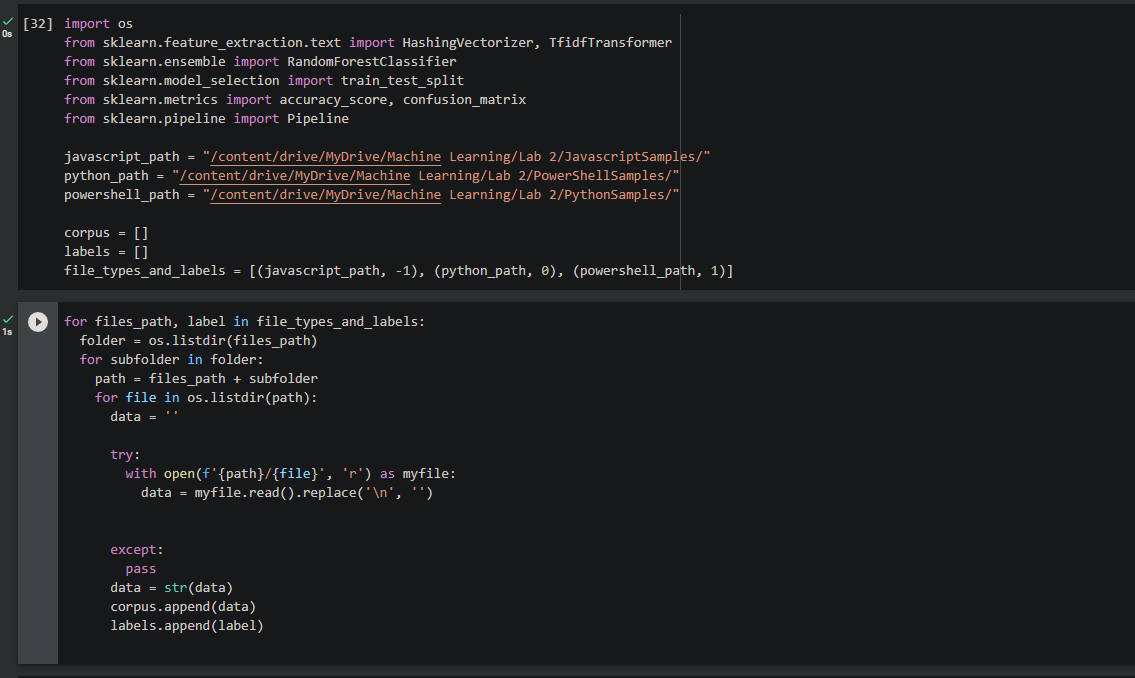
Truy vấn về Python

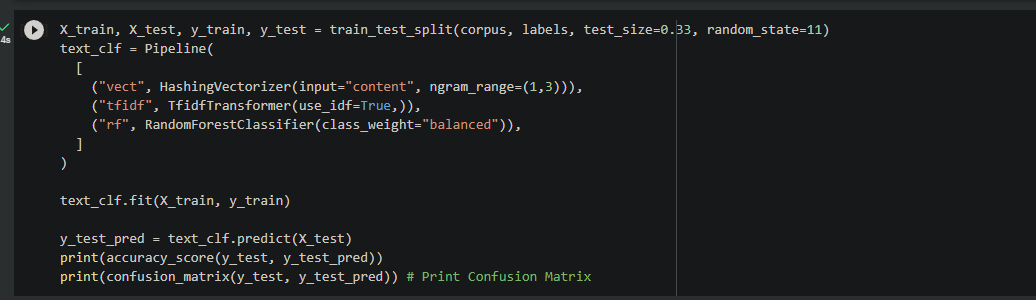


**f) Phân loại tập tin theo kiểu**

Ta sẽ phân loại các tập tin JavaScript, Python, hoặc PowerShell từ phần trên đã thu thập.

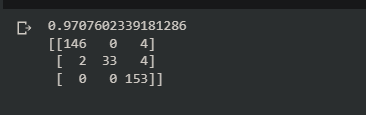
(Tham khảo kỹ hơn tại file Lab 2.ipynb)





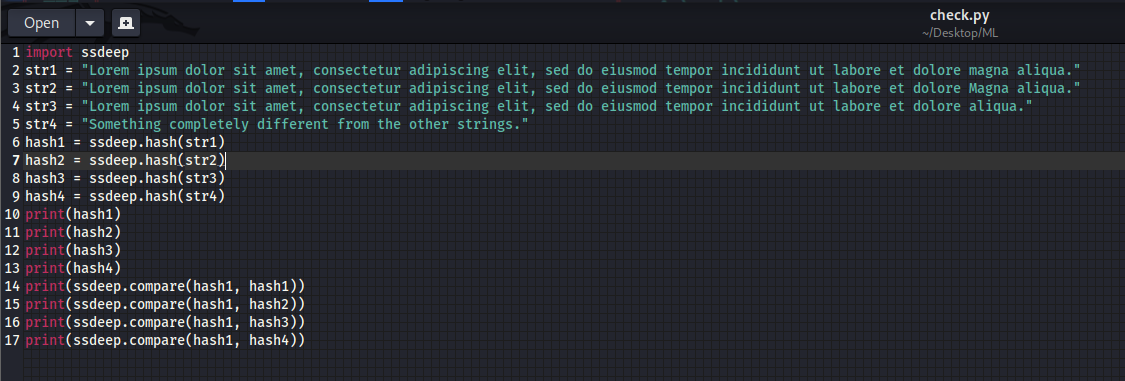
*Yêu cầu 5. Sinh viên cho biết kết quả của đoạn code trên*

Kết quả cho ra accuracy\_score là 97.07% và cofusion\_matrix

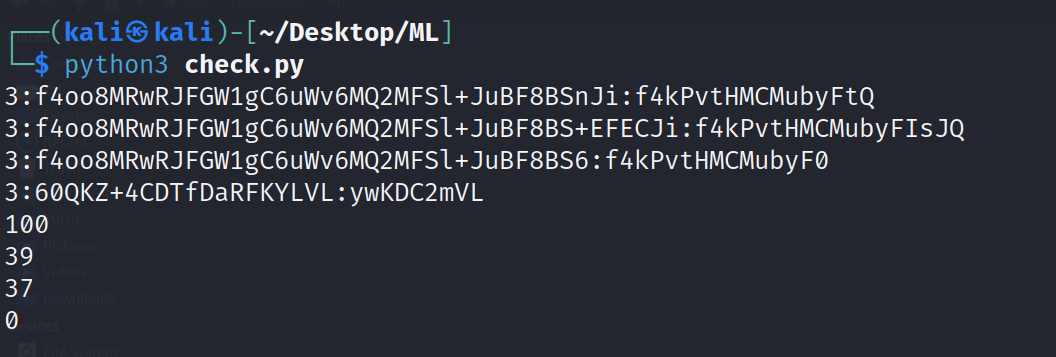


**g) Đo lường sự giống nhau giữa hai chuỗi**

Tạo code so sánh các chuỗi



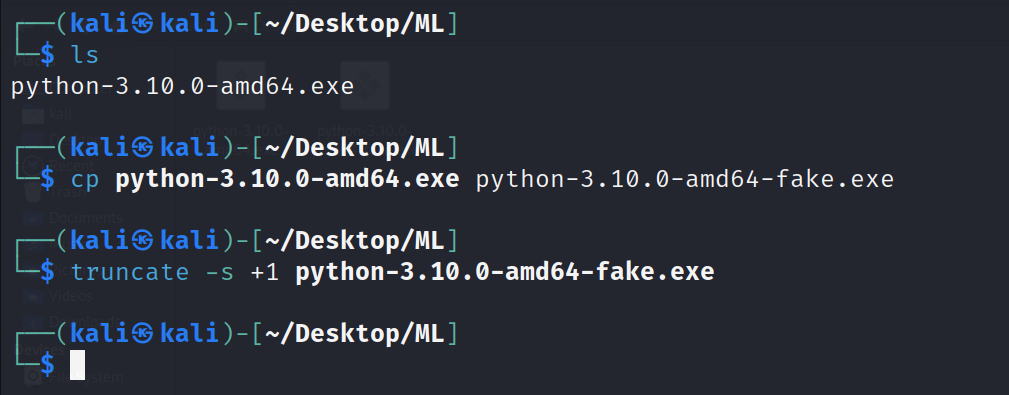
Kết quả thực thi code



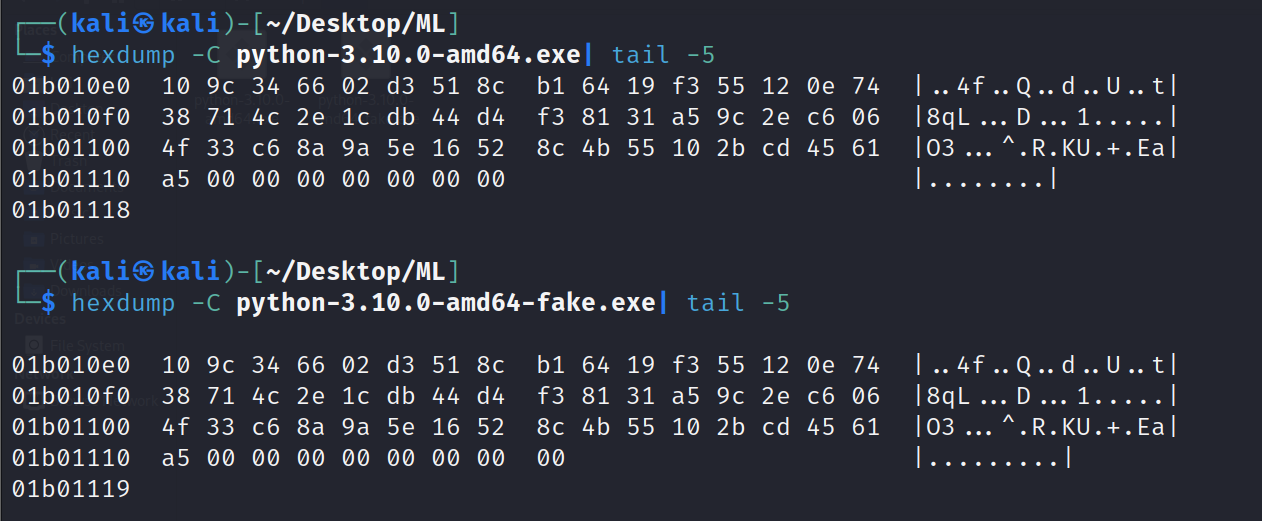
- 4 hàng đầu có hash khác nhau vì các chuỗi khác nhau, 4 hàng cuối là so sánh tỉ lệ giống nhau giữa các chuỗi

**h) Đo lường mức độ giống nhau giữa hai tập tin**

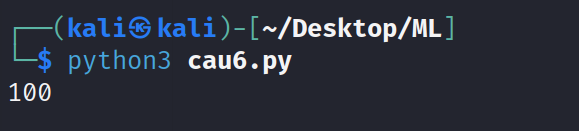
Đầu tiên tạo một bản sao từ tập tin python-3.10.0-amd64.exe thành python-3.10.0-amd64-fake.exe và thêm vài null bytes vào file đó bằng lệnh truncate -s +1 python-3.10.0-amd64-fake.exe



Dùng hexdump để xem sự khác nhau giữa hai tập tin



Dùng ssdeep để so sánh 2 tập tin



*Yêu cầu 6. Sinh viên cho biết quả của đoạn code trên*

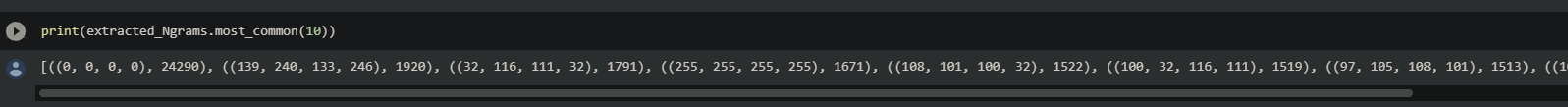
Kết quả 2 file khớp nhau hoàn toàn.

1. **Trích xuất N-grams**

Tham khảo thêm trong file notebook Lab 2.ipynb



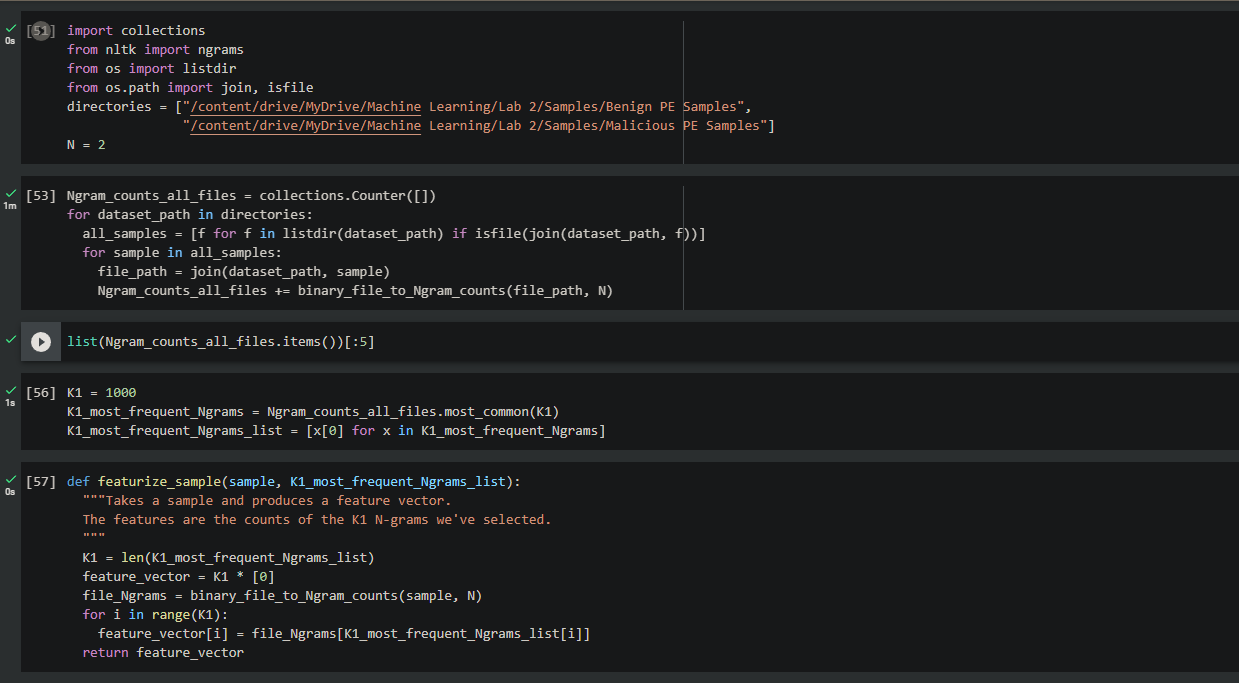
Ta được kết quả của đoạn code trên

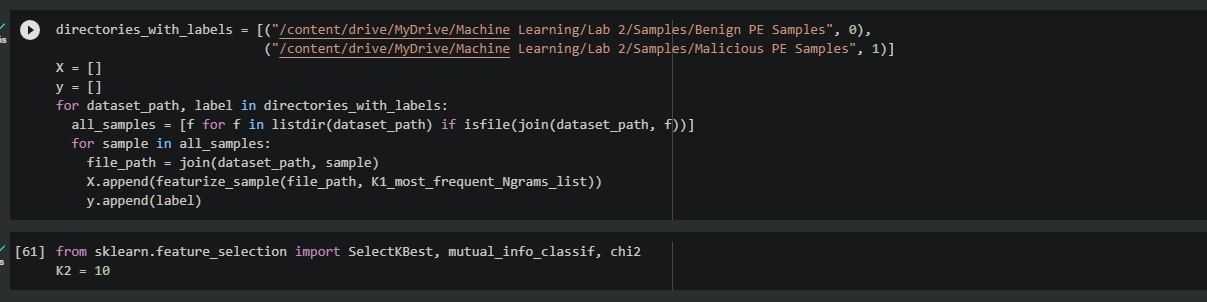


[((0, 0, 0, 0), 24290),   
((139, 240, 133, 246), 1920),   
((32, 116, 111, 32), 1791),   
((255, 255, 255, 255), 1671),   
((108, 101, 100, 32), 1522),   
((100, 32, 116, 111), 1519),   
((97, 105, 108, 101), 1513),   
((105, 108, 101, 100), 1513),   
((70, 97, 105, 108), 1505),   
((101, 100, 32, 116), 1503)]

**j) Chọn N-grams tốt nhất**

Tham khảo thêm trong file notebook Lab 2.ipynb

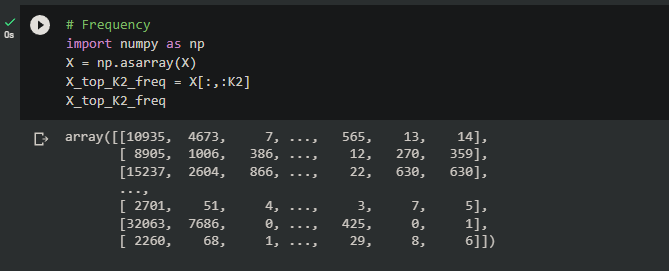




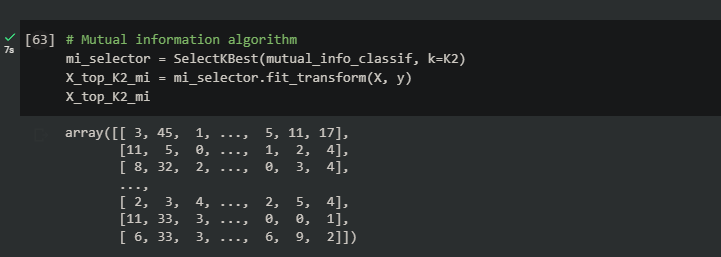
*7. Sinh viên cho biết quả của đoạn code trên*

Kết quả lấy N-grams theo từng kiểu

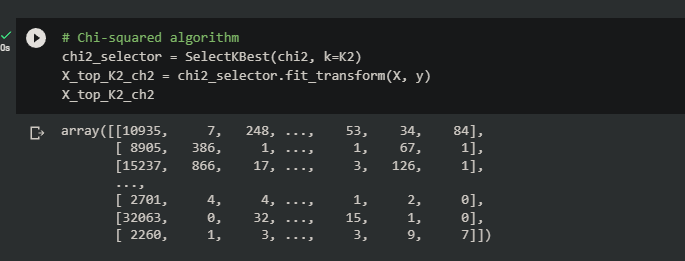
**Frequency:** Chọn N-grams phổ biến



**Mutual information:** Chọn N-grams có xếp hạng cao theo thuật toán mutual information



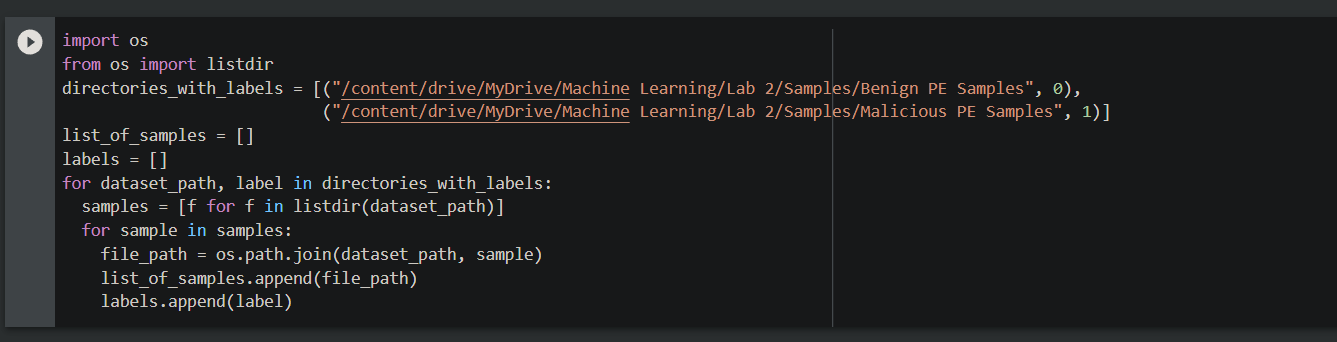
**Chi-squared:** Chọn N-grams có xếp hạng cao theo thuật toán chi squared

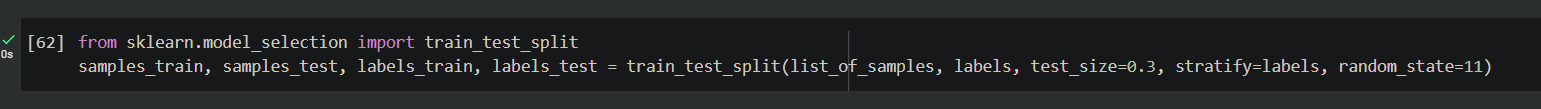


**4. Xây dựng trình phát hiện phần mềm độc hại bằng phân tích tĩnh**

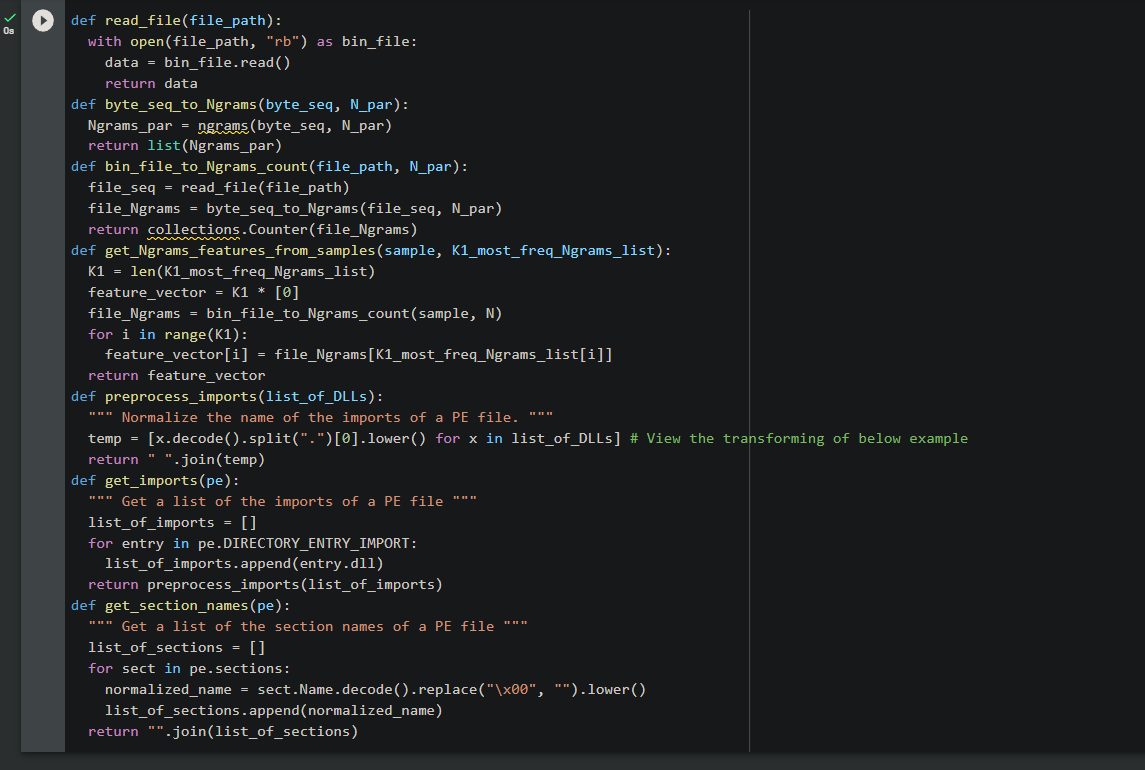
8. *Sinh viên hoàn thành các bước sau*

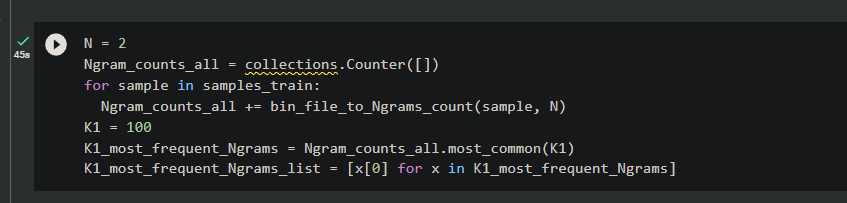
B1. Tạo list các mẫu và gán nhãn cho chúng.

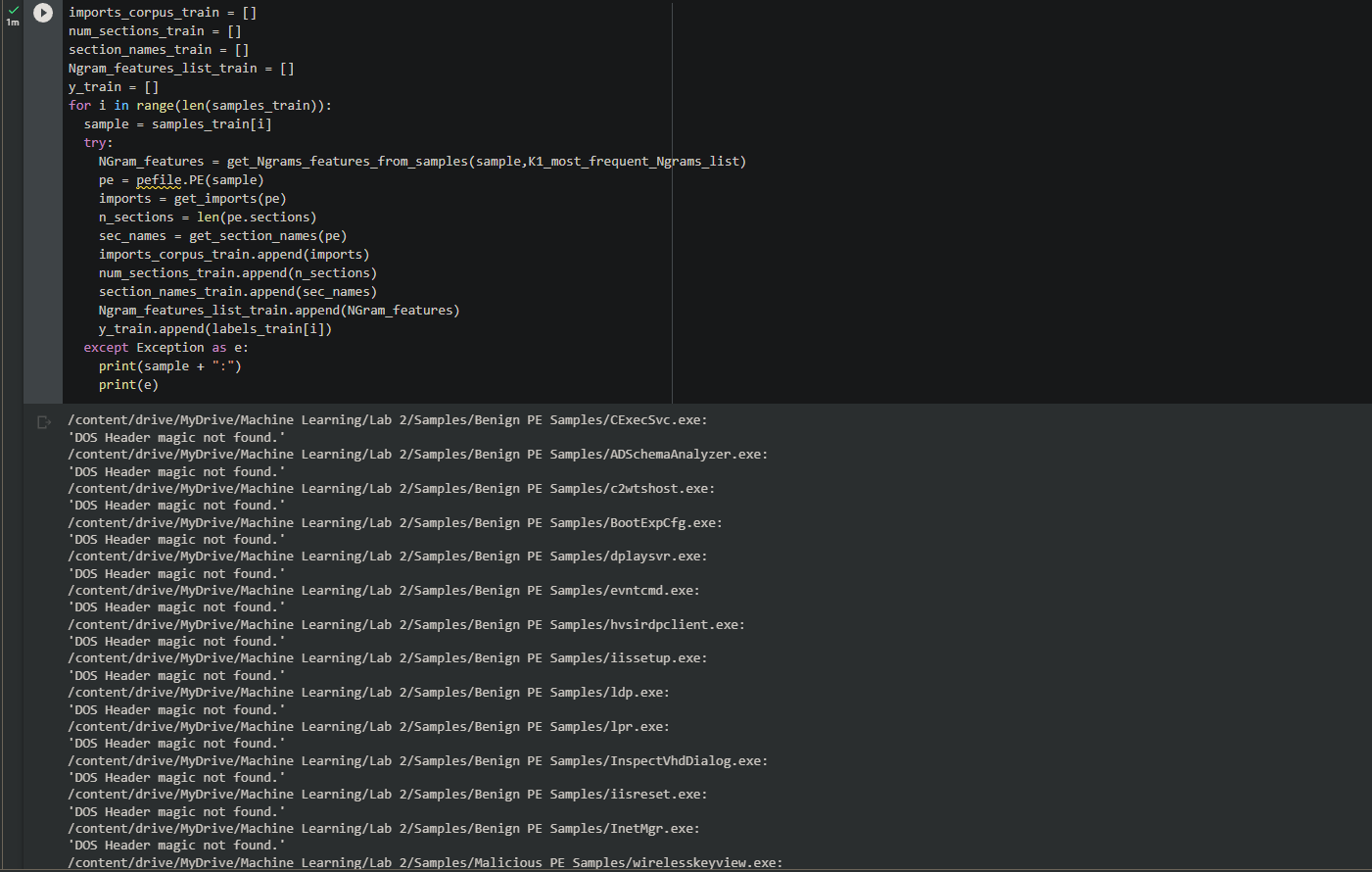
  
B2. Chia dữ liệu train-test



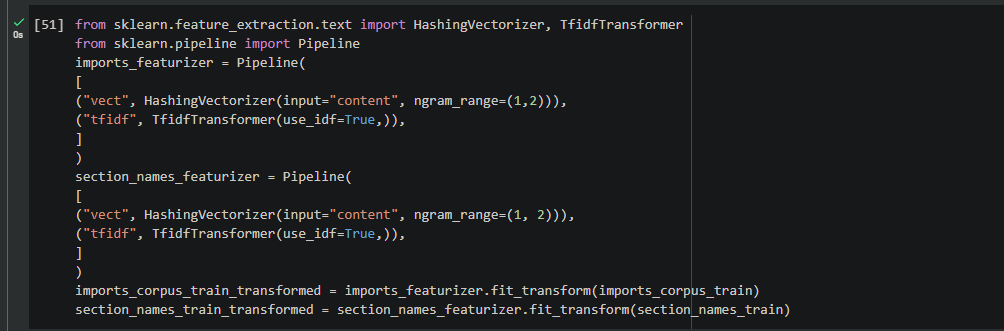
B3. Các hàm lấy thuộc tính

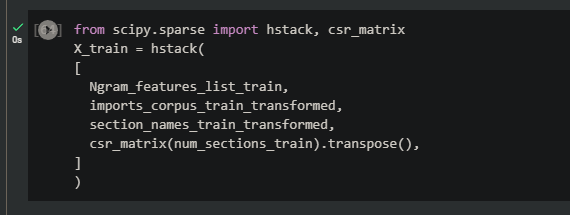
  
B4. Chọn 100 thuộc tính phổ biến với 2-grams

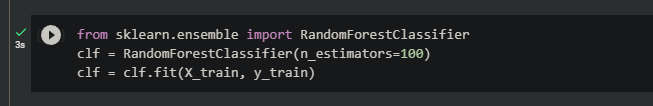
  
B5. Trích xuất số lượng N-grams count, section names, imports và số lượng sections của mỗi mẫu trong train-test.

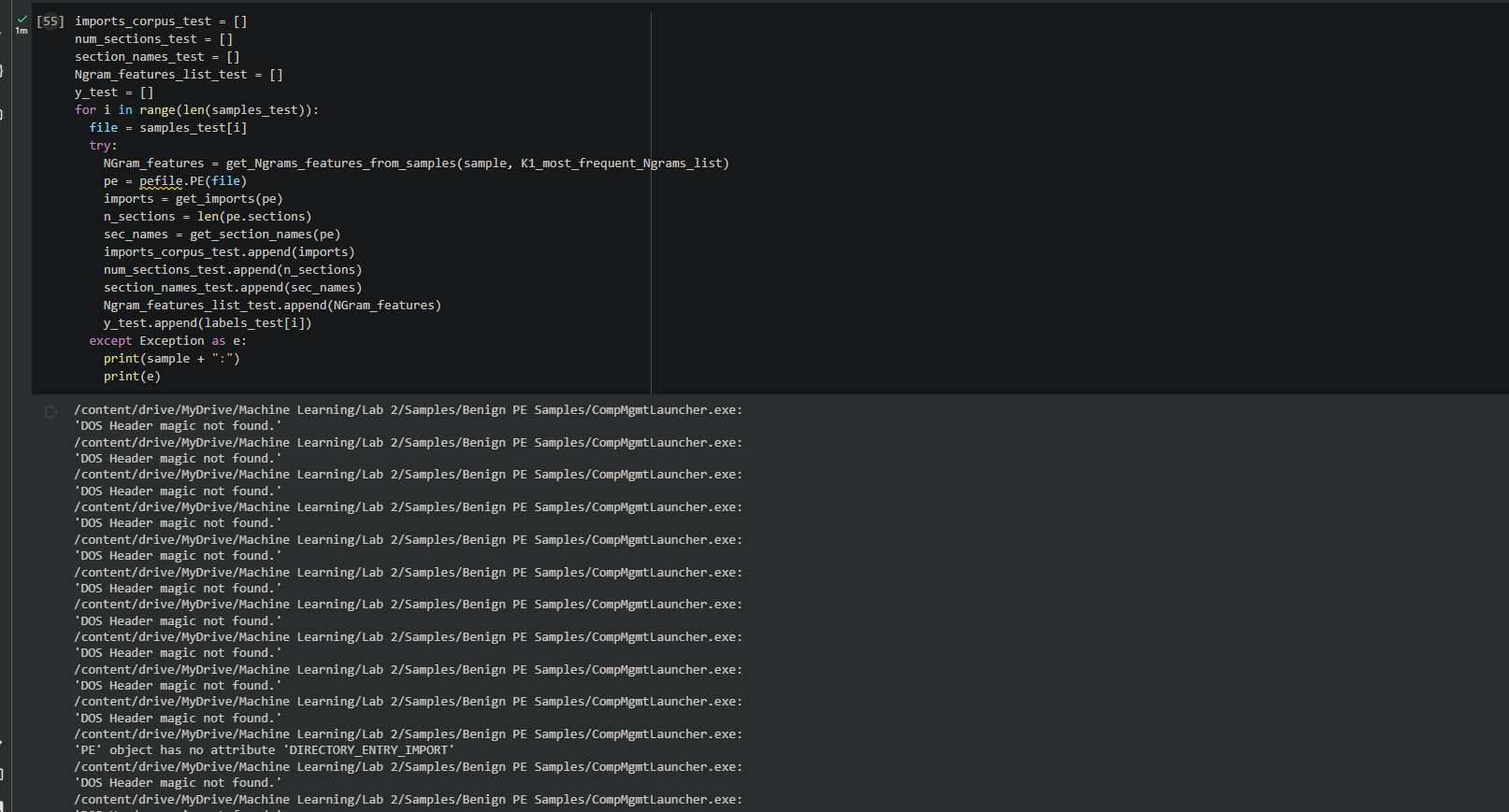


B6. Sử dụng hàm băm tfidf để chuyển imports, section names từ văn bản thành dạng số

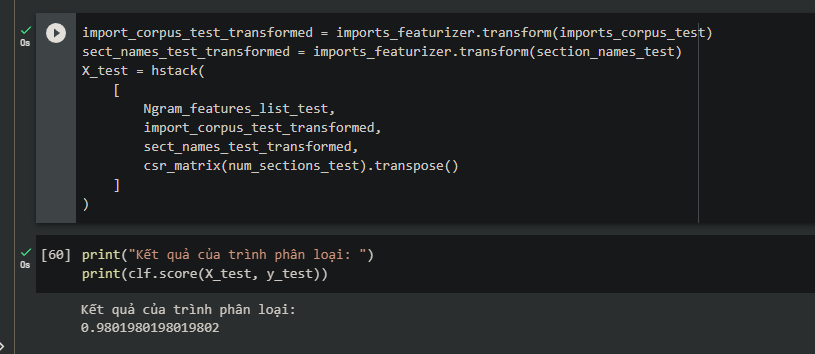
  
B7. Kết hợp các vector thuộc tính thành 1 mảng.

  
B8. Ta huấn luyên bằng phân loại Random Forest cho tập train

  
B9. Thu thập các thuộc tính của tập test, giống như tập huấn luyện



B10. Ta chuyển đổi vector từ thuộc tính test, và kiểm tra kết quả của trình phân loại.



(Xem chi tiết tại file Notebook yeucau8.ipynb)

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-0)