DầuKhí
KHOA HỌC, CÔNG NGHỆ VÀ ĐỔI MỚI SÁNG TẠO PETROVIETNAM

ỨNG DỤNG PYTHON VÀ MACHINE LEARNING TRONG XỬ LÝ VÀ PHÂN LOẠI DỮ LIỆU XUẤT NHẬP KHẨU

Đỗ Hồng Hạnh[1], Đoàn Tiến Quyết[1], Đoàn Trọng Sinh[2], Nguyễn Bằng Linh[1]
[1]Viện Dầu khí Việt Nam (VPI)
[2]Tập đoàn Công nghiệp - Năng lượng Quốc gia Việt Nam (PVN)
Email: hanhdh@vpi.pvn.vn
https://doi.org/10.47800/PVSI.2025.03-05

Tóm tắt

Dữ liệu xuất nhập khẩu tại Việt Nam đang gia tăng cả về quy mô và độ phức tạp, tạo ra thách thức lớn trong việc chuẩn hóa và phân loại thông tin khai báo hải quan. Bài viết đề xuất quy trình xử lý dữ liệu tự động hóa sử dụng ngôn ngữ lập trình Python, nhằm nâng cao hiệu quả và tính nhất quán trong phân tích thông tin khai báo hải quan. Tập dữ liệu đầu vào gồm hơn 10.000 bản ghi xuất nhập khẩu thực tế, được xử lý qua các bước kỹ thuật như: chuẩn hóa tên hàng, quy đổi đơn vị tính, tính toán chỉ tiêu định lượng và gán nhãn nhóm sản phẩm dựa trên từ khóa.

Kết quả cho thấy quy trình này vận hành hiệu quả trên tập dữ liệu có quy mô vừa và tính phức tạp cao, đồng thời đảm bảo độ chính xác và tính đồng nhất trong phân loại nhóm sản phẩm. Trên cơ sở đó, nhóm tác giả đề xuất tích hợp mô hình học máy như công cụ hỗ trợ để nâng cao khả năng khái quát hóa và thích ứng với các trường hợp ngoại lệ, khi tên hàng không được chuẩn hóa và thay đổi liên tục theo thực tiễn thương mại quốc tế.

Từ khóa: Xuất nhập khẩu, học máy, chuẩn hóa dữ liệu, TF-IDF, random forest, Python.

1. Giới thiệu

Dữ liệu xuất nhập khẩu đóng vai trò thiết yếu trong việc phân tích chuỗi cung ứng, theo dõi thương mại và quản lý nhà nước. Tuy nhiên, tính chất phi cấu trúc và không đồng nhất của dữ liệu đầu vào, đặc biệt ở trường "Tên hàng" hoặc "Mã HS khai báo", gây cản trở lớn cho việc khai thác thông tin. Việc xử lý thủ công hoặc bán thủ công bằng Excel thường tiềm ẩn sai sót, thiếu tính mở rộng và khó kiểm soát chất lượng. Do đó, yêu cầu đặt ra là cần xây dựng quy trình xử lý dữ liệu hiệu quả, linh hoạt và có khả năng tự động hóa cao. Trong nghiên cứu này, nhóm tác giả áp dụng các thư viện Python để xây dựng một chuỗi quy trình (pipeline) xử lý dữ liệu nhập khẩu nhựa từ file Excel được cung cấp bởi bên thứ ba. Các bước gồm làm sạch dữ liệu, ánh xạ, chuẩn hóa đơn vị, phân loại sản phẩm và tính toán các chỉ tiêu định lượng. Kết quả được so sánh với phương pháp truyền thống nhằm đánh giá hiệu quả xử lý và đề xuất khả năng mở rộng ứng dụng học máy (machine learning).

Ngày nhận bài: 12/6/2025
Ngày nhận bài sửa: 12 - 29/6/2025
Ngày chấp nhận đăng: 29/6/2025

Trong thực tế, dữ liệu khai báo xuất nhập khẩu tại Việt Nam chủ yếu được xử lý thủ công hoặc bán thủ công bằng phần mềm Excel. Mặc dù phổ biến nhờ tính trực quan và khả năng tiếp cận rộng rãi, nhưng công cụ này đang bộc lộ nhiều bất cập trong việc xử lý tập dữ liệu có quy mô vừa và tính phức tạp cao, không chuẩn hóa. Các nghiên cứu thực tiễn đã cho thấy các thao tác như chuẩn hóa tên hàng, phân loại từ khóa, tính toán định lượng và kiểm tra logic dữ liệu nếu thực hiện hoàn toàn trong Excel sẽ tiêu tốn nhiều thời gian, dễ xảy ra lỗi và thiếu tính mở rộng. Ngược lại, quy trình xử lý dữ liệu được thiết kế và thực thi bằng Python cho thấy ưu thế vượt trội trên nhiều khía cạnh: hiệu suất xử lý, độ chính xác, khả năng tái sử dụng.

Về hiệu suất xử lý, nghiên cứu của Kelvin và cộng sự [1] đã chỉ ra rằng Python có mức sử dụng CPU trung bình là 10,01%, thấp hơn so với 30,8% của Excel khi thực hiện các tác vụ cơ bản như đọc dữ liệu, áp dụng hàm logic và tạo báo cáo tổng hợp. Thử nghiệm này sử dụng kiểm định thống kê Wilcoxon để xác nhận sự khác biệt có ý nghĩa giữa 2 nền tảng. Điều này không chỉ phản ánh hiệu quả sử dụng tài nguyên hệ thống mà còn nhấn mạnh rằng Python vận hành ổn định hơn trong các tác vụ xử lý tập dữ liệu có quy

mô vừa và tính phức tạp cao, một đặc điểm thường xuyên xuất hiện trong phân tích dữ liệu xuất nhập khẩu.

Nghiên cứu của Mansour và cộng sự [2] thực hiện trên các mô hình dòng tiền đã minh chứng khả năng rút ngắn thời gian xử lý lên đến 99% khi chuyển từ Excel sang Python. Trong trường hợp cụ thể, mô hình dòng tiền được tính toán trong Excel mất 179 giây, trong khi phiên bản tương đương trong Python chỉ mất 0,8 giây, mà vẫn duy trì sai số tuyệt đối dưới 0,01%. Ngoài ra, Python còn cho phép triển khai các kỹ thuật tính toán hiện đại như vector hóa (trong Pandas) và lập trình mảng (NumPy), giúp tăng tốc xử lý đồng thời giữ được khả năng mở rộng với các dữ liệu phức tạp hơn.

Về độ chính xác và hiệu quả, nghiên cứu của Panko và Halverson [3] cho thấy trong các kiểm toán thực tế, tỷ lệ ô sai sót trong bảng tính thường dao động từ 1% đến 3%, có trường hợp thấp hơn (0,38%). Kết quả thí nghiệm của tác giả với Excel cũng ghi nhận rằng sinh viên làm 1 mình tạo ra bảng tính có tỷ lệ lỗi trung bình 4,6%, trong khi làm theo nhóm 3 người con số này giảm xuống còn 1,0%. Trong khi đó, các nền tảng lập trình như Python có thể tích hợp logic kiểm tra điều kiện, kiểm soát lỗi và tự động hóa xử lý, giúp giảm rủi ro sai sót phát sinh từ yếu tố con người. Quy trình Excel phụ thuộc nhiều vào thao tác thủ công nên dễ mắc lỗi định dạng, gán sai đơn vị, hoặc bỏ sót bước kiểm tra logic. Nghiên cứu của Panko [4] đã nhấn mạnh rằng gần như tất cả bảng tính đều chứa lỗi. Với tỷ lệ lỗi công thức ở mức 5%, khi 1 bảng tính có khoảng 100 công thức gốc thì xác suất tồn tại ít nhất 1 lỗi gần như đạt 100%. Bên cạnh đó, khả năng phát hiện lỗi của người dùng còn hạn chế: các nghiên cứu cho thấy cá nhân khi kiểm tra bảng tính chỉ phát hiện được khoảng 50 - 60% lỗi. Trong khi đó, môi trường sử dụng ngôn ngữ Python có thể tích hợp các bước chuẩn hóa, lọc ngoại lệ và kiểm tra tính nhất quán của dữ liệu ngay trong quy trình xử lý, giúp tỷ lệ dòng dữ liệu hợp lệ sau xử lý đạt trên 98%, cao hơn đáng kể so với khoảng 90 - 92% của phương pháp thủ công.

Về khả năng tái sử dụng và mở rộng, Excel thường yêu cầu người dùng chỉnh sửa lại toàn bộ công thức hoặc logic nếu danh mục sản phẩm thay đổi hoặc mở rộng. Panko [4] cảnh báo rằng khi công thức phải tham chiếu đến các ô nằm ngoài màn hình hoặc trên các trang tính khác, tỷ lệ lỗi phát sinh tăng đáng kể, đồng thời làm giảm khả năng duy trì và kiểm tra mô hình. Ngược lại, ngôn ngữ Python cho phép tổ chức từ khóa và logic xử lý thành các module riêng biệt (function, script, config), nhờ đó dễ dàng cập nhật và mở rộng hệ thống mà không cần can thiệp thủ công vào từng bước xử lý. Nghiên cứu của Ziogas và cộng sự [5] đã chứng minh rằng Python có thể đạt được hiệu suất cao và khả năng mở rộng tốt khi xử lý tập dữ liệu có quy mô vừa và tính phức tạp cao, đặc biệt khi sử dụng các công cụ như NumPy và các thư viện hỗ trợ tính toán song song. Điều này cho thấy Python không chỉ đơn giản và dễ học mà còn mạnh mẽ trong việc xử lý tập dữ liệu có quy mô vừa và tính phức tạp cao.

Nhìn chung, các bằng chứng định lượng và thực nghiệm đã chỉ ra rằng quy trình xử lý dữ liệu bằng Python không chỉ cải thiện đáng kể hiệu suất, mà còn nâng cao độ chính xác, khả năng kiểm soát chất lượng và khả năng thích ứng với các yêu cầu thay đổi trong thực tế. Các doanh nghiệp và cơ quan quản lý ngày càng phải xử lý lượng dữ liệu phức tạp và đa dạng, việc chuyển đổi từ Excel sang nền tảng xử lý tự động hóa như Python giúp nâng cao năng lực xử lý và phân tích chuyên sâu trên tập dữ liệu.

2. Quy trình xử lý dữ liệu bằng ngôn ngữ lập trình Python

2.1. Mô tả dữ liệu

Dữ liệu sử dụng trong nghiên cứu được trích xuất từ các bộ số liệu khai báo xuất nhập khẩu hàng tháng do cơ quan hải quan công bố hoặc được cung cấp bởi bên thứ 3. Đây là dữ liệu thực tế phản ánh hoạt động giao thương thường xuyên, được chuẩn hóa theo từng kỳ báo cáo, với quy mô mỗi tháng dao động trong khoảng từ 2.000 - 3.000 dòng dữ liệu, tập trung vào các mặt hàng nhựa polypropylene (PP). Mỗi dòng tương ứng với một tờ khai nhập khẩu gồm các trường thông tin như mã tờ khai, mã chi cục, loại hình nghiệp vụ, mã HS khai báo, tên hàng, số lượng, đơn vị tính và trị giá giao dịch. Quy trình xử lý dữ liệu được thiết kế với mục tiêu tự động hóa toàn bộ quá trình xử lý thông tin nhập khẩu nhựa PP từ file Excel đầu vào. Các bước gồm làm sạch dữ liệu, chuẩn hóa chuỗi văn bản, ánh xạ thông tin chi cục và loại hình, phân loại sản phẩm theo từ khóa kỹ thuật, quy đổi đơn vị và tính toán chỉ tiêu định lượng như sản lượng tính theo tấn, đơn giá và trị giá theo USD. Kết quả đầu ra có thể được sử dụng để thống kê, phân tích, trực quan hóa hoặc làm đầu vào cho các mô hình học máy nhằm hỗ trợ ra quyết định quản lý.

Điểm đáng chú ý là quy mô dữ liệu theo từng tháng không lớn, có tính chất đơn lẻ, chi phí triển khai và vận hành hệ thống hiện tại rất thấp, ngay cả khi áp dụng cho nhiều tháng hoặc nhiều sản phẩm khác nhau, hệ thống vẫn duy trì khả năng xử lý ổn định mà không đòi hỏi nâng cấp tài nguyên tính toán đáng kể. Do đó, quy trình xử lý này đặc

biệt phù hợp để áp dụng riêng cho từng nhóm mặt hàng xuất nhập khẩu theo mã HS hoặc theo từng sản phẩm, như nhóm nhựa PP trong nghiên cứu. Điều này mở ra tiềm năng triển khai linh hoạt theo từng ngành hàng, giúp tăng cường khả năng phân tích theo chiều sâu và hỗ trợ hoạt động giám sát chuyên đề trong lĩnh vực xuất nhập khẩu.

2.2. Chuẩn hóa toàn bộ dữ liệu văn bản

Quy trình xử lý dữ liệu xuất nhập khẩu được thiết kế và thực hiện trong môi trường Google Colab (Python 3.10.2), tối ưu hóa cho tập dữ liệu có quy mô vừa và tính phức tạp cao. Các trường văn bản như tên doanh nghiệp xuất nhập khẩu, tên hàng thường tồn tại dưới nhiều định dạng không đồng nhất, chứa ký tự đặc biệt. Để đảm bảo khả năng xử lý và phân tích tự động, tất cả các cột dạng văn bản được chuẩn hóa bằng hàm biến đổi tùy chỉnh chuyển về chữ in hoa, loại ký tự đặc biệt và chuẩn hóa khoảng trắng.

2.3. Kiểm tra dữ liệu

Dữ liệu được rà soát theo 3 tiêu chí: (i) Tính chính xác: Kiểm tra đơn vị và giá trị của các trường dữ liệu để loại bỏ các dòng bất hợp lệ; (ii) Tính nhất quán: Kiểm tra định dạng dữ liệu; (iii) Tính đầy đủ: Kiểm tra dữ liệu thiếu.

2.4. Trích xuất và ánh xạ mã chi cục, loại hình

Trong cấu trúc chuỗi "Tờ khai", thông tin quan trọng như mã chi cục hải quan và mã loại hình nghiệp vụ thường được nhúng dưới dạng ký hiệu kết hợp, ví dụ: 106977513900/C11/47I1. Để phục vụ mục đích phân tích theo vùng địa lý và mục đích nghiệp vụ, hệ thống thực hiện trích xuất có điều kiện từ chuỗi dạng này như sau:

- Mã chi cục được xác định là 4 ký tự cuối cùng trong chuỗi "Tờ khai" sau khi loại bỏ khoảng trắng và chuẩn hóa viết hoa.

- Mã loại hình được trích xuất là ký hiệu ở vị trí thứ hai sau dấu gạch chéo (/), thường theo định dạng chuẩn của Tổng cục Hải quan (ví dụ: C11, A12, B13...).

Quy trình trích xuất được chuẩn hóa bằng phép biến đổi chuỗi như sau:

Trích xuất mã chi cục: 4 ký tự cuối

df["Mã chi cục"] = df["Tờ khai"].str[-4:]

Trích xuất mã loại hình: phần tử thứ hai trong chuỗi tách bằng "/"

df["Mã loại hình"] = df["Tờ khai"].str.split("/").str[1]

Sau đó, mã chi cục được ánh xạ với bảng mã địa phương để gán nhãn miền (Bắc, Trung, Nam), còn mã loại hình được đối chiếu với bảng danh mục mã nghiệp vụ để phục vụ phân tích mục đích khai báo như kinh doanh, gia công, sản xuất xuất khẩu.

2.5. Phân loại nhóm sản phẩm

Việc phân loại tên hàng vào các nhóm sản phẩm như PP_HOMO, PP_COPOLYMER, PP_COMPOUND được thực hiện bằng phương pháp khớp từ khóa. Danh sách từ khóa đã chuẩn hóa được ánh xạ với từng nhãn phân loại. Hệ thống thực hiện quy trình này tự động dựa trên Regex hoặc tìm kiếm chuỗi, có khả năng xử lý toàn bộ dữ liệu nhanh chóng. Với những dòng dữ liệu chưa được khớp theo phương pháp ánh xạ do thiếu hụt danh sách từ khóa, tác giả tiếp tục phân loại bằng phương pháp học máy được giới thiệu trong nghiên cứu này. Trong thực tế triển khai, dù hệ thống phân loại bằng từ khóa (rule-based) và mô hình học máy đã cho thấy hiệu quả nhất định trong xử lý tên hàng nhập khẩu, vẫn tồn tại một tỷ lệ nhỏ các trường hợp không thể phân loại do không khớp với từ khóa hiện có hoặc chứa nội dung không rõ ràng. Để đảm bảo tính bao phủ và độ tin cậy của toàn bộ quy trình, nghiên cứu đề xuất tích hợp một cơ chế phản hồi có sự tham gia của chuyên gia (Human-in-the-loop - HITL) nhằm xử lý linh hoạt các ngoại lệ và cải thiện hệ thống theo thời gian. Cơ chế HITL được triển khai như một bước hậu kiểm, cụ thể như sau:

(i) Sau khi hệ thống thực hiện phân loại tên hàng bằng cả rule-based và mô hình học máy, các dòng không phân loại được (unclassified) hoặc có độ tin cậy thấp sẽ được tự động ghi log vào một bảng riêng (ví dụ: unclassified_items.xlsx) kèm theo các thông tin hỗ trợ như mô tả tên hàng, ngày khai báo, đơn vị nhập khẩu và kết quả phân loại gần đúng (nếu có);

(ii) Các dòng này sau đó được chuyển tới bộ phận chuyên môn để rà soát và gán nhãn thủ công dựa trên hiểu biết kỹ thuật và kinh nghiệm nghiệp vụ;

(iii) Kết quả rà soát được lưu lại và sử dụng để cập nhật vào các tập luật (keyword_map.json) hoặc bổ sung vào tập huấn luyện của mô hình học máy ở các chu kỳ sau, từ đó nâng cao độ chính xác và khả năng thích ứng của hệ thống.

Cơ chế đảm bảo rằng dữ liệu không rõ ràng hoặc chưa từng xuất hiện sẽ không bị loại bỏ, mà thay vào đó trở thành đầu vào cho quá trình cải tiến liên tục của hệ thống. Hơn nữa, việc lưu lại log đầy đủ giúp quá trình rà soát minh bạch, dễ kiểm tra và có thể đánh giá hiệu quả phân loại

theo thời gian. Đây là một bước quan trọng trong việc xây dựng quy trình xử lý dữ liệu linh hoạt và có khả năng học hỏi, một yêu cầu thiết yếu trong môi trường nghiệp vụ thực tế, nơi dữ liệu phát sinh liên tục với mức độ biến động cao

2.6. Quy đổi đơn vị và tính toán các chỉ tiêu định lượng

Tất cả dữ liệu về khối lượng được quy đổi về đơn vị chuẩn là tấn thông qua bảng chuyển đổi đơn vị. Bước này đảm bảo tính đồng nhất trong báo cáo sản lượng. Trên cơ sở đơn giá và sản lượng đã chuẩn hóa, hệ thống tự động tính ra trị giá USD và các chỉ tiêu thống kê cần thiết.

2.7. Kiểm tra chất lượng dữ liệu đầu ra

Dữ liệu được rà soát theo 3 tiêu chí:

- Loại bỏ các dòng trùng lặp hoàn toàn;

- Loại bỏ giá trị ngoại lai/không phù hợp (outliers) dựa theo thống kê và chuyên môn nghiệp vụ;

- Kiểm tra logic đơn vị và giá trị để loại bỏ các dòng bất hợp lệ.

3. Đánh giá hiệu quả của quy trình xử lý dữ liệu

Để kiểm chứng hiệu quả thực tiễn của quy trình xử lý dữ liệu đề xuất, nghiên cứu tiến hành thực nghiệm trên tập dữ liệu gồm 10.000 dòng dữ liệu xuất nhập khẩu phát sinh trong 1 tháng. Đây là khối lượng dữ liệu hải quan thực tế, phản ánh mật độ nghiệp vụ cao và tính chất biến động liên tục của hoạt động xuất nhập khẩu trong ngắn hạn. Mục tiêu của đánh giá là xác định mức độ cải thiện về hiệu suất xử lý, độ chính xác dữ liệu đầu ra và khả năng mở rộng hệ thống khi sử dụng Google Colab (Python 3.10.2).

3.1. Thời gian xử lý và hiệu suất vận hành

Kết quả đo lường thực tế (Bảng 1) cho thấy tổng thời gian xử lý toàn bộ quy trình đối với 10.000 dòng dữ liệu trong môi trường Google Colab (Python 3.10.2) là khoảng 303 giây (tương đương 5 phút 3 giây). Trong đó, các công đoạn như chuẩn hóa dòng và cột, quy đổi đơn vị và tính toán định lượng, kiểm tra lỗi logic và lọc dòng ngoại lệ đều được thực hiện gần như tức thời, chỉ mất khoảng 1 giây mỗi bước. Điều này đạt được nhờ khả năng xử lý song song và vector hóa mạnh mẽ của các thư viện Python như Pandas, Regex và NumPy, giúp thao tác nhanh chóng trên toàn bộ DataFrame. Ngược lại, khi thực hiện các công đoạn tương tự bằng Excel, thời gian xử lý tổng thể ước tính dao động trong khoảng 3 - 6 phút, tùy thuộc vào cấu hình máy tính và phiên bản Excel sử dụng. Đặc biệt, công đoạn phân loại tên hàng theo từ khóa, vốn đòi hỏi quét chuỗi và so khớp điều kiện, khi thực hiện bằng các hàm như SEARCH, IF, INDEX-MATCH hoặc TEXTJOIN trong Excel có thể gây chậm đáng kể khi áp dụng trên tập dữ liệu có quy mô vừa và tính phức tạp cao. Trong một số trường hợp, Excel còn xuất hiện tình trạng "Not responding" khi xử lý bảng tính có nhiều công thức mảng động. Bên cạnh yếu tố tốc độ, phương pháp xử lý bằng Python còn cho thấy tính ổn định và linh hoạt cao hơn. Cụ thể, môi trường Google Colab không gặp tình trạng treo ứng dụng như Excel và cho phép mở rộng thuật toán kiểm tra lỗi, lọc ngoại lệ hoặc thiết lập quy tắc phân loại phức tạp mà không bị giới hạn bởi giao diện. Ngoài ra, môi trường mã nguồn mở như Python còn hỗ trợ khả năng tự động hóa, kiểm thử và ghi log chi tiết, điều mà Excel chỉ có thể thực hiện khi tích hợp thêm VBA hoặc Power Query với độ phức tạp cao hơn. Kết quả cho thấy việc sử dụng Python trên Google Colab trong xử lý dữ liệu khai báo sản phẩm mang lại hiệu quả vượt trội về thời gian, tính linh hoạt và khả năng mở rộng, đặc biệt khi áp dụng cho các quy trình lặp lại hoặc tập dữ liệu có quy mô vừa và tính phức tạp cao.

3.2. Độ chính xác và tính nhất quán của dữ liệu đầu ra

Bên cạnh hiệu suất thời gian, nghiên cứu cũng đánh giá chất lượng đầu ra thông qua các chỉ số về tỷ lệ lỗi và dữ liệu không hợp lệ sau xử lý. Cụ thể, 3 loại lỗi được thống kê gồm: (i) không phân loại được tên hàng, (ii) lỗi đơn vị và quy đổi sai, và (iii) không tính toán được giá trị.

Kết quả cho thấy chất lượng đầu ra của quy trình xử lý dữ liệu bằng Google Colab (Python 3.10.2) đạt độ chính xác cao, với tổng tỷ lệ lỗi ghi nhận xấp xỉ bằng 0% trên tập dữ liệu thử nghiệm đã được chuẩn hóa gồm 10.000 dòng (Bảng 2). Cụ thể, 3 loại lỗi được quan sát đều không xảy ra

Bảng 1. *Thời gian xử lý trung bình bằng Google Colab (Python 3.10.2)*

Công đoạn xử lý	Google Colab (Python 3.10.2)
Chuẩn hóa dòng và cột	1 giây
Phân loại	300 giây
Quy đổi đơn vị và tính toán định lượng	1 giây
Kiểm tra lỗi logic và lọc dòng ngoại lệ	1 giây
Tổng thời gian xử lý	303 giây

trong quá trình xử lý gồm: không gán nhãn được các dòng tên hàng dù đã chứa từ khóa có trong bộ quy tắc, lỗi trong quá trình quy đổi đơn vị và sai số lượng, cũng như lỗi không thể tính toán giá trị do đầu vào không hợp lệ. Điều này cho thấy quy trình tiền xử lý như chuẩn hóa chuỗi văn bản, lọc ký tự đặc biệt, đồng nhất hóa đơn vị và kiểm tra tính hợp lệ của số liệu đã được thiết kế đầy đủ và hoạt động hiệu quả.

Việc sử dụng ngôn ngữ lập trình Python giúp đảm bảo tính minh bạch trong logic, khả năng kiểm soát lỗi rõ ràng và dễ dàng thiết lập các rào chắn kỹ thuật để ngăn chặn dữ liệu sai sót ngay từ bước đầu vào. Tuy nhiên, cần nhấn mạnh rằng kết quả này có được trong điều kiện dữ liệu đã được kiểm tra sơ bộ và chuẩn hóa tương đối tốt trước khi đưa vào hệ thống xử lý tự động. Trong môi trường thực tế, khi dữ liệu đầu vào đến từ nhiều nguồn khác nhau với mức độ "nhiễu" cao, việc duy trì tỷ lệ lỗi bằng 0% có thể khó đạt được nếu không có cơ chế kiểm tra và log lỗi đầy đủ. Do đó, nghiên cứu cũng đề xuất rằng trong các lần triển khai thực tế, cần tích hợp thêm các bước đánh dấu dòng nghi ngờ, phân loại lỗi và xác nhận thủ công đối với các trường hợp không rõ ràng, nhằm tăng cường độ tin cậy và tính linh hoạt của hệ thống. Kết quả về tỷ lệ lỗi cho thấy hệ thống xử lý bằng môi trường Google Colab (Python 3.10.2) không chỉ đạt hiệu quả về thời gian mà còn đảm bảo được tính chính xác và độ sạch của dữ liệu sau xử lý, là tiền đề quan trọng để thực hiện các bước phân tích tiếp theo như thống kê, mô hình hóa hoặc trực quan hóa thông tin.

Để kiểm chứng vai trò của bước chuẩn hóa trong quy trình xử lý, nghiên cứu đã tiến hành thực nghiệm đối chứng trên tập dữ liệu đầu vào ở trạng thái ban đầu, tức dữ liệu chưa trải qua bất kỳ bước xử lý sơ bộ nào như chuyển chữ in hoa, loại bỏ ký tự đặc biệt hay chuẩn hóa khoảng trắng. Trong trạng thái này, các chuỗi văn bản mô tả tên hàng và tên doanh nghiệp xuất nhập khẩu tồn tại nhiều định dạng khác nhau, bao gồm chữ hoa, chữ thường lẫn lộn, ký hiệu kỹ thuật không thống nhất, dấu câu xen lẫn và khoảng trắng thừa. Đây là đặc điểm

Bảng 2. *Tỷ lệ lỗi trung bình trong dữ liệu sau xử lý*

Loại lỗi	Google Colab (Python 3.10.2) (%)
Không gán nhãn được các dòng tên hàng dù đã chứa từ khóa có trong bộ quy tắc	0
Gán sai đơn vị hoặc quy đổi không chính xác	0
Không thể tính toán giá trị (lỗi đầu vào)	0
Tổng tỷ lệ dòng lỗi	0

Bảng 3. *Tỷ lệ lỗi của dữ liệu khi chưa được chuẩn hóa*

Tiêu chí	Tập dữ liệu chuẩn hóa	Tập dữ liệu chưa chuẩn hóa
Số dòng phân loại sai	0	610
Số dòng không phát hiện trùng lặp	0	280
Tỷ lệ lỗi tổng hợp (%)	0	8,9

phổ biến trong dữ liệu nhập khẩu thực tế, khi doanh nghiệp khai báo không theo chuẩn định dạng thống nhất. Kết quả thử nghiệm trên 10.000 dòng dữ liệu cho thấy hệ thống vẫn có thể xử lý được dữ liệu thô (raw), tuy nhiên độ chính xác giảm mạnh. Cụ thể, có 610 dòng không được phân loại chính xác theo nhóm sản phẩm tương ứng và 280 dòng bị bỏ sót do không nhận diện được trùng lặp, tổng cộng 890 dòng lỗi, tương đương với tỷ lệ lỗi 8,9%. Nguyên nhân là do việc mô tả tên hàng trong trạng thái raw không đảm bảo tính nhất quán cú pháp, khiến hệ thống rule-based không thể khớp đúng từ khóa kỹ thuật. Đồng thời, sự thiếu thống nhất trong cách viết tên doanh nghiệp (viết tắt, chữ thường, khoảng trắng thừa) khiến các dòng dữ liệu hoàn toàn trùng khớp về thông tin giao dịch nhưng khác định dạng bị hệ thống hiểu thành các bản ghi riêng biệt, không lọc trùng được. Điều này dẫn đến sai lệch thống kê về số lượng giao dịch, sản lượng và giá trị nhập khẩu.

Thử nghiệm này khẳng định rằng bước chuẩn hóa không chỉ là thao tác kỹ thuật phụ trợ mà là bước bắt buộc trong quy trình xử lý dữ liệu, ảnh hưởng trực tiếp đến khả năng phân loại chính xác, nhận diện trùng lặp và tính nhất quán của kết quả phân tích. Việc bỏ qua bước chuẩn hóa dẫn đến sai sót tích lũy trong toàn bộ quy trình, đặc biệt ở các khâu thống kê định lượng và lọc dữ liệu đầu vào. Như vậy, thử nghiệm trên tập raw không những phản ánh giới hạn của hệ thống khi thiếu chuẩn hóa, mà còn làm nổi bật vai trò thiết yếu của quy trình tiền xử lý trong việc đảm bảo hiệu quả, độ tin cậy và tính tự động hóa cao của quy trình xử lý dữ liệu xuất nhập khẩu.

3.3. Khả năng mở rộng và bảo trì hệ thống

Dưới góc độ kỹ thuật và thực tiễn triển khai, việc sử dụng ngôn ngữ lập trình Python trong xây dựng hệ thống xử lý dữ liệu xuất nhập khẩu mang lại lợi thế vượt trội về khả năng mở rộng và bảo trì so với các phương pháp truyền thống như Microsoft Excel. Trong khi Excel phụ thuộc chủ yếu vào công

thức cố định và thao tác thủ công, dễ gây lỗi lan truyền khi mở rộng quy mô hoặc cập nhật danh mục sản phẩm mới, thì hệ thống Python cho phép tổ chức kiến trúc mở, tách biệt rõ ràng giữa mã xử lý và các tệp cấu hình (ví dụ: keyword_map.json, unit_map.csv). Điều này cho phép người dùng chỉ cần cập nhật các file ánh xạ mà không phải sửa đổi mã nguồn chính, từ đó nâng cao tính linh hoạt, khả năng kiểm soát và khả năng thích ứng với các yêu cầu nghiệp vụ thay đổi liên tục. Không chỉ dừng lại ở tính tiện lợi, nghiên cứu [6] đã chỉ ra rằng quản lý cấu hình tách biệt là một trong những thực hành tốt nhất trong các dự án xử lý dữ liệu bằng Python, giúp giảm rủi ro vận hành và tăng khả năng mở rộng hệ thống khi xử lý các luồng dữ liệu phức tạp trong môi trường doanh nghiệp. Bên cạnh đó, Python cũng được ghi nhận là ngôn ngữ có khả năng tích hợp mạnh mẽ với các nền tảng đám mây như Google Colab, AWS, Azure - điều mà các nền tảng như Excel không thể đáp ứng ở quy mô lớn. Theo Ayyagiri và cộng sự [7], các công cụ xử lý song song như Dask và PySpark trong hệ sinh thái Python có thể tăng tốc độ xử lý lên đáng kể và giảm tải cho máy cá nhân, đặc biệt hữu ích khi xử lý dữ liệu nhập khẩu ngày càng mở rộng về quy mô và độ phức tạp.

Kết quả trên cho thấy quy trình xử lý dữ liệu xuất nhập khẩu sử dụng ngôn ngữ lập trình Python triển khai trên nền tảng Google Colab đã mang lại kết quả tích cực đặc biệt về hiệu suất thời gian, độ chính xác và khả năng mở rộng, bảo trì hệ thống, cho thấy tính khả thi của ứng dụng trong thực tiễn. Về hiệu suất, toàn bộ quy trình xử lý 10.000 dòng dữ liệu chỉ mất khoảng 303 giây (tương đương hơn 5 phút). Kết quả này đạt được nhờ vào việc khai thác khả năng xử lý theo vector và song song của các thư viện mã nguồn mở như Pandas, NumPy và Regex. Các bước như chuẩn hóa chuỗi văn bản, tính toán định lượng, và kiểm tra điều kiện logic đều được thực hiện gần như tức thì. Trong khi đó, Excel tuy phổ biến và trực quan thường gặp khó khăn khi áp dụng vào các tác vụ xử lý chuỗi phức tạp hoặc tập dữ liệu có quy mô vừa và tính phức tạp cao. Việc sử dụng công thức lồng nhau hoặc công thức mảng có thể gây chậm trễ rõ rệt, thậm chí dẫn đến tình trạng "treo" ứng dụng khi dữ liệu vượt ngưỡng xử lý hiệu quả (thường từ 5.000 - 10.000 dòng trở lên, tùy cấu hình máy tính và phiên bản Excel). Về độ chính xác, hệ thống xử lý bằng Python cho thấy kết quả đầu ra có độ tin cậy rất cao, với tỷ lệ lỗi sau xử lý bằng 0% trong thử nghiệm. Thành công này phản ánh hiệu quả của kiến trúc pipeline tách biệt rõ từng khâu: từ chuẩn hóa dữ liệu, kiểm tra logic đến phân loại bằng từ khóa. Nhờ khả năng kiểm soát logic lập trình chặt chẽ, việc truy vết và xử lý các tình huống ngoại lệ trở nên rõ ràng, minh bạch hơn so với Excel, nơi các thao tác chủ yếu dựa vào thủ công hoặc công thức cố định, ít khả năng kiểm tra lỗi tự động và khó truy xuất nguyên nhân khi có sai sót xảy ra. Đặc biệt, phương pháp xử lý bằng Python thể hiện ưu thế vượt trội về khả năng mở rộng và bảo trì hệ thống. Các tệp cấu hình riêng biệt (như keyword_map.json cho từ khóa hoặc unit_map.csv cho đơn vị đo) giúp hệ thống dễ dàng cập nhật theo thay đổi thực tiễn mà không cần can thiệp vào mã nguồn cốt lõi. Khi xuất hiện sản phẩm mới, biến thể tên thương mại, hay yêu cầu phân tích theo khu vực địa lý, người dùng chỉ cần cập nhật file cấu hình thay vì sửa hàng loạt công thức trong bảng tính như với Excel. Điều này giúp giảm thiểu rủi ro lỗi hệ thống, đồng thời tăng tính linh hoạt và khả năng thích ứng với dữ liệu mới trong môi trường nghiệp vụ biến động nhanh.

Kết quả đánh giá cho thấy việc triển khai quy trình xử lý dữ liệu bằng Python trên Google Colab không chỉ mang lại lợi thế rõ ràng về hiệu suất và độ chính xác, mà còn đặt nền móng cho một hệ thống bền vững, có khả năng tự động hóa, bảo trì dễ dàng và thích ứng cao với thay đổi thực tế.

4. Ứng dụng học máy trong phân loại dữ liệu xuất nhập khẩu

Khi xử lý dữ liệu khai báo hải quan, nhóm nghiên cứu đang sử dụng phương pháp phân loại tên hàng dựa trên quy tắc từ khóa (rule-based keyword matching) nhờ tính đơn giản và khả năng kiểm soát trực tiếp. Theo Sebastiani [8], các hệ thống phân loại dựa trên quy tắc (rule-based) thường phát huy hiệu quả đối với các trường hợp có quy chuẩn từ ngữ rõ ràng và ổn định, cho phép ánh xạ trực tiếp văn bản đầu vào sang các nhóm sản phẩm nhờ các quy tắc xác định sẵn. Trên thực tế, phương pháp này hoạt động hiệu quả đối với các mô tả sản phẩm phổ biến, có thể tra cứu được thông tin kỹ thuật trên các nguồn dữ liệu công khai và đã được chuẩn hóa trước đó. Tuy nhiên, theo thời gian, dữ liệu thực tế phát sinh liên tục khiến phương pháp rule-based dần bộc lộ những hạn chế cố hữu. Trung bình mỗi tháng, nhóm nghiên cứu ghi nhận khoảng 400 dòng tên hàng mới không thể phân loại theo cách truyền thống. Nguyên nhân chủ yếu đến từ việc các mô tả này không khớp hoàn toàn với tập từ khóa hiện có, sử dụng cách diễn đạt khác biệt, viết tắt theo nội bộ doanh nghiệp, hoặc không mang đủ thông tin kỹ thuật để xác định chính xác. Việc mở rộng hệ thống phân loại rule-based trong trường hợp này không chỉ mất thời gian, mà còn đòi hỏi sự can thiệp thủ công của chuyên gia, làm giảm tính linh hoạt và khả năng thích nghi của hệ thống [8].

Để khắc phục tình trạng trên, nhóm nghiên cứu đề xuất tích hợp module học máy như một công cụ bổ trợ, nhằm tự động gợi ý nhóm phân loại đối với các tên hàng chưa từng xuất hiện. Theo hướng tiếp cận này, mô hình học máy được huấn luyện trên tập dữ liệu có gán nhãn từ trước, sử dụng biểu diễn văn bản bằng kỹ thuật TF-IDF, một phương pháp vẫn được xem là hiệu quả và dễ triển khai trong các tác vụ phân loại văn bản ngắn, văn bản không có cấu trúc ngữ pháp rõ ràng [9]. Biểu diễn này cho phép mô hình học các tín hiệu ngôn ngữ tiềm ẩn từ văn bản gốc mà không cần xây dựng đặc trưng thủ công. Để thực hiện bài toán phân loại, thuật toán random forest được lựa chọn do tính ổn định và hiệu suất cao trong môi trường dữ liệu nhiều chiều và không tuyến tính - vốn là đặc trưng của tên hàng hóa nhập khẩu. Nghiên cứu của Fernández-Delgado và cộng sự [10] cho thấy random forest nằm trong nhóm mô hình hiệu quả nhất trong hơn 100 thuật toán phân loại được đánh giá trên các tập dữ liệu thực tế, đạt độ chính xác trung bình 94,1%. Khi dữ liệu khai báo hải quan thường có đặc trưng phi cấu trúc, không chuẩn hóa và thay đổi liên tục, các phương pháp phân loại thuần rule-based gặp nhiều hạn chế về khả năng bao phủ và tính thích ứng. Các nghiên cứu gần đây như Aubaid và cộng sự [11] đã chỉ ra rằng các hệ thống phân loại lai (hybrid) kết hợp giữa rule-based và học máy, chẳng hạn như random forest, decision tree hoặc các mô hình embedding, có thể cải thiện đáng kể hiệu suất phân loại, đặc biệt trong trường hợp thiếu dữ liệu nhãn hoặc không thể xác lập quy tắc thủ công rõ ràng. Bằng cách sử dụng kiến thức chuyên gia và khả năng học đặc trưng từ dữ liệu (trong AI-based), mô hình hybrid cho thấy ưu thế vượt trội về độ chính xác và khả năng mở rộng trong phân loại văn bản ngắn như mô tả hàng hóa trong khai báo hải quan. Các ứng dụng trong lĩnh vực hải quan như Singh và cộng sự [12] chỉ ra tiềm năng của học máy trong việc xử lý dữ liệu khai báo bất cấu trúc và phát hiện bất thường.

Khi hệ thống phân loại dựa trên từ khóa truyền thống dần bộc lộ những giới hạn về khả năng bao phủ và tính mở rộng, việc tích hợp mô hình học máy như 1 module bổ trợ mang lại triển vọng cải thiện đáng kể hiệu quả vận hành. Phương pháp tiếp cận dựa trên học máy không nhằm thay thế hoàn toàn quy trình hiện tại, mà đóng vai trò hỗ trợ việc gợi ý phân loại đối với các dòng tên hàng chưa từng xuất hiện hoặc khó quy chuẩn hóa bằng luật cố định. Các phần tiếp theo sẽ trình bày chi tiết quy trình xây dựng, huấn luyện và đánh giá mô hình phân loại này, với trọng tâm là tính khả thi và khả năng ứng dụng thực tế trong môi trường xử lý dữ liệu.

4.1. Dữ liệu và tiền xử lý

Nghiên cứu sử dụng tập dữ liệu gồm 8.236 dòng tên hàng nhập khẩu, trong đó mỗi dòng chứa 2 trường thông tin chính: (i) TEN_HANG là mô tả sản phẩm được ghi dưới dạng chuỗi văn bản tự do trong tờ khai hải quan và (ii) PHAN_LOAI là nhãn phân loại tương ứng được gán thủ công theo các nhóm tiêu chuẩn như PP, PE, PS... Đây là những dữ liệu phản ánh thực tế trong lĩnh vực nhập khẩu nhựa tại Việt Nam, với tính đa dạng cao về cách viết, tên thương mại, từ viết tắt và ký hiệu kỹ thuật. Do mục tiêu là đánh giá khả năng phân loại tự động từ văn bản gốc, nghiên cứu không áp dụng bất kỳ đặc trưng kỹ thuật nào được trích xuất thủ công. Toàn bộ thông tin đầu vào được xử lý trực tiếp từ chuỗi văn bản TEN_HANG. Quy trình tiền xử lý được thiết kế nhằm loại bỏ nhiễu và chuẩn hóa định dạng văn bản gồm: chuyển toàn bộ ký tự về dạng chữ thường, loại bỏ ký tự đặc biệt, chuẩn hóa khoảng trắng và chuyển đổi văn bản sang dạng biểu diễn số học bằng TF-IDF (term frequency-inverse document frequency). TF-IDF được áp dụng với ngữ cảnh unigram và bigram, giới hạn tối đa 500 đặc trưng xuất hiện phổ biến nhất nhằm duy trì tính tổng quát, tránh hiện tượng quá khớp và giảm chi phí tính toán. Cách biểu diễn này giúp mã hóa được các tín hiệu định danh quan trọng trong mô tả sản phẩm mà không cần hiểu ngữ nghĩa một cách đầy đủ, qua đó tạo điều kiện thuận lợi cho mô hình học máy xử lý và khai thác giá trị từ dữ liệu văn bản.

4.2. Mô hình hóa và huấn luyện

Trên cơ sở dữ liệu đã được chuẩn hóa, nghiên cứu triển khai nhiệm vụ phân loại tên hàng bằng mô hình random forest, một thuật toán thuộc nhóm học máy tổ hợp (ensemble learning), nổi bật với khả năng xử lý dữ liệu phi tuyến, kháng nhiễu tốt và đặc biệt phù hợp với các bài toán phân loại đa lớp. Đây là lựa chọn phù hợp khi tên hàng nhập khẩu có biểu hiện đa dạng, nhiều biến thể ngôn ngữ và không tuân theo quy tắc ngữ pháp cụ thể. Quy trình mô hình hóa được tổ chức thành 1 pipeline 2 giai đoạn: giai đoạn đầu thực hiện trích xuất đặc trưng văn bản bằng TF-IDF (sử dụng lại kết quả tiền xử lý từ phần trước) và giai đoạn tiếp theo là huấn luyện mô hình phân loại bằng random forest với 200 cây quyết định. Đặc biệt, để xử lý vấn đề mất cân bằng dữ liệu khi một số nhóm sản phẩm như PP_TERT hay PP_RANDOM có số lượng mẫu huấn luyện rất nhỏ, tham số class_weight='balanced' được tích hợp trong mô hình. Tham số này giúp điều chỉnh trọng số của từng lớp theo tỷ lệ xuất hiện trong tập huấn luyện, từ đó tăng cường khả năng nhận diện đối với các lớp hiếm

mà không làm ảnh hưởng đáng kể đến hiệu suất chung của mô hình. Việc tích hợp toàn bộ quy trình vào pipeline không chỉ đảm bảo tính nhất quán và khả năng tái sử dụng trong triển khai thực tế mà còn cho phép linh hoạt điều chỉnh các tham số kỹ thuật trong quá trình tối ưu hóa. Tập dữ liệu được chia theo phương pháp phân tầng (stratified sampling) nhằm đảm bảo phân bố đồng đều các nhãn trong cả hai tập huấn luyện và kiểm tra. Cụ thể, 5.765 dòng dữ liệu được sử dụng để huấn luyện mô hình, trong khi 2.471 dòng còn lại được giữ lại để đánh giá hiệu quả phân loại. Cách chia này giúp giảm thiểu sai lệch thống kê và cải thiện độ tin cậy của kết quả, đặc biệt quan trọng khi một số nhóm sản phẩm có số lượng quan sát rất thấp.

4.3. Đánh giá hiệu quả mô hình

Hiệu quả của mô hình random forest được kiểm định trên tập kiểm tra gồm 2.471 dòng dữ liệu tên hàng hóa nhập khẩu đã được gán nhãn, sử dụng các chỉ số phổ biến trong bài toán phân loại đa lớp như độ chính xác (precision), độ bao phủ (recall), điểm F1 (F1-score) và số lượng mẫu (support) trên từng nhóm nhãn cụ thể (Bảng 4). Kết quả cho thấy mô hình đạt độ chính xác tổng thể (accuracy) ở mức 81%, với các chỉ số trung bình gia quyền của precision (0,84), recall (0,76) và F1-score (0,79) (Bảng 5). Đây là kết quả khả quan khi dữ liệu thực tế có phân bố không đều giữa các lớp, nhiều nhóm nhãn có số lượng mẫu rất nhỏ và nội dung mô tả tên hàng không theo cấu trúc ngôn ngữ chuẩn hóa.

Bảng 4. *Hiệu suất phân loại theo từng nhóm nhãn trên tập kiểm tra*

Nhãn (Label)	Độ chính xác (Precision)	Độ bao phủ (Recall)	Điểm F1 (F1-Score)	Hỗ trợ (Support)
PE	1	0,5	0,67	10
PP	0,86	0,88	0,87	1100
PP_Block_Copolymer	1	0,97	0,98	59
PP_Compound	1	0,75	0,86	4
PP_Copolymer	0,77	0,85	0,81	327
PP_Homo	0,84	0,78	0,81	441
PP_Impact	0,82	0,68	0,74	301
PP_Other	0,54	0,64	0,58	140
PP_Random	0,58	0,57	0,57	86
PP_Tert	1	1	1	3

Bảng 5. *Chỉ số trung bình đánh giá tổng thể mô hình phân loại*

Chỉ số (Metrics)	Giá trị (Score)
Độ chính xác (Accuracy)	0,81
Trung bình cộng độ chính xác (Macro avg precision)	0,84
Trung bình cộng độ bao phủ (Macro avg recall)	0,76
Trung bình cộng điểm F1 (Macro avg F1-score)	0,79
Trung bình gia quyền độ chính xác (Weighted avg precision)	0,82
Trung bình gia quyền độ bao phủ (Weighted avg recall)	0,81
Trung bình gia quyền điểm F1 (Weighted avg F1-score)	0,81

Xét theo từng nhóm nhãn, mô hình thể hiện năng lực phân loại tốt đối với các nhóm phổ biến có quy mô dữ liệu lớn. Nhóm PP với 1.100 mẫu đạt F1-score 0,87, phản ánh khả năng học hiệu quả các đặc trưng từ vựng đại diện cho dòng nhựa polypropylene nói chung. Tương tự, các nhóm PP_Homo (441 mẫu) và PP_Copolymer (327 mẫu) cũng đạt F1-score lần lượt là 0,81, cho thấy mô hình có thể nhận diện tốt các biến thể kỹ thuật khác nhau trong cùng một họ sản phẩm. Đối với nhóm PP_Block_Copolymer, F1-score đạt tới 0,98 với recall lên đến 0,97, thể hiện độ chính xác cao khi phân biệt lớp có từ khóa kỹ thuật rõ ràng. Tuy nhiên, đối với các nhóm hiếm như PE (10 mẫu), PP_Compound (4 mẫu) và PP_TERT (3 mẫu), mô hình vẫn gặp khó khăn nhất định. Mặc dù precision đạt tuyệt đối (1), recall lại thấp hoặc không ổn định (lần lượt là 0,5, 0,75 và 1), dẫn đến F1-score chỉ dao động từ 0,67 - 1. Điều này phản ánh thực trạng thường thấy trong các bài toán mất cân bằng lớp, khi mô hình thiếu dữ liệu huấn luyện để học được các đặc trưng chuyên biệt của nhóm, dẫn đến hiện tượng bỏ sót (false negative).

Ngoài ra, các nhóm có ranh giới mô tả không rõ ràng như PP_Other (140 mẫu) và PP_Random (86 mẫu) chỉ đạt F1-score lần lượt là 0,58 và 0,57. Điều này cho thấy mô hình gặp khó khăn trong việc phân biệt các tên hàng chứa nội dung không đặc trưng hoặc dễ gây nhầm lẫn với các nhóm khác có từ vựng tương đồng. Đặc biệt, chỉ số macro average recall đạt mức 0,7669, thấp hơn rõ rệt so với macro precision (0,849) và macro F1-score (0,795), phản ánh sự chênh lệch trong hiệu suất mô hình giữa các lớp phổ biến và các lớp hiếm. Tuy nhiên, so với mô hình không áp dụng trọng số cho từng lớp, việc sử dụng tham số class_weight='balanced' trong huấn luyện đã giúp cải thiện đáng kể khả năng nhận diện các nhóm nhãn có số lượng mẫu hạn chế. Các chỉ số macro recall và F1-score đều tăng lên, cho thấy mô hình đã giảm được độ thiên lệch trong phân loại và trở nên công bằng hơn giữa các lớp.

Khi dữ liệu nhập khẩu thực tế liên tục phát sinh các mô tả mới, việc mô hình hoạt động

kém trên các nhóm ít phổ biến là một vấn đề cần được khắc phục. Các giải pháp kỹ thuật phù hợp có thể gồm: (i) áp dụng kỹ thuật tăng cường dữ liệu như smote hoặc data augmentation for text để mở rộng tập huấn luyện cho các lớp nhỏ; (ii) kết hợp các mô hình phân loại tiên tiến hơn như: XGBoost, LightGBM, hoặc Transformer-based models (BERT, PhoBERT) nếu bài toán yêu cầu nhận diện ngữ cảnh sâu hơn trong các mô tả sản phẩm dài. Mô hình được xây dựng bằng biểu diễn TF-IDF, phương pháp đơn giản, hiệu quả và dễ triển khai, kết quả đạt được vẫn có tính khả thi và ổn định. Điều này cho thấy trong bài toán có cấu trúc dữ liệu ngắn, không đầy đủ ngữ pháp như tên hàng trong tờ khai hải quan, TF-IDF vẫn là lựa chọn phù hợp. Bên cạnh đó, random forest được lựa chọn vì khả năng xử lý tốt dữ liệu nhiều chiều, tính ổn định cao, không quá nhạy với nhiễu và phù hợp với bài toán không yêu cầu tốc độ suy luận quá nhanh như trong thời gian thực. Kết quả này phù hợp với nghiên cứu của Fernández-Delgado và cộng sự [10], nghiên cứu chỉ ra rằng random forest là thuật toán phân loại có độ chính xác cao trong số hơn 100 thuật toán được đánh giá trên nhiều tập dữ liệu khác nhau.

Mô hình học máy không nhằm thay thế hoàn toàn hệ thống phân loại dựa trên từ khóa truyền thống, mà đóng vai trò như module hỗ trợ, cung cấp gợi ý phân loại trong các trường hợp mà phương pháp rule-based không thể áp dụng, ví dụ như với các mô tả mới, viết tắt, hoặc thiếu thông tin định danh rõ ràng... Điều này giúp nâng cao hiệu quả xử lý tự động, giảm bớt công việc thủ công trong quá trình rà soát và gán nhãn cho sản phẩm mới.

5. Kết luận

Nghiên cứu trình bày quy trình ứng dụng ngôn ngữ lập trình Python và học máy vào việc xử lý và phân loại dữ liệu xuất nhập khẩu, với mục tiêu cải thiện hiệu suất, độ chính xác và khả năng thích ứng của hệ thống xử lý dữ liệu khai báo hải quan. Thông qua việc triển khai chuỗi thao tác kỹ thuật bao gồm chuẩn hóa tên hàng, ánh xạ địa lý, quy đổi đơn vị, tính toán định lượng và phân loại sản phẩm, quy trình tự động hóa được xây dựng bằng Python đã chứng minh ưu thế rõ rệt so với phương pháp truyền thống sử dụng Excel trong các nghiên cứu quốc tế.

Kết quả thực nghiệm cho thấy việc sử dụng công cụ mới đã rút ngắn đáng kể thời gian xử lý, giảm thiểu sai sót, tăng tính nhất quán và dễ dàng mở rộng khi có dữ liệu phát sinh theo thời gian. Khi số lượng và biến thể tên hàng gia tăng nhanh, việc tích hợp mô hình học máy, cụ thể là thuật toán random forest kết hợp với biểu diễn văn bản TF-IDF, đã mở ra hướng tiếp cận khả thi nhằm tự động hóa việc gợi ý phân loại cho các dòng tên hàng mới. Mô hình đạt độ chính xác tổng thể 83% trên tập kiểm tra, với hiệu suất đặc biệt tốt ở các nhóm sản phẩm phổ biến. Tuy nhiên, các nhóm có số lượng mẫu hạn chế vẫn gặp khó khăn trong việc nhận diện, cần cải tiến thông qua việc mở rộng tập dữ liệu huấn luyện, áp dụng kỹ thuật cân bằng lớp hoặc thử nghiệm các mô hình tiên tiến hơn như LightGBM hoặc BERT.

Kết quả của nghiên cứu này khẳng định tính khả thi và hiệu quả của việc ứng dụng ngôn ngữ lập trình Python và học máy vào xử lý dữ liệu xuất nhập khẩu, cung cấp khung kỹ thuật có thể tái sử dụng và mở rộng cho các lĩnh vực tương tự như thương mại, logistics, và phân tích dữ liệu phi cấu trúc trong quản lý doanh nghiệp.

Tài liệu tham khảo

[1] Kelvin Kelvin, Wahidin Wahab, and Meirista Wulandari, "Computer resource utilization analysis for microsoft excel and python in data processing", *Engineering, Mathematics and Computer Science Journal* (EMACS), Volume 6, Issue 2, pp. 137 - 142, 2024. DOI: 10.21512/emacsjournal.v6i2.11736.

[2] Mohamed Fakhry Mansour, Tarek Aly, and Mervat Gheith, "Python based end user computing framework to empowering excel efficiency", *International Journal for Research in Applied Science and Engineering Technology*, Volume 12, Issue 4, pp. 2719 - 2729, 2024. DOI: 10.22214/ijraset.2024.60097.

[3] Raymond R. Panko and Richard P. Halverson Jr., "An experiment in collaborative spreadsheet development", *Journal of the Association for Information Systems*, Volume 2, No. 1, pp. 1 - 31, 2001. DOI: 10.17705/1jais.00016.

[4] Raymond R. Panko, "Thinking is bad: Implications of human error research for spreadsheet research and practice", *European Spreadsheet Risk Interest Group*, 2007. DOI: 10.48550/arXiv.0801.3114.

[5] Alexandros Nikolaos Ziogas, Timo Schneider, Tal Ben-Nun, Alexandru Calotoiu, Tiziano De Matteis, Johannes de Fine Licht, Luca Lavarini, and Torsten Hoefler, "Productivity, portability, performance", *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2021*. DOI: 10.1145/3458817.3476176.

[6] Diyyala Sravani, Jonnala Rohith Reddy, Pilla Sri Viswas, N.M. Jyothi, and Potru Chandukiran, "Python security in devOps: Best practices for secure

coding, configuration management, and continuous testing and monitoring", *4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 6 - 8 July 2023*. DOI: 10.1109/icesc57686.2023.10193128.

[7] Aravind Ayyagiri, Arpit Jain, and Om Goel, "Utilizing Python for scalable data processing in cloud environments", *Darpan International Research Analysis*, Volume 12, Issue 2, pp. 183 - 198, 2024. DOI: 10.36676/dira.v12.i2.78.

[8] Fabrizio Sebastiani, "Machine learning in automated text categorization", *ACM Computing Surveys*, Volume 34, Issue 1, pp. 1 - 47, 2002. DOI: 10.1145/505282.505283.

[9] ChengXiang Zhai and Sean Massung, *Text data management and analysis: A practical introduction to information retrieval and text mining*. Association for Computing Machinery and Morgan & Claypool, 2016. DOI: 10.1145/2915031.

[10] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim, "Do we need hundreds of classifiers to solve real world classification problems?", *Journal of Machine Learning Research*, Volume 15, pp. 3133 - 3181, 2014.

[11] Asmaa M. Aubaid, Alok Mishra, and Atul Mishra, "Machine learning and rule-based embedding techniques for classifying text documents", *International Journal of Systems Assurance Engineering and Management*, Volume 15, Issue 12, pp. 5637 -5652, 2024. DOI: 10.1007/s13198-024-02555-w.

[12] Karandeep Singh, Yu-Che Tsai, Cheng-Te Li, Meeyoung Cha, and Shou-De Lin, "GraphFC: Customs fraud detection with label scarcity", *32nd ACM International Conference on Information and Knowledge Management, 2023*. DOI: 10.1145/3583780.3614690.

APPLICATION OF PYTHON AND MACHINE LEARNING IN PROCESSING AND CLASSIFYING IMPORT-EXPORT DATA

Do Hong Hanh[1], Doan Tien Quyet[1], Doan Trong Sinh[2], Nguyen Bang Linh[1]
[1]Vietnam Petroleum Institute (VPI)
[2]Vietnam National Industry - Energy Group (PVN)
Email: hanhdh@vpi.pvn.vn

Summary

Vietnam's import-export data is increasing substantially in both scale and complexity, creating significant challenges in standardizing and classifying customs declaration information. This study proposes an automated data-processing pipeline implemented in the Python programming language, with the objective of enhancing efficiency and ensuring greater consistency in the analysis of customs information. The input dataset comprises more than 10,000 real-world import-export records, which are processed through a structured sequence of technical steps, including product name normalization, unit conversion, computation of quantitative indicators, and keyword-based product group labeling.

The experimental results demonstrate that this processing pipeline operates effectively on medium-scale, high-complexity datasets, while considerably improving classification accuracy and ensuring uniformity across product categories. Based on these findings, the authors propose integrating machine learning models as a supplementary tool to enhance generalization capabilities and adaptability to exceptional cases - particularly relevant in a trade environment where product names are increasingly diverse, unstandardized, and continuously evolving.

Key words: Import, export, Python, machine learning, data normalization, product classification, TF-IDF, random forest, automation.