

MINI PROJECT: AIR GUARD – DỰ BÁO PM2.5 VÀ CẢNH BÁO AQI THEO TRẠM

ThS. Lê Thị Thùy Trang

2026-01-03

1 AIR GUARD: Dự báo chất lượng không khí dựa trên dữ liệu

1.1 Giới thiệu dự án

Dự báo chất lượng không khí (AQI, đặc biệt dựa trên nồng độ PM2.5) là một bài toán quan trọng trong bối cảnh đô thị hóa và biến đổi khí hậu. Các mô hình học máy có giám sát truyền thống thường yêu cầu nhiều dữ liệu có nhãn để đạt độ chính xác cao. Tuy nhiên, trong thực tế, việc thu thập dữ liệu có nhãn (ví dụ, phân loại mức độ ô nhiễm không khí cho từng giờ) có thể khó khăn và tốn kém, dẫn đến tình trạng thiếu nhãn trên một lượng lớn dữ liệu.

Học bán giám sát xuất hiện như một hướng tiếp cận giúp tận dụng cả dữ liệu có nhãn lẫn chưa có nhãn để cải thiện mô hình. Trong mini project này, chúng ta tập trung vào hai phương pháp bán giám sát phổ biến là Self-Training (tự huấn luyện) và Co-Training (đồng huấn luyện):

1. Self-Training (Tự huấn luyện): Bắt đầu với một mô hình huấn luyện trên tập dữ liệu nhỏ có nhãn, sau đó dùng mô hình này dự đoán nhãn cho dữ liệu chưa gán nhãn. Các dự đoán tự tin nhất (độ xác tín cao hơn ngưỡng) sẽ được coi như "nhãn giả" và bổ sung vào tập huấn luyện. Mô hình được huấn luyện lại với tập dữ liệu mở rộng này. Quá trình lặp lại nhiều vòng nhằm dần dần cải thiện mô hình với sự hỗ trợ của chính dự đoán của nó.
2. Co-Training (Đồng huấn luyện): Sử dụng hai mô hình khác nhau huấn luyện song song trên hai nhóm đặc trưng (feature views) khác nhau của cùng dữ liệu. Mỗi mô hình được huấn luyện trên tập nhãn thật ban đầu sẽ dự đoán nhãn cho dữ liệu chưa có nhãn. Mô hình thứ nhất chọn ra một số mẫu mà nó tự tin nhất để gán nhãn giả và cung cấp cho mô hình thứ hai huấn luyện bổ sung, và ngược lại. Qua mỗi vòng, hai mô hình trao đổi kiến thức cho nhau thông qua nhãn giả, giúp cải thiện kết quả nếu hai nhóm đặc trưng mang thông tin bổ trợ lẫn nhau.

Việc ứng dụng hai phương pháp trên trong bài toán dự đoán AQI giúp chúng ta trả lời các câu hỏi quan trọng: Liệu sử dụng thêm dữ liệu không có nhãn (ví dụ các khoảng thời gian chưa được phân loại AQI) có thể nâng cao độ chính xác phân loại so với chỉ dùng dữ liệu ít ỏi có nhãn hay không? Thuật toán self-training và co-training có ưu nhược điểm gì trong bối cảnh dữ liệu chuỗi thời gian? Kết quả dự án sẽ không chỉ dừng ở việc "chạy mô hình đúng", mà còn hướng tới việc phân tích, diễn giải hiệu quả mô hình, hiểu rõ hành vi mô hình trong những tình huống cụ thể, cũng như cân nhắc khả năng ứng dụng thực tế.

1.2 Mục tiêu

Sau khi thực hiện xong dự án, sinh viên có thể:

1. Hiểu quy trình kết hợp các thuật toán phân lớp, chuỗi thời gian, bán giám sát.
2. Áp dụng self-training và co-training để cải thiện mô hình dự báo AQI khi dữ liệu nhãn khan hiếm, thông qua việc sử dụng nhãn giả sinh ra từ mô hình có giám sát.
3. So sánh, đánh giá hiệu năng giữa mô hình giám sát truyền thống và mô hình bán giám sát (self-training, co-training) bằng các chỉ số như độ chính xác, F1-score đa lớp (macro-F1) và ma trận nhầm lẫn.

4. Phân tích ảnh hưởng của các tham số trong thuật toán bán giám sát (tỷ lệ dữ liệu không nhãn, ngữ cảnh tự tin, số vòng lặp, v.v.) đến chất lượng mô hình.

1.3 Pipeline

Dự án được thực hiện theo các bước chính sau:

1. Tiền xử lý và khai phá luật (tái sử dụng Lab 4): Trước tiên cần làm sạch dữ liệu (loại bỏ hoặc xử lý giá trị thiếu, ngoại lai) và tái định dạng cột thời gian. Sau đó chọn mốc thời gian cắt (cutoff) vào ngày 2017-01-01 để chia dữ liệu thành hai phần: tập huấn luyện+unlabeled (trước 2017) và tập kiểm tra (test) (năm 2017 trở đi). Điều này đảm bảo mô phỏng đúng bối cảnh dự báo tương lai, tránh rò rỉ dữ liệu tương lai vào huấn luyện.
2. Gắn nhãn phân loại AQI: Chuyển đổi giá trị PM2.5 thành các nhãn phân loại chất lượng không khí. Sử dụng tiêu chuẩn AQI để gắn nhãn cho mỗi mẫu giờ theo các mức. Đây sẽ là biến mục tiêu (target) cho bài toán phân loại:
 - Good (Tốt);
 - Moderate (Trung bình);
 - Unhealthy for Sensitive Groups (Không lành mạnh cho nhóm nhạy cảm);
 - Unhealthy (Không lành mạnh);
 - Very Unhealthy (Rất không lành mạnh)
 - Hazardous (Nguy hại).
3. Tách tập có nhãn vs không nhãn: Xác định một phần dữ liệu huấn luyện có nhãn ban đầu và phần còn lại xem như chưa có nhãn. Có thể chọn theo hai hướng:
 - Theo thời gian – ví dụ, lấy dữ liệu của một khoảng thời gian đầu (hoặc của một vài trạm) làm tập có nhãn, những thời gian sau coi như chưa có nhãn. Cách này phản ánh kịch bản thực tế: ta chỉ có nhãn lịch sử đến một thời điểm, sau đó mô hình phải tự học với dữ liệu mới.
 - Ngẫu nhiên có kiểm soát – chọn ngẫu nhiên X% mẫu trong tập trước 2017 để giữ lại nhãn (đảm bảo mỗi lớp AQI đều có một vài mẫu), số còn lại bỏ nhãn. Cách này giúp đảm bảo đa dạng lớp trong tập gốc.

Trong mini project này, dữ liệu đang được tách theo phương pháp ngẫu nhiên có kiểm soát để giữ nguyên nguyên tắc chống leak: test ($>= 2017$) không bị đựng vào. Đồng thời, mô phỏng đúng tình huống nhãn hiếm nhưng phân phối thời gian trong train không bị lệch quá mạnh.

4. Feature Engineering: Xây dựng các đặc trưng đầu vào cho mô hình phân loại tương tự bước chuẩn bị dữ liệu hồi quy trong lab 4.
5. Huấn luyện mô hình Supervised baseline: Sử dụng tập dữ liệu có nhãn ban đầu (nhỏ) để huấn luyện một mô hình phân loại baseline (sử dụng mô hình HistGradientBoostingClassifier với các siêu tham số cố định để đảm bảo công bằng giữa các thử nghiệm). Đánh giá mô hình baseline trên validation để có điểm so sánh ban đầu (các chỉ số như Accuracy, F1-score macro, và phân tích sơ bộ độ chính xác trên từng lớp). Kết quả này được lưu làm mốc (benchmark).
6. Huấn luyện mô hình Self-training.
7. Huấn luyện mô hình Co-training.
8. Đánh giá kết quả.

1.4 Danh sách các module trong project

Dự án được tổ chức với các lớp (module) tương tự **Lab 4** để đảm bảo tính rõ ràng và tái sử dụng.

Chúng ta tái sử dụng các lớp cũ cho những bước chung và bổ sung lớp mới cho học bán giám sát. Xem chi tiết danh sách các module tại `readme.md` trên repo Github.

1.5 Chuẩn bị môi trường thực hành

Kích hoạt môi trường ảo

```
conda activate beijing_env
```

Toàn bộ các thư viện cần thiết cho dự án được liệt kê trong file `requirements.txt`. Vì vậy, sau khi kích hoạt môi trường, cài đặt các thư viện cần thiết bằng câu lệnh:

```
pip install -r requirements.txt
```

2 Huấn luyện mô hình học bán giám sát

2.1 Huấn luyện mô hình Self-training

Các bước triển khai mô hình Self-training có thể được mô tả một cách súc tích trong hình 1.

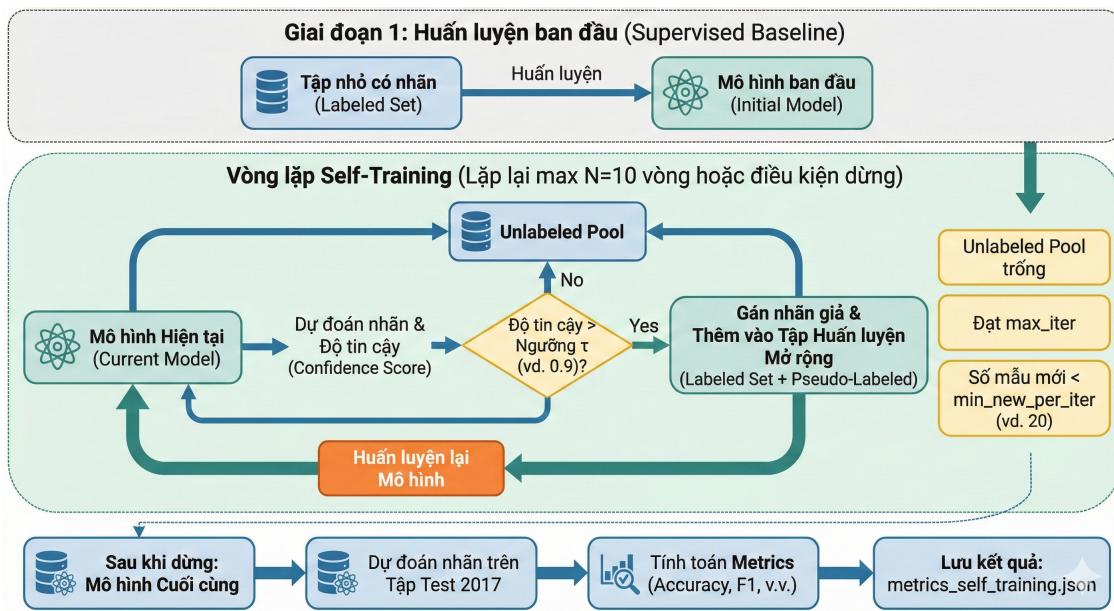


Figure 1: Self-training pipeline

- Đầu tiên, huấn luyện mô hình ban đầu trên tập nhỏ có nhãn (Labeled Set) tương tự như việc huấn luyện mô hình Supervised baseline.
- Sau đó, lặp lại các vòng: dùng mô hình hiện tại để dự đoán nhãn cho Unlabeled Pool. Với mỗi mẫu chưa nhãn, lấy xác suất dự đoán cao nhất (confidence). Chọn các mẫu có độ tin cậy cao hơn ngưỡng τ ($\tau = 0.9$) để gán nhãn giả theo dự đoán của mô hình. Thêm những mẫu gán nhãn này vào tập huấn luyện, đồng thời loại chúng khỏi Unlabeled Pool. Huấn luyện lại mô hình với tập huấn luyện mở rộng đó.
- Tiếp tục lặp cho đến khi đạt tối đa N vòng lặp hoặc Unlabeled Pool trống hoặc số mẫu mới thêm vào quá ít (dưới một ngưỡng min_batch). Trong triển khai có thể giới hạn, (ví dụ: $max_iter = 10$ và $min_new_per_iter = 20$) để tránh vòng lặp vô tận.

4. Ở mỗi vòng, lưu lại số lượng mẫu mới được gán nhãn, tổng số mẫu chưa nhãn còn lại, và độ chính xác (hoặc F1) trên tập validation. Sinh viên cần trực quan hóa kết quả self-training qua các vòng, ví dụ vẽ biểu đồ **số lượng mẫu gán nhãn vs. vòng lặp** và **độ chính xác validation vs. vòng lặp**. Điều này giúp thấy rõ mô hình đã học thêm được bao nhiêu từ dữ liệu không nhãn và chất lượng có tăng hay không.
5. Sau khi dừng, sử dụng mô hình cuối cùng (sau các vòng self-training) để dự đoán nhãn cho tập test 2017 và tính toán các metrics trên test giống bước huấn luyện mô hình Supervised baseline. Lưu các kết quả này (ví dụ trong file `metrics_self_training.json`).

2.2 Huấn luyện mô hình Co-training

Các bước triển khai mô hình Co-training có thể được mô tả một cách súc tích trong hình 2.

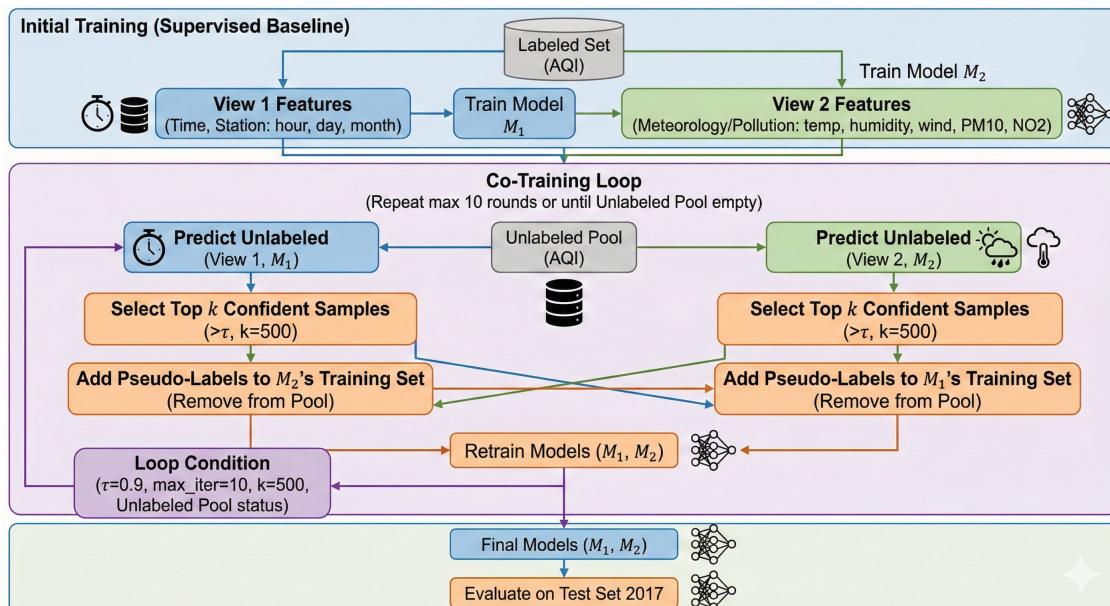


Figure 2: Co-training pipeline

Triển khai thuật toán co-training với hai mô hình khác nhau để tận dụng hai “view” đặc trưng:

1. Chọn view và mô hình: Phân chia bộ đặc trưng thành hai nhóm dựa trên tính chất khác nhau, ví dụ: (View 1) các đặc trưng thời gian + tự tương quan (giờ, ngày, tháng, lag PM2.5) và (View 2) các đặc trưng thời tiết, môi trường (nhiệt độ, áp suất, độ ẩm, gió...). Hai nhóm đặc trưng này cung cấp hai góc nhìn khác nhau về dữ liệu. Chọn hai mô hình phân loại phù hợp (có thể cùng thuật toán nhưng mỗi mô hình chỉ dùng một nhóm đặc trưng). Cả hai đều được huấn luyện ban đầu bằng tập dữ liệu nhỏ có nhãn.
2. Vòng lặp huấn luyện đồng thời: Tương tự self-training nhưng với hai mô hình hỗ trợ lẫn nhau: ở mỗi vòng, để Mô hình A dự đoán nhãn cho các mẫu chưa nhãn (dựa trên đặc trưng view 1). Chọn ra một số mẫu mà Mô hình A tự tin nhất (vượt ngưỡng xác suất $\tau = 0.9$). Những mẫu này cùng nhãn dự đoán sẽ được thêm vào tập huấn luyện của Mô hình B (tức Mô hình B sẽ học thêm những mẫu này, với đặc trưng view 2 của chúng, gán nhãn theo dự đoán của A). Tương tự, dùng Mô hình B dự đoán và chọn các mẫu tự tin để gán nhãn cho Mô hình A. Lưu ý loại bỏ các mẫu được gán nhãn ra khỏi pool unlabeled sau mỗi vòng để tránh lặp lại. Sau khi trao đổi nhãn xong, cập nhật cả hai mô hình bằng cách huấn luyện tiếp với các mẫu mới được bổ sung. Lặp lại quá trình cho các vòng tiếp theo.
3. Để tránh đưa quá nhiều nhãn nhiễu, có thể giới hạn mỗi mô hình chỉ thêm tối đa N mẫu mới mỗi vòng (ví dụ $N = 100\text{--}500$ mẫu mỗi phía) và chỉ thêm nếu vượt ngưỡng tin cậy. Dừng lại khi đã đạt số vòng tối đa (ví dụ 10

vòng) hoặc không còn đủ mẫu mới để thêm (dưới một ngưỡng min).

4. Ghi nhận và trực quan hóa: Tương tự self-training, lưu lại lịch sử: mỗi vòng thêm bao nhiêu mẫu từ mỗi mô hình, tổng số mẫu unlabeled còn lại, và hiệu suất trên validation của từng mô hình (có thể theo dõi trung bình hoặc theo mô hình chính nếu chọn một mô hình làm chính). Sinh viên cần vẽ biểu đồ hoặc bảng so sánh độ chính xác trên validation qua các vòng cho cả hai mô hình, xem xu hướng đồng huấn luyện ra sao (hai mô hình có cải thiện, có hội tụ về chất lượng hay không).
5. Kết quả mô hình co-training: Sau khi dừng, có thể chọn một trong hai mô hình (hoặc trung bình phiếu của cả hai) làm mô hình cuối. Đem mô hình này dự đoán trên tập test 2017, tính các chỉ số (Accuracy, F1, v.v.) và lưu kết quả dự báo.

3 Hướng dẫn thực hiện hoạt động FIT-DNU CONQUER

3.1 Mục tiêu của hoạt động

Hoạt động FIT-DNU CONQUER được thiết kế nhằm giúp sinh viên:

- Hiểu sâu bản chất của các thuật toán đã học bằng cách áp dụng chúng trên dữ liệu thực tế và quan sát hiệu quả. Cụ thể ở đây là self-training và co-training trong bối cảnh dữ liệu chuỗi thời gian.
- Rèn luyện kỹ năng diễn giải kết quả: Sinh viên phải trình bày được kết quả mô hình bằng ngôn ngữ đơn giản, trực quan, tránh chỉ liệt kê mã code hay bảng số liệu. Thay vào đó, tập trung giải thích mô hình của mình đã hoạt động ra sao, học được gì từ dữ liệu.
- Phát triển kỹ năng thuyết trình và thuyết phục: thông qua việc chuẩn bị blog báo cáo và slide trình bày kết quả, sinh viên học cách chọn lọc và nhấn mạnh những điểm quan trọng, trình bày mạch lạc và trả lời câu hỏi phản biện.
- Tư duy phân tích theo góc nhìn ứng dụng: Không chỉ dừng ở kết quả mô hình tốt hay không, sinh viên cần suy nghĩ như một Data Scientist thực thụ: từ kết quả mô hình để rút ra insight cho bài toán (ở đây là xu hướng ô nhiễm, thời điểm nguy cơ cao, v.v.), cũng như đề xuất được hành động hoặc cải thiện tiếp theo (cần thêm dữ liệu gì, cảnh báo ra sao,...).

Sinh viên cần xem xét dự án như một nhiệm vụ Data Scientist thực thụ: không chỉ “chạy đúng thuật toán”, mà phải chuyển đổi dữ liệu thành tri thức hữu ích.

3.2 Yêu cầu

Trong Mini Project này, các nhóm sẽ sử dụng repo cơ sở `air_guard` để xây dựng một pipeline dự báo chất lượng không khí, thực hiện các yêu cầu như sau:

1. Huấn luyện thuật toán Self-training. Sử dụng mô hình baseline làm mô hình ban đầu, thực hiện self-training trên tập dữ liệu không nhãn.
 - Thiết lập thông số: thay đổi ngưỡng τ , so sánh kết quả và chọn ngưỡng phù hợp.
 - Lưu lại kết quả qua các vòng và trình bày bảng hoặc biểu đồ (khuyến khích sử dụng) thể hiện diễn biến self-training. Từ đó nhận xét: Lúc đầu mô hình tự tin gán nhãn được nhiều không? Xu hướng tăng/giảm: có thể vòng 1 thêm rất nhiều (mô hình tự tin quá mức hoặc gặp nhiều mẫu đẽ), nhưng sau vài vòng có thể hết mẫu đẽ hoặc mô hình thận trọng hơn. Nếu thấy độ chính xác validation giảm ở vòng nào, thảo luận nguyên nhân (có thể do mô hình đã thêm nhãn sai và học theo chúng, gây giảm hiệu năng tạm thời). Quyết định dừng ở vòng bao nhiêu?
 - Hiệu năng của mô hình: Báo cáo các chỉ số trên tập test và so sánh với baseline. Ít nhất phải có Accuracy và F1-score macro. Nhận xét: self-training cải thiện/giảm so với baseline bao nhiêu? Cần chỉ rõ những lớp nào được hưởng lợi?

2. Huấn luyện thuật toán Co-training. Thực hiện co-training với hai mô hình và hai view đặc trưng. Yêu cầu:

- Mô tả rõ hai nhóm đặc trưng mà nhóm sử dụng cho hai mô hình.
 - Thiết lập self-labeling cho mỗi mô hình: Chọn ngưỡng τ giống hoặc khác nhau cho hai mô hình (đơn giản thì dùng cùng $\tau = 0.9$). Quyết định mỗi vòng mỗi mô hình sẽ thêm tối đa bao nhiêu mẫu cho mô hình kia
 - Theo dõi diễn biến: Tương tự self-training, yêu cầu có bảng/biểu đồ cho co-training: mỗi vòng, mô hình A thêm bao nhiêu mẫu, mô hình B thêm bao nhiêu; độ chính xác validation của từng mô hình sau vòng đó. Đặc biệt, xem hai mô hình có cải thiện song song không: lý tưởng là cả hai cùng tăng dần và sát nhau. Nếu co-training thất bại (độ chính xác không tăng hoặc giảm), cố gắng lý giải: có phải do hai view không thực sự độc lập (dẫn đến hai mô hình mắc lỗi giống nhau và cung cấp lỗi cho nhau)? Hay do một mô hình quá mạnh, mô hình kia yếu nên nhận trao đổi không tốt?
 - Kết quả mô hình co-training: Đánh giá trên tập test, báo cáo các chỉ số tương tự trên. Xem mô hình nào (A hay B hoặc ensemble) được chọn làm cuối. So sánh *co-training vs. self-training vs. baseline*. Yêu cầu tối thiểu: nêu rõ co-training có tốt hơn self-training không trong trường hợp của nhóm. Nếu không tốt bằng, phân tích lý do có thể: ví dụ do dữ liệu không đủ tách thành hai view hiệu quả, self-training đã dùng tối đa dữ liệu rồi; hay tham số co-training chưa phù hợp. Nếu có tốt hơn, giải thích: hai view bổ sung thông tin giúp mô hình học tự tin hơn những mẫu mà self-training có thể bỏ qua, v.v.
3. So sánh các cấu hình/tham số: thực hiện ít nhất một phép thử so sánh khi thay đổi tham số so với thiết lập gốc. Bắt buộc: thử nghiệm với một giá trị τ khác cho self-training hoặc co-training và quan sát sự khác biệt. Ngoài ra, có thể thử thêm một trong các cấu hình:
- Thay đổi kích thước tập có nhãn ban đầu (ví dụ dùng nhiều hơn một chút dữ liệu có nhãn xem có cải thiện đáng kể không).
 - Thử mô hình/thuật toán khác: ví dụ, thử chuyển sang RandomForest xem self-training có cải thiện khác không.
 - Thủ tách view khác đi (nếu ban đầu tách theo loại đặc trưng, có thể thử tách theo tập trạm: mô hình A học dữ liệu trạm này, B học trạm khác, rồi trao đổi – cách này ít phô biến nhưng sáng tạo nếu có lý do).
 - Những thử nghiệm mở rộng này nhằm giúp sinh viên hiểu rõ hơn tác động của các yếu tố trong thuật toán. Đưa kết quả và nhận xét ngắn gọn.

4. Xây dựng dashboard trực quan (Streamlit)

3.3 Khuyến khích nâng cấp dự án

Ngoài phần bắt buộc, các nhóm được khuyến khích chọn thêm các hướng mở rộng để nâng chất lượng bài làm (Nhóm nào tham vọng tổng kết 10 thì phải làm thôi):

1. Nhóm có thể thử áp dụng Label Propagation/Label Spreading (thuật toán truyền nhãn trên đồ thị có sẵn trong scikit-learn) trên bộ dữ liệu này. Thuật toán đó coi mỗi mẫu (có nhãn và chưa nhãn) là nút trên đồ thị và lan truyền nhãn dựa trên cấu trúc khoảng cách của dữ liệu. Thủ so sánh kết quả với self-training/co-training. Việc này giúp mở rộng hiểu biết về các tiếp cận bán giám sát khác nhau.
2. Dynamic Threshold theo lớp (FlexMatch-lite, tăng hiệu quả lớp hiếm): Thay vì dùng một ngưỡng chung τ , ta dùng ngưỡng theo lớp τ_c để giảm thiên lệch về lớp phổ biến, tăng recall cho lớp AQI nặng và cải thiện Macro-F1. Ngưỡng này sẽ thay đổi dựa trên trạng thái học của mô hình: $\tau_c(t) = \text{AvgConf}_c(t) \cdot \tau_{base}$ (Trong đó, $\text{AvgConf}_c(t)$ là độ tin cậy trung bình của mô hình đối với lớp c trong các vòng lặp trước). Có thể tích hợp Focal Loss vào quá trình tái huấn luyện để giảm trọng số của các mẫu dễ (easy examples) và tập trung vào các mẫu khó (hard examples) 16: $\mathcal{L}_{Focal} = -\alpha(1 - p_t)^\gamma \log(p_t)$ Khi p_t tiến tới 1 (mô hình dự đoán đúng và tự tin), nhân tử $(1 - p_t)^\gamma$ sẽ tiến tới 0, triệt tiêu loss của mẫu đó. Điều này giúp mô hình không bị chi phối bởi hàng triệu giao dịch bình thường dễ đoán, mà tập trung sửa lỗi trên các giao dịch gian lận hiếm gặp.

3.4 Kết quả kỳ vọng

Mỗi nhóm cần hoàn thành:

- Blog/Report (link Notion/GitHub).
- Slide trình bày.

3.5 Gợi ý cho phần trình bày tại lớp

1. Giới thiệu bài toán và mục tiêu nhóm.
2. Trình bày kết quả trọng tâm (không kê lê, không giải thích code).
3. Diễn giải các biểu đồ.
4. Kết luận và đề xuất hành động.