ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO TRIỂN KHAI KỸ THUẬT HỌC MÁY

MALLORN Astronomical Classification Challenge

Lớp học phần :	2526I_INT3405E_4
Môn học :	Học máy
Giảng viên :	TS. Lê Đức Trọng
Nhóm:	21

Hà Nội, 2025

Thành viên

Mã Sinh Viên	Họ và Tên	Công việc	Phần trăm đóng góp
22025512	Nguyễn Gia Bảo	- Xây dựng mô hình, tham gia huấn luyện tại Kaggle - Làm silde thuyết trình - Làm báo cáo xây dựng mô hình	33.3%
22025520	Trần Khánh Duy	- Xây dựng mô hình, tham gia huấn luyện tại Kaggle - Làm silde thuyết trình - Thuyết trình bài báo cáo	33.3%
22025525	Phạm Quang Anh	- Xây dựng mô hình, tham gia huấn luyện tại Kaggle - Làm silde thuyết trình - Làm báo cáo xây dựng mô hình	33.3%

Mục lục

1. Tóm tắt *6*

2. Giới thiệu *6*

3. Phân tích dữ liệu khám phá *7*

3.1. Mô tả dữ liệu và thống kê cơ bản 7

3.2. Phân bố số lượng quan sát theo đối tượng 7

3.3. Phân bố quan sát theo dải bước sóng (filter) 9

3.4. Phân tích phân bố độ sáng và sai số quan sát (Flux, Flux_err) 10

3.4.1. Phân bố giá trị (Flux) 10

3.4.2. Phân bố sai số (Flux_err) 11

3.4.3. Quan hệ giữa Flux và Flux_err 12

3.5. Phân bố thời gian quan sát (MJD) 14

3.6. Phân bố tỷ lệ theo nhãn target, loại phổ và đặc trưng quan sát 15

3.7. Một số lightcurve tiêu biểu 22

3.8. Tổng hợp các đặc trưng quan trọng từ EDA 24

4. Tiền xử lý dữ liệu *26*

4.1. Tách và kết hợp theo split. 26

4.2. Trích xuất đặc trưng 26

4.3. Thêm đặc trưng meta 27

4.4. Xử lý dữ liệu thiếu và chuẩn hoá 27

4.6. Kết quả đầu ra 28

5. Mô hình *28*

5.1. XGBoost 29

5.2. LightGBM 30

5.3. Ưu điểm của các mô hình 31

6. Phương pháp huấn luyện *33*

6.1. Chiến lược chia dữ liệu huấn luyện và đánh giá. 33

6.2. Hàm mất mát 33

6.3. Optimizer và tham số huấn luyện. 34
6.4. Chiến lược chống overfitting 34
6.5. Quy trình huấn luyện chính 35
6.6. Dự đoán và đánh giá sau huấn luyện 37
7. Cải tiến mô hình 38
7.1. Phân tích cải tiến EDA và trích xuất đặc trưng 38
7.2. Tối ưu hoá mô hình và chiến lược huấn luyện 39
8. Đánh giá Hiệu suất mô hình 41
8.1. Hiệu suất của XGBoost và LightBGM 41
8.2. Nguyên nhân giới hạn điểm 42
8.3. Hướng cải tiến đề xuất 43

Hình Ảnh

Hình 1: Histogram phân bố số lần quan sát trên mỗi đối tượng..... 8
Hình 2:Số lượng điểm quan sát trên từng kênh filter..... 9
Hình 3: Phân bố giá trị Flux 11
Hình 4: Phân bố sai số Flux_err..... 12
Hình 5: Biểu đồ scatter giữa flux và flux_err 13
Hình 6: Phân bố thời gian quan sát (MJD)..... 14
Hình 7: Phân bố tỷ lệ đối tượng theo nhãn target (0 hoặc 1) trong tập huấn luyện 15
Hình 8: Số lượng đối tượng phân theo loại phổ (SpecType) trong tập huấn luyện 16
Hình 9: Tỷ lệ đối tượng nhãn TDE (target =.) theo từng SpecType trên tập huấn luyện..... 17
Hình 10: Phân bố độ dịch chuyển đỏ (redshift Z) và thông số EBV của đối tượng trong tập huấn luyện so với tập kiểm tra. 18
Hình 11: Số điểm quan sát theo từng bộ lọc (filter) cho toàn bộ bộ dữ liệu (cột màu tượng trưng cho mỗi filter). 19
Hình 12: Phân bố số lượng bộ lọc (filter) mà mỗi đối tượng được quan sát. 19
Hình 13: Phân bố giá trị khoảng thời gian trung vị giữa hai lần quan sát liên tiếp (median cadence) cho mỗi object. 20
Hình 14: Phân bố giá trị khoảng gián đoạn dài nhất giữa hai lần quan sát liên tiếp (max_gap) cho mỗi object. 21
Hình 15: Phân bố mẫu các giá trị flux (đã chuẩn hóa) thu được trên các filter khác nhau. 22
Hình 16: Lightcurve của đối tượng "cair_rudh_Sindarin" trên 6 kênh màu..... 23
Hình 17: Lightcurve của đối tượng “daen_muil_dundadan”..... 23
Hình 18: : Lightcurve của đối tượng “harad_bragol_ablad” 24
Hình 19: Kiến trúc mô hình XGBoost. 30
Hình 20: Kiến trúc mô hình LightGBM 31
Hình 21: Sơ đồ cơ chế hoạt động phương pháp huấn luyện mô hình 36
Hình 22: Kiến trúc phương pháp huấn luyện ở 2 phiên bản điểm 0.5706 và 0.6120..... 40
Hình 23: Biểu đồ AUC-PR cho XGBoost..... 41
Hình 24: Biểu đồ AUC-PR cho LightGBM 41
Hình 25: Kết quả F1-score 41

1. Tóm tắt

Sự ra đời của khảo sát LSST tại Vera C. Rubin Observatory sẽ tạo ra khối lượng lớn các sự kiện thiên văn biến thiên, đặt ra thách thức trong việc phân loại và lựa chọn các đối tượng quan trọng để theo dõi chuyên sâu. Trong số đó, Tidal Disruption Events (TDEs) là các hiện tượng hiếm nhưng có giá trị khoa học cao, tuy nhiên số lượng đã được xác nhận hiện nay còn rất hạn chế.

Báo cáo này trình bày một phương pháp học máy nhằm tham gia cuộc thi MALLORN Astronomical Classification Challenge, với mục tiêu phân loại TDE dựa trên dữ liệu đường cong ánh sáng đa băng mô phỏng từ LSST. Pipeline đề xuất bao gồm trích xuất đặc trưng thủ công mang ý nghĩa vật lý kết hợp với đặc trưng chuỗi thời gian từ tsfresh, sử dụng hai mô hình Gradient Boosting là XGBoost và LightGBM. Mô hình được huấn luyện bằng cross-validation, tối ưu theo F1-score và AUC-PR, đồng thời áp dụng ensemble để nâng cao độ ổn định và hiệu quả phân loại trong điều kiện dữ liệu mất cân bằng.

2. Giới thiệu

Khảo sát LSST được kỳ vọng sẽ mở ra một kỷ nguyên mới cho thiên văn học biến thiên, với số lượng các sự kiện được phát hiện tăng mạnh so với các khảo sát trước đây. Tuy nhiên, khả năng phân loại chính xác các đối tượng dựa trên dữ liệu phổ là không khả thi ở quy mô này, đòi hỏi các phương pháp phân loại quang học tự động dựa trên đường cong ánh sáng.

Tidal Disruption Events (TDEs) là các hiện tượng xảy ra khi một ngôi sao bị lực thủy triều của hố đen siêu nặng xé toạc, cung cấp thông tin quan trọng về môi trường gần hố đen. Do tính chất hiếm và dữ liệu mất cân bằng, việc nhận diện TDE từ lightcurve là một bài toán thách thức.

Cuộc thi MALLORN Astronomical Classification Challenge cung cấp dữ liệu mô phỏng LSST dựa trên quan sát từ ZTF, nhằm đánh giá khả năng áp dụng học máy cho bài toán này. Trong báo cáo này, sử dụng pipeline học máy kết hợp trích xuất đặc trưng vật lý và thống kê từ lightcurve với các mô hình Gradient Boosting (XGBoost, LightGBM), đồng thời áp dụng ensemble để cải thiện hiệu quả phân loại và phù hợp với các ràng buộc thực tế của thiên văn học quan sát.

3. Phân tích dữ liệu khám phá

3.1. Mô tả dữ liệu và thống kê cơ bản

Tập dữ liệu	Số dòng	Nội dung
lightcurves.csv	1,624,509	Tất cả các điểm quan sát (flux, flux_er, thời gian)
train_log.csv	3,043	Danh sách đối tượng dùng để huấn luyện (có nhãn)
test_log.csv	7,135	Danh sách đối tượng dùng để dự đoán (chưa có nhãn)

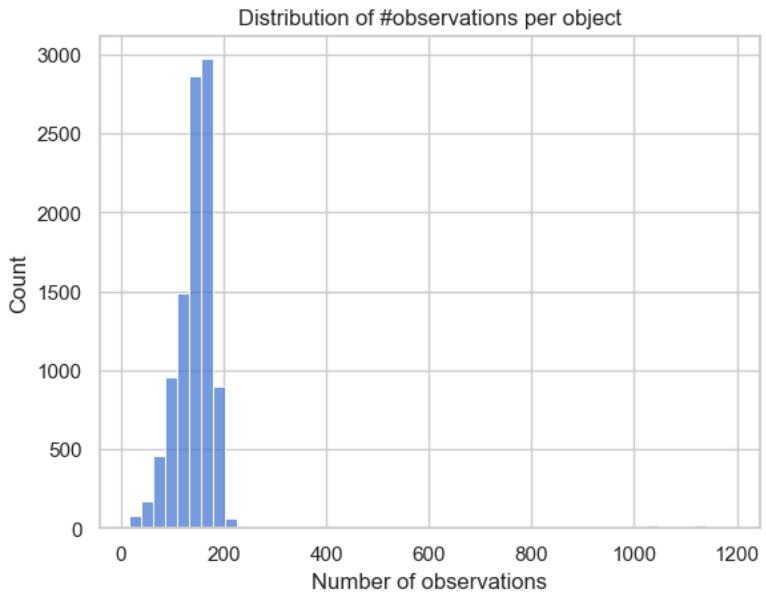
Mỗi đối tượng thiên văn được quan sát tại nhiều thời điểm khác nhau trên tối đa 6 kênh màu (filter). Tập train có 3043 đối tượng trong khi tập test có 7135 đối tượng – sự chênh lệch này đòi hỏi mô hình có khả năng tổng quát hóa tốt, tránh phụ thuộc quá nhiều vào các đối tượng cụ thể trong tập huấn luyện.

Kiểm tra sơ bộ cho thấy không có giá trị thiếu (NA). Tuy nhiên, cảnh báo về giá trị không hợp lệ (*invalid value*) cho thấy có thể tồn tại những giá trị ngoại lệ lớn hoặc vô hạn (Inf) trong dữ liệu – đây có thể là kết quả của nhiễu đo đạc hoặc lỗi thiết bị. Những giá trị này cần được xử lý để tránh ảnh hưởng xấu đến quá trình phân tích và mô hình.

3.2. Phân bố số lượng quan sát theo đối tượng

Phân tích đầu tiên xem xét số lượng điểm quan sát trên mỗi đối tượng (số lần quan sát mỗi object). Hình 1 dưới đây là biểu đồ histogram cho thấy đa số đối tượng có khoảng

130–210 lần quan sát. Số lần quan sát trung bình khoảng ~170. Bên cạnh đó, có một số ít đối tượng là ngoại lệ với số lần quan sát rất lớn (trên 400, cao nhất khoảng 1200).



Hình 1: Histogram phân bố số lần quan sát trên mỗi đối tượng

Điều này có nghĩa phần lớn các lightcurve có dữ liệu khá dày, đủ để mô tả hình dạng biến thiên theo thời gian – thuận lợi cho việc nhận dạng các sự kiện thiên văn đặc biệt như TDE. Tuy nhiên, cũng có vài đối tượng chỉ có dưới 20 quan sát – với số lượng điểm ít ỏi như vậy thì hầu như không thể phát hiện được sự kiện TDE (nếu có) do dữ liệu quá thưa.

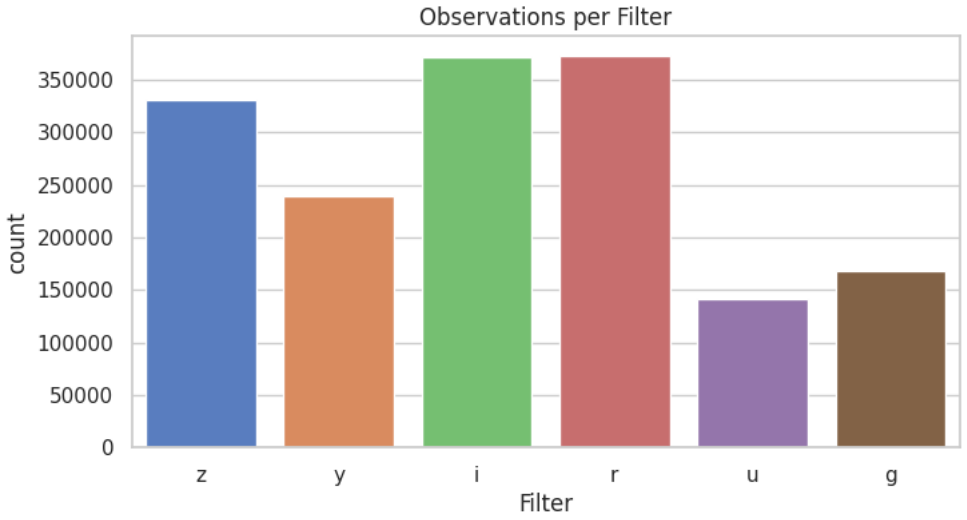
Từ phân tích này, có thể đề xuất một số đặc trưng liên quan đến lượng dữ liệu của mỗi object, ví dụ:

- Số lần quan sát (*n_obs*) của mỗi đối tượng
- Mật độ quan sát (*obs_density* = n_obs / tổng thời gian quan sát)
- Khoảng trống lớn nhất và trung bình giữa các lần quan sát (*max_gap*, *mean_gap*)

Những đặc trưng trên phản ánh mức độ đầy đủ và dày đặc của dữ liệu cho mỗi object. Trực giác cho thấy chúng ảnh hưởng lớn đến khả năng phát hiện và phân biệt hình dạng của sự kiện TDE (vốn đòi hỏi có độ phủ quan sát tốt theo thời gian).

3.3. Phân bố quan sát theo dải bước sóng (filter)

Mỗi quan sát được thực hiện trên một trong 6 dải bước sóng (kênh màu) ký hiệu là g, r, i, z, u, y. Hình 2 là biểu đồ phân bố số lượng điểm quan sát trên từng filter. Ta thấy hai filter r và i có số lượng quan sát cao nhất – mỗi kênh khoảng 370 nghìn điểm (chiếm ưu thế rõ rệt). Kế đó là filter z (~330k) và y (~240k). Trong khi đó, hai kênh g và u có số lượng quan sát thấp nhất, khoảng 150k điểm mỗi kênh.



Hình 2:Số lượng điểm quan sát trên từng kênh filter

Từ góc độ vật lý, các sự kiện TDE thường biểu hiện mạnh ở vùng quang phổ đỏ (tương ứng các kênh r, i, z), do đó việc có nhiều dữ liệu ở các kênh này là thuận lợi cho bài toán. Ngược lại, các kênh xanh (g, u) có ít dữ liệu nhất – điều này cho thấy nếu chúng ta áp dụng chuẩn hóa toàn cục (global normalization) hoặc mô hình hóa mà không tách biệt theo kênh, thì dữ liệu từ các kênh ít sẽ dễ bị nhiễu và không đóng góp nhiều thông tin.

Vì vậy, một hướng tiếp cận là trích xuất các đặc trưng theo từng dải màu thay vì chỉ dùng giá trị gộp chung. Chẳng hạn, có thể tính các thống kê đặc trưng trên từng filter quan trọng (như r, i, z) hoặc so sánh giữa các filter. Một số đặc trưng gợi ý bao gồm:

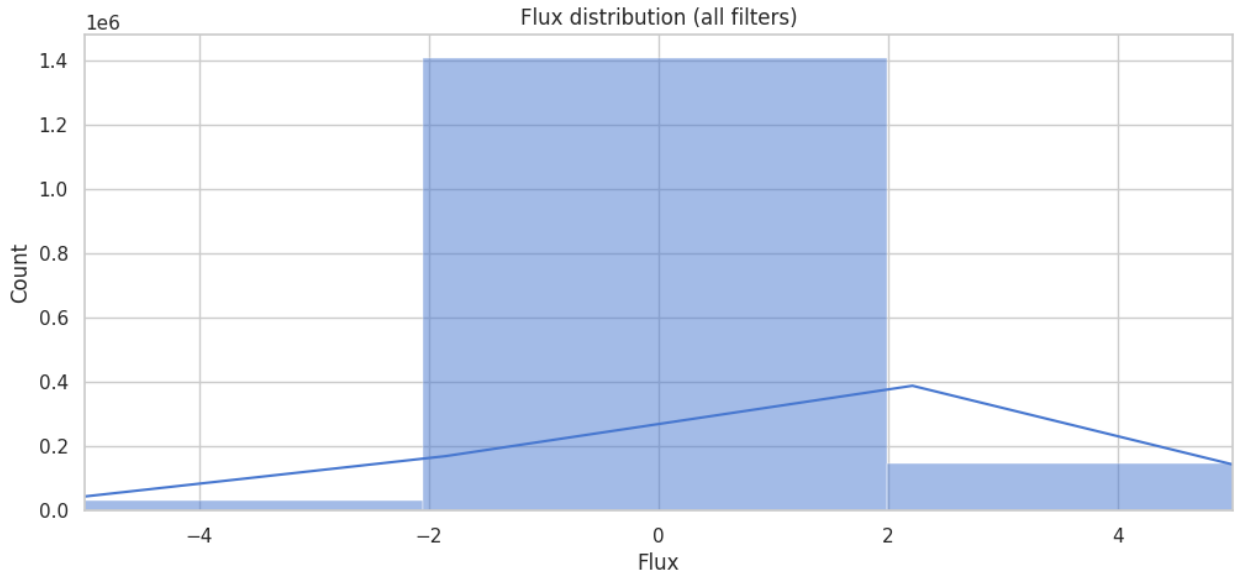
- Giá trị trung bình, lớn nhất của *flux* trên từng filter (ví dụ: mean_flux_r, max_flux_r, …)
- Độ chênh lệch giữa các kênh về biên độ đỉnh (ví dụ: tỷ số flux đỉnh r/i, hay chênh lệch thời gian đạt đỉnh giữa r và i)
- Tỷ số tín hiệu trên nhiễu (SNR) riêng cho các kênh đỏ (r, i, z)

3.4. Phân tích phân bố độ sáng và sai số quan sát (Flux, Flux_err)

3.4.1. Phân bố giá trị (Flux)

Giá trị độ sáng (flux) của các quan sát có phân bố xấp xỉ theo phân phối chuẩn với trung bình ~0 và độ lệch chuẩn ~1. Tuy nhiên, phân phối này có đuôi kéo dài về cả phía âm và dương (tới khoảng ±5 đơn vị chuẩn hóa), thể hiện có một số điểm ngoại lai với độ sáng rất cao hoặc rất thấp. (Những ngoại lệ này có thể do nhiễu hoặc lỗi khi quan sát – ví dụ: điều kiện thời tiết kém hoặc trục trặc thiết bị khiến phép đo không chính xác).

Đáng chú ý, có khá nhiều giá trị flux âm. Điều này xảy ra do quá trình trừ nền (*background subtraction*) trong đo lường thiên văn: khi cường độ tín hiệu thực tế rất thấp, phép đo sau khi loại bỏ nhiễu nền có thể cho kết quả âm. Ngoài ra, phân bố tổng thể của flux không đồng nhất giữa các kênh quan sát – mỗi kênh (filter) có độ nhạy và khoảng giá trị khác nhau. Vì vậy, việc chuẩn hóa flux theo từng filter riêng biệt có thể cần thiết để giảm sự khác biệt về thang đo giữa các kênh.



Hình 3: Phân bố giá trị Flux

Từ những quan sát này, có thể rút ra một số đặc trưng liên quan đến phân bố flux của mỗi object, chẳng hạn:

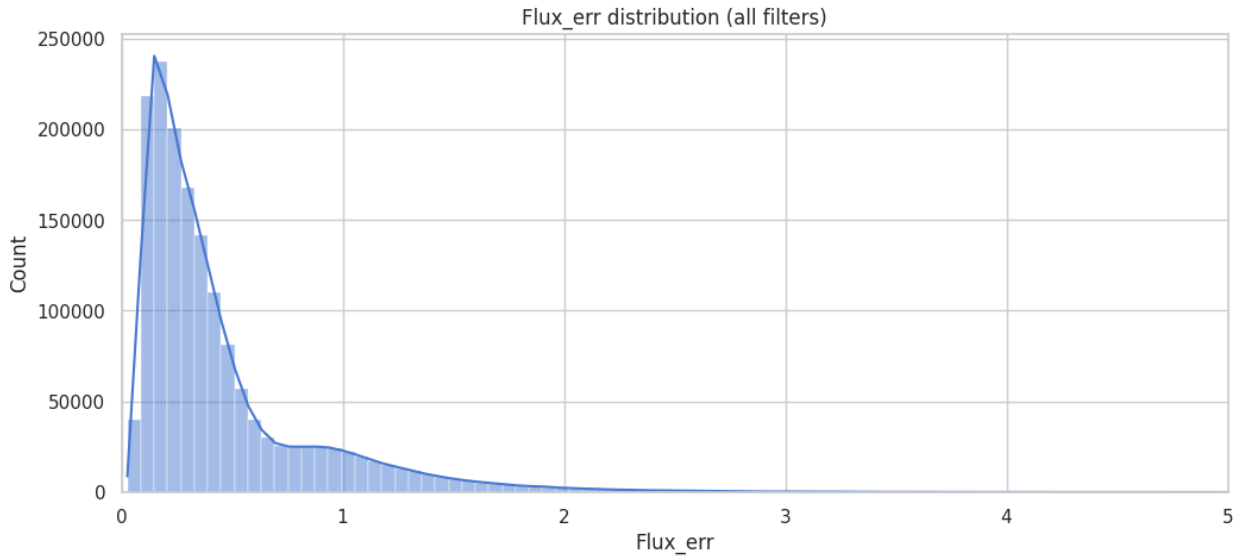
- Cắt ngưỡng *flux* (flux_clipped): giới hạn giá trị flux trong khoảng [-5, 5] để giảm ảnh hưởng của các ngoại lệ quá lớn
- Chuẩn hóa flux theo từng filter (flux_scaled_by_filter) thay vì trên toàn bộ dataset
- Đo lường độ lệch và độ nhọn của phân bố flux (ví dụ: *skewness*, *kurtosis* của flux) cho từng object

3.4.2. Phân bố sai số (Flux_err)

Sai số quan sát (flux_err) có phân bố lệch phải rất mạnh. Phần lớn các điểm có sai số nhỏ (< 0.6), nhưng cũng có một số không nhỏ trường hợp sai số lên tới ~3–4. Điều này có nghĩa nhiều phép đo có độ chính xác cao (sai số thấp), trong khi một số phép đo rất kém chính xác (sai số cao bất thường).

Các sự kiện TDE thường tạo ra đường cong ánh sáng tương đối mượt mà và nhất quán, do đó các quan sát của TDE kỳ vọng có sai số nhỏ và ổn định hơn so với những nguồn biến thiên phức tạp (ví dụ các nhân chuẩn – AGN – với biến động hỗn loạn). Sự hiện diện của những điểm dữ liệu có sai số quá cao sẽ gây khó khăn cho việc phát hiện

đỉnh (peak) của đường cong ánh sáng, bởi nhiễu lớn có thể che lấp tín hiệu. Vì vậy, một chiến lược là loại bỏ hoặc giảm trọng số những điểm có sai số cao nhất (ví dụ loại bỏ top 1% giá trị flux_err) khi xây dựng mô hình hoặc tính toán đặc trưng.



Hình 4: Phân bố sai số Flux_err

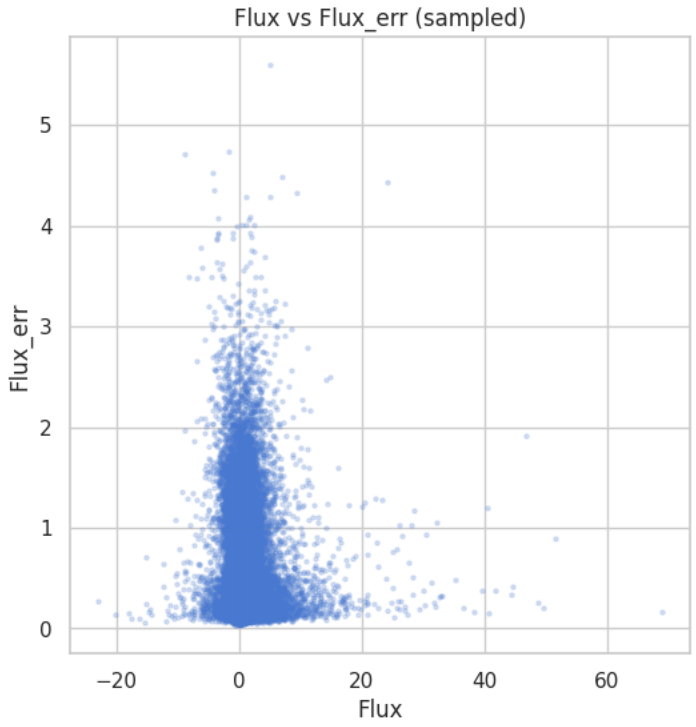
Những đặc trưng có thể tính dựa trên phân tích sai số bao gồm:

- Sai số trung bình và độ biến thiên của sai số trên mỗi object (mean_err, std_err)
- Tỷ số tín hiệu trên nhiễu (signal-to-noise ratio – SNR), ví dụ SNR trung bình, SNR lớn nhất (snr_peak), v.v.
- Chỉ số nhiễu (noise_index): định nghĩa là độ lệch chuẩn của flux chia cho sai số trung bình (*std_flux / mean_err*) của object

3.4.3. Quan hệ giữa Flux và Flux_err

Để hiểu rõ hơn mối quan hệ giữa độ sáng và sai số đo, ta xem xét biểu đồ phân tán (scatter plot) giữa flux và flux_err cho tất cả các điểm dữ liệu. Kết quả cho thấy một xu hướng: khi flux tăng cao thì sai số cũng tăng theo. Điều này phù hợp với thực tế là các quan sát cường độ mạnh thường cũng đi kèm bất định cao hơn ở mức độ tuyệt đối.

Phần lớn (~90%) các điểm dữ liệu tập trung trong vùng |flux| < 2 (đơn vị chuẩn hóa) – tức đa phần các quan sát có độ sáng không quá lớn so với nhiễu nền. Các điểm này tạo thành một đám mây khá dày đặc quanh gốc tọa độ (flux ~0, flux_err thấp). Một số điểm nằm xa (flux rất cao kèm sai số cao) chính là những trường hợp ngoại lai đã đề cập ở trên.



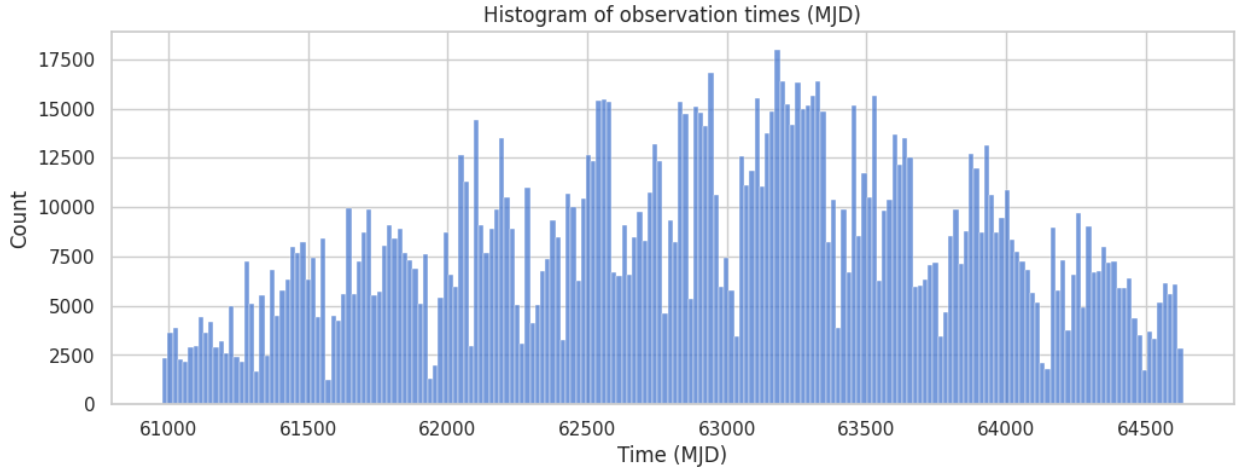
Hình 5: Biểu đồ scatter giữa flux và flux_err

Hình dạng phân bố của đám mây điểm khá "mờ" và không có cấu trúc phân tách rõ ràng giữa các nhóm đối tượng. Điều này gợi ý rằng việc phân loại trực tiếp dựa trên các điểm dữ liệu thô (raw time-series) sẽ gặp nhiều khó khăn – rất khó xác định đối tượng nào là TDE hay không chỉ dựa vào cặp giá trị (flux, flux_err) của từng lần quan sát. Do đó, chúng ta cần chuyển sang sử dụng các đặc trưng tổng hợp (summary features) được trích xuất từ toàn bộ lightcurve của mỗi đối tượng để phân loại hiệu quả hơn.

3.5. Phân bố thời gian quan sát (MJD)

Các quan sát trong bộ dữ liệu được thực hiện trong khoảng thời gian vài năm. Histogram phân bố thời điểm quan sát (theo đơn vị ngày Julius hiệu chỉnh – MJD) cho thấy các quan sát trải dài từ MJD ~61000 đến ~64500. Phân bố theo thời gian không đều: có những giai đoạn tập trung rất nhiều quan sát (các đỉnh trong biểu đồ), xen kẽ những khoảng thời gian dài hầu như không có quan sát nào. Điều này phản ánh lịch trình quan sát gián đoạn (ví dụ: do những thời kỳ không quan sát được hoặc chiến dịch quan sát theo đợt).

Một hệ quả là khung thời gian quan sát của mỗi đối tượng có thể khác nhau đáng kể. Một số đối tượng chỉ được theo dõi trong khoảng thời gian ngắn, trong khi có những đối tượng được quan sát trải dài nhiều năm. Thực tế, phân bố độ dài khoảng thời gian quan sát của mỗi object nằm chủ yếu trong khoảng ~1500–3000 ngày (tương đương ~4–8 năm). Sự khác biệt về độ dài chuỗi quan sát (duration) này ảnh hưởng trực tiếp đến khả năng nhận dạng hình dạng đường cong: những object có chuỗi quan sát dài và liên tục sẽ thuận lợi hơn trong việc phát hiện TDE so với object chỉ được quan sát trong thời gian ngắn.

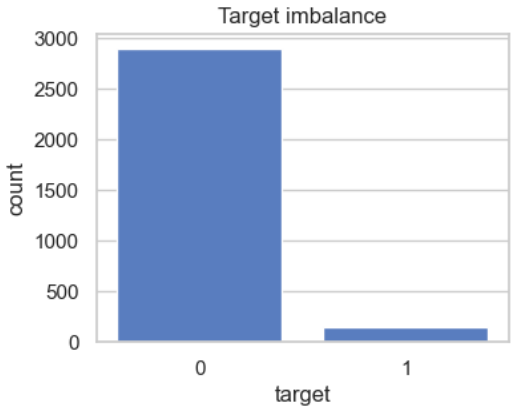


Hình 6: Phân bố thời gian quan sát (MJD)

Từ phân tích này, ta có thể trích xuất các đặc trưng liên quan đến thời gian cho mỗi đối tượng, ví dụ:

- Tổng thời gian được quan sát (*duration_days* = MJD_max – MJD_min của object)
- Tần suất trung bình (*cadence* = duration_days / n_obs, thể hiện khoảng cách trung bình giữa các lần quan sát)
- Khoảng trống dài nhất không có quan sát (*longest_gap* giữa hai lần quan sát liên tiếp)
- Mật độ quan sát (*observation_density* = số quan sát mỗi đơn vị thời gian, là nghịch đảo của cadence)

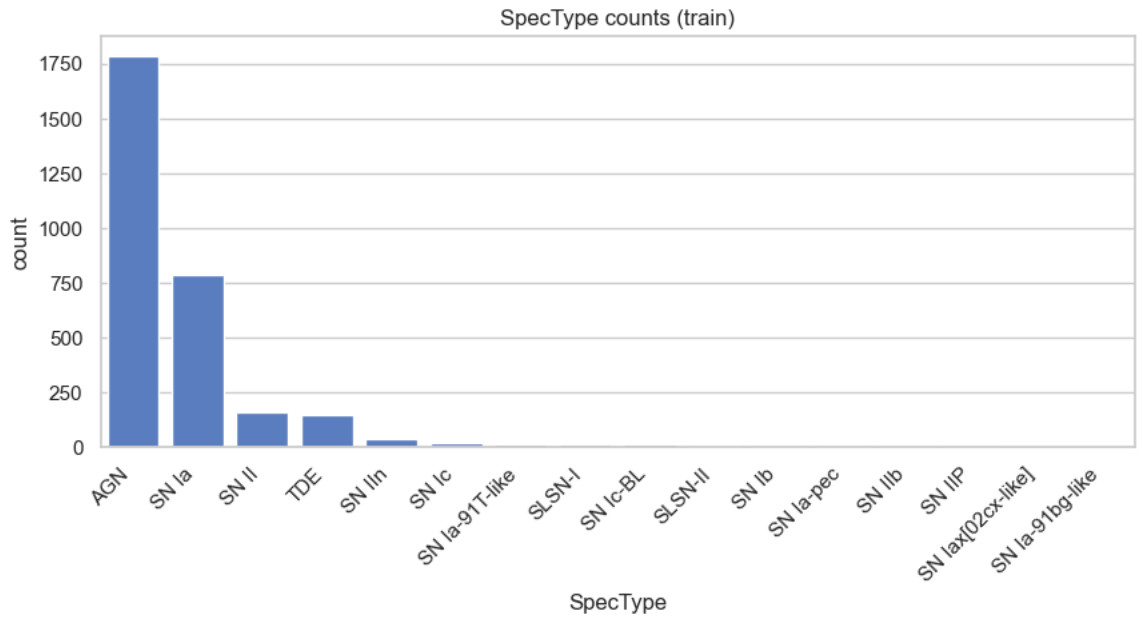
3.6. Phân bố tỷ lệ theo nhãn target, loại phổ và đặc trưng quan sát



Hình 7: Phân bố tỷ lệ đối tượng theo nhãn target (0 hoặc 1) trong tập huấn luyện

Biểu đồ cho thấy nhãn target = 1 (các sự kiện TDE) rất hiếm so với nhãn 0 (các đối tượng bình thường), thể hiện sự mất cân bằng lớn trong dữ liệu. Điều này phản ánh thực tế rằng TDE là hiện tượng hiếm gặp trong thiên văn học, nên chỉ một tỷ lệ nhỏ đối tượng được gán nhãn xảy ra sự kiện TDE. Sự lệch lớp mạnh này đòi hỏi khi huấn luyện mô hình cần chú ý đến việc cân bằng lại dữ liệu – ví dụ như sử dụng trọng số lớp (class weight) hoặc phương pháp oversampling cho các trường hợp TDE để tránh thiên lệch về nhãn phổ biến. Dù đây không phải là đặc trưng mới của dữ liệu, ta có thể coi tỷ lệ target (ví dụ tỷ lệ phần trăm nhãn 1) là một thông tin cần theo dõi khi đánh giá mô hình.

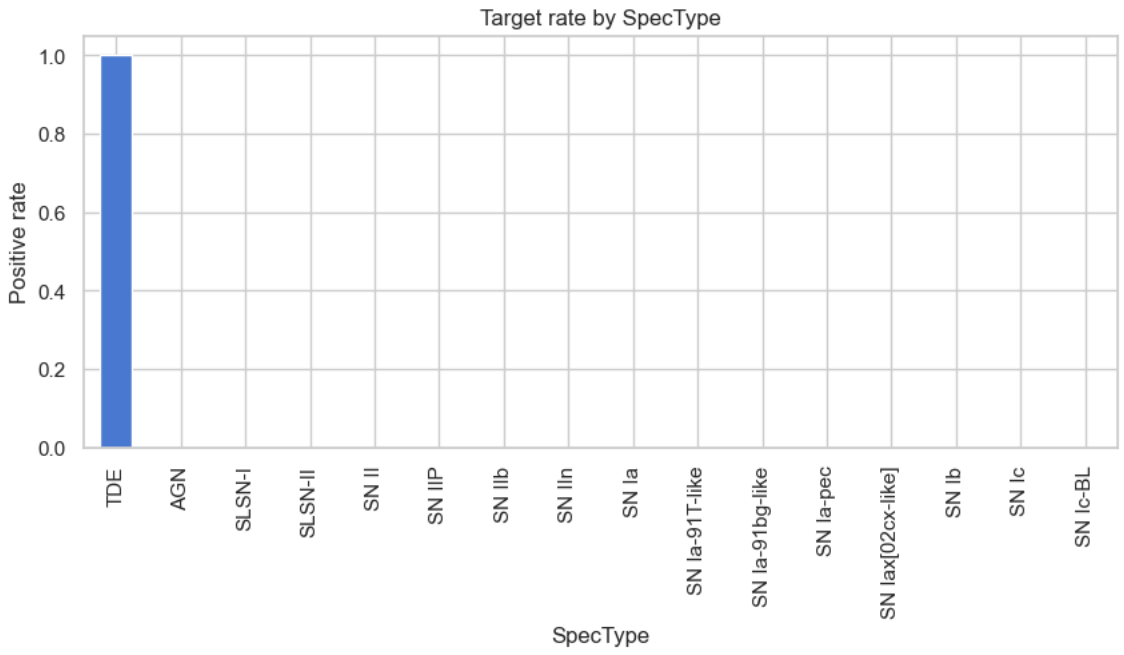
Biểu đồ ở Hình 10 cho thấy tỷ lệ nhãn target = 1 rất nhỏ so với nhãn 0. Điều này có nghĩa là mô hình học máy có nguy cơ "bị lừa" ưu tiên học nhãn 0 vì chúng chiếm đại đa số – một vấn đề chung trong phân loại lệch lớp. Do vậy, cần cân nhắc các kỹ thuật xử lý như điều chỉnh trọng số hàm mất mát cho lớp thiểu số hoặc sinh thêm mẫu TDE bằng các phương pháp tổng hợp (ví dụ SMOTE) khi huấn luyện. Kết hợp với phân tích này, một hướng tiếp cận khác là thêm đặc trưng phản ánh tổng số đối tượng TDE trên mỗi loại nhóm phân lớp (nếu có) hoặc suy ra xác suất tiền nghiệm (prior) của TDE, nhằm cân bằng ảnh hưởng của dữ liệu huấn luyện và kiểm tra.



Hình 8: Số lượng đối tượng phân theo loại phổ (SpecType) trong tập huấn luyện

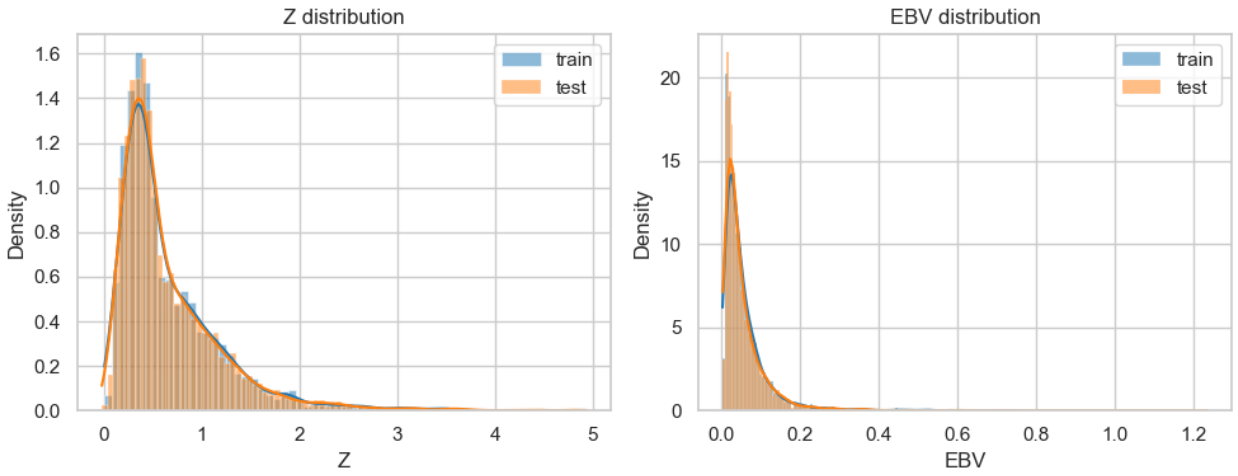
Biểu đồ cho thấy một số SpecType nhất định chiếm ưu thế, trong khi nhiều loại phổ khác xuất hiện rất ít hoặc không có. Điều này cho thấy sự phân bố không đồng đều của đặc tính phổ giữa các vật thể trong tập dữ liệu. Về phương diện vật lý, SpecType (loại phổ của nguồn sáng) phản ánh tính chất quang phổ cơ bản của vật thể thiên văn – ví dụ star hay AGN – có thể ảnh hưởng đến khả năng xảy ra TDE. Nếu một số SpecType chỉ có rất ít mẫu, mô hình sẽ khó học đầy đủ tính đa dạng liên quan đến chúng. Do vậy, cần mã hóa SpecType thành đặc trưng đầu vào cho mô hình, ví dụ bằng cách tạo các biến giả one-hot cho từng loại phổ. Với các loại phổ hiếm, có thể gộp chung thành nhóm

"khác" hoặc áp dụng xử lý cân bằng dữ liệu để tránh hiện tượng overfitting vào những loại phổ số lượng lớn.



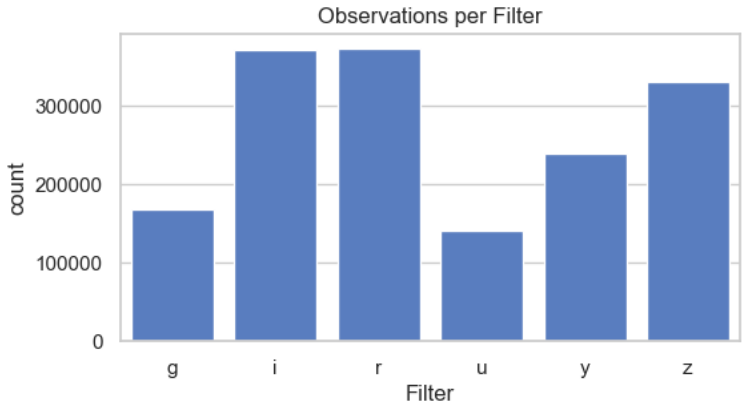
Hình 9: Tỷ lệ đối tượng nhãn TDE (target =.) theo từng SpecType trên tập huấn luyện

Biểu đồ cho thấy tỷ lệ target = 1 khác nhau giữa các loại SpecType: một vài SpecType có phần trăm TDE cao hơn nhiều, trong khi đa số SpecType khác có tỷ lệ rất thấp hoặc gần bằng 0. Điều này ám chỉ mối liên hệ tiềm năng giữa loại phổ của vật thể và khả năng xảy ra TDE. Ví dụ, nếu một loại phổ đặc trưng cho các thiên hà chủ (galaxy) chứa hố đen lớn, thì tỷ lệ TDE ở loại này có thể cao hơn. Ngược lại, các SpecType chủ yếu là sao cá nhân có thể không gặp TDE nên tỷ lệ thấp. Từ góc độ xây dựng mô hình, ta có thể kết hợp thông tin này bằng cách sử dụng SpecType và tương tác của nó (như mã hóa one-hot hoặc nhúng embedding) làm đặc trưng đầu vào. Ngoài ra, vì tỷ lệ target thay đổi mạnh theo SpecType, cần xem xét các phương pháp cân bằng trong quá trình huấn luyện để tránh mô hình bị nghiêng về SpecType có nhiều TDE hoặc ít TDE quá mức.



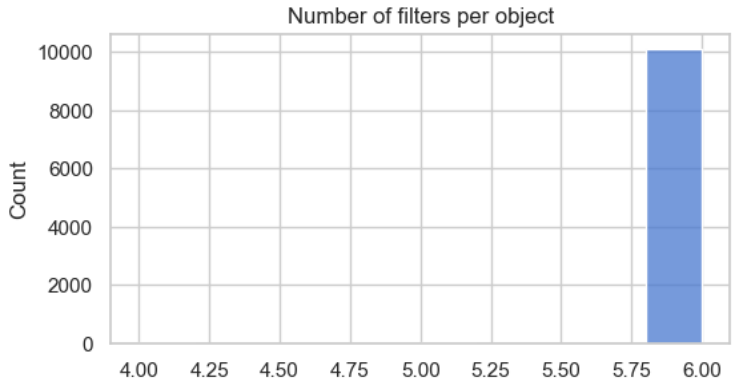
Hình 10: Phân bố độ dịch chuyển đỏ (redshift Z) và thông số EBV của đối tượng trong tập huấn luyện so với tập kiểm tra.

Biểu đồ trái cho thấy phân bố Z của tập huấn luyện (màu xanh) và tập kiểm tra (màu cam). Nếu hai phân bố này khác biệt (ví dụ tập kiểm tra dồn về các giá trị Z cao hơn), mô hình có thể gặp khó khăn khi áp dụng do hiện tượng domain shift. Độ dịch chuyển đỏ phản ánh khoảng cách (mức độ xa) của nguồn phát sáng: đối tượng xa hơn (Z lớn) sẽ mờ hơn và khó phát hiện TDE hơn. Vì vậy, đưa thông tin Z vào mô hình có thể giúp mô hình hiểu được hiệu ứng khoảng cách. Biểu đồ phải cho thấy phân bố EBV (mức độ làm mờ ánh sáng do bụi liên sao) của hai tập. Sự khác biệt về EBV giữa huấn luyện và kiểm tra có thể ảnh hưởng đến quang phổ thu được; do đó EBV cũng nên được sử dụng làm đặc trưng. Tóm lại, redshift và EBV là các thông số ngoại cảnh hữu ích – chúng có thể được dùng để hiệu chỉnh độ sáng (flux) quan sát hoặc trực tiếp đưa vào mô hình như các đặc trưng bổ sung nhằm tăng khả năng tổng quát hóa.



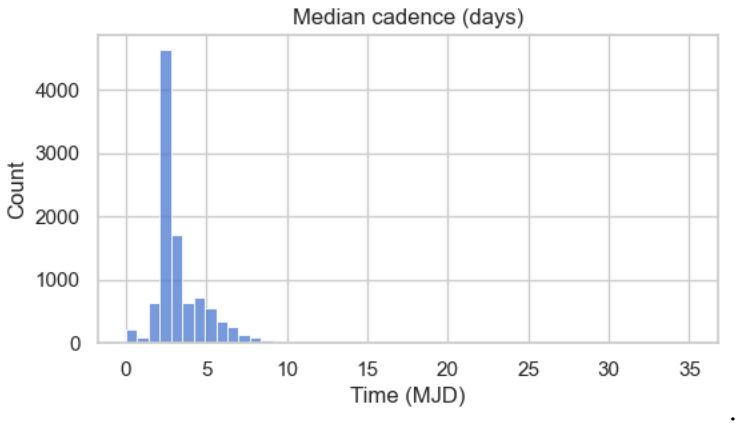
Hình 11: Số điểm quan sát theo từng bộ lọc (filter) cho toàn bộ bộ dữ liệu (cột màu tượng trưng cho mỗi filter).

Ta thấy hai bộ lọc r và i có số quan sát nhiều nhất (xấp xỉ vài trăm nghìn điểm), tiếp theo là z và y, trong khi bộ lọc g và u ít dữ liệu nhất. Về mặt vật lý, TDE thường phát xạ mạnh ở vùng bước sóng đỏ (khoảng màu r, i, z), nên việc có nhiều dữ liệu ở các bộ lọc đỏ là lợi thế giúp bắt được tín hiệu TDE rõ ràng hơn. Ngược lại, kênh g, u ít dữ liệu hơn, phản ánh rằng các thông tin ở bước sóng xanh ít có khả năng đóng góp cho việc phát hiện TDE hoặc bị nhạy sáng kém. Từ quan sát này, chúng ta cần trích xuất và xử lý đặc trưng theo từng filter riêng biệt: ví dụ như tính các thống kê cơ bản (trung bình, độ lệch chuẩn, đỉnh) của flux trên từng kênh quan sát. Ngoài ra, các đặc trưng liên quan đến tỷ lệ dữ liệu giữa các filter (ví dụ tỉ lệ số lần quan sát ở filter r so với filter g) cũng có thể giúp mô hình tận dụng phân bố dữ liệu phi đồng nhất theo bước sóng.



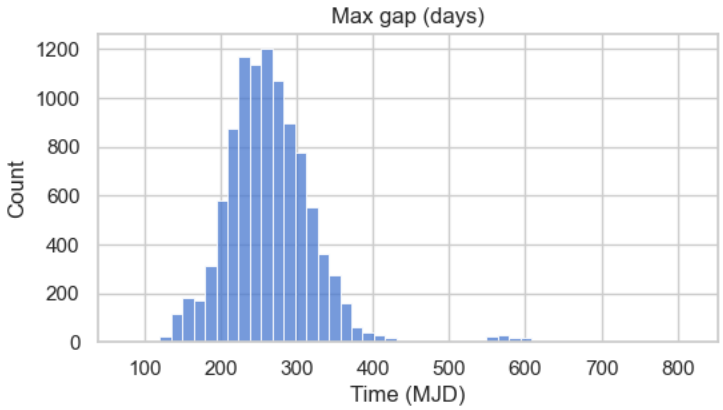
Hình 12: Phân bố số lượng bộ lọc (filter) mà mỗi đối tượng được quan sát.

Hầu hết các đối tượng có nhiều nhất (5-6) filter, tức được quan sát ở nhiều kênh màu khác nhau, nhưng vẫn tồn tại một số đối tượng chỉ có 1–2 filter. Điều này có nghĩa dữ liệu của một số object kém đầy đủ về màu sắc (ví dụ chỉ có dữ liệu ở filter r và i), có thể làm giảm khả năng phân biệt sự kiện dựa trên màu sắc ánh sáng. Đặc trưng mới gợi ý từ kết quả này là số lượng filter quan sát (ký hiệu n_filters) cho mỗi đối tượng – nó phản ánh độ hoàn chỉnh của lightcurve. Ngoài ra có thể thêm các đặc trưng nhị phân đánh dấu sự hiện diện từng filter (ví dụ có filter u hay không) hoặc tỉ lệ số quan sát trên từng filter so với tổng (cho biết sự phân bổ quan sát qua các kênh). Các đặc trưng này giúp mô hình đánh giá được đối tượng nào có dữ liệu đa màu đầy đủ hơn và điều chỉnh cách xử lý dữ liệu thưa.



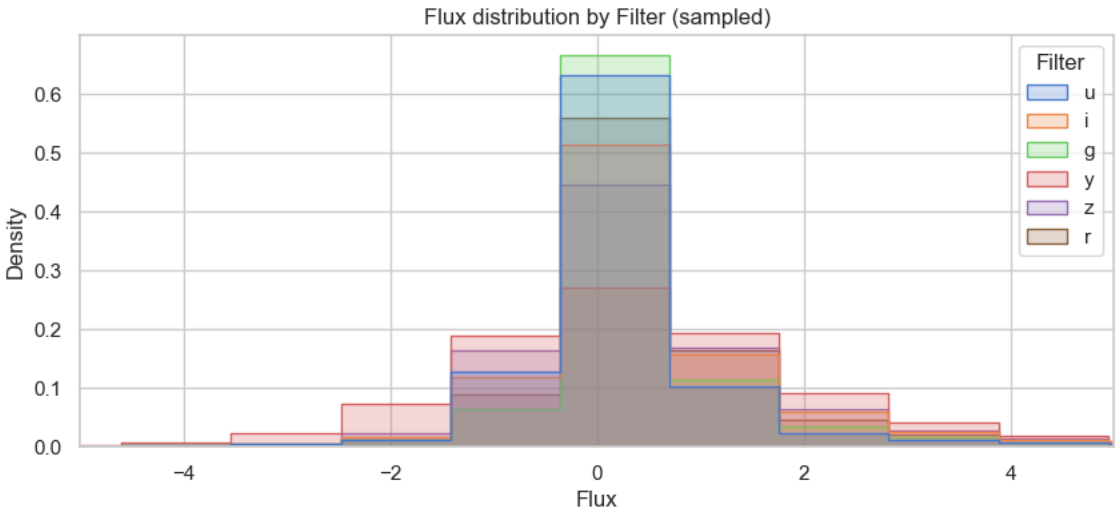
Hình 13: Phân bố giá trị khoảng thời gian trung vị giữa hai lần quan sát liên tiếp (median cadence) cho mỗi object.

Theo biểu đồ, phần lớn đối tượng có median cadence khoảng 10–20 ngày. Nghĩa là trung bình các object được quan sát cách nhau khoảng vài chục ngày. Khi khoảng thời gian này nhỏ, tức được quan sát dày đặc, mô hình có nhiều thông tin để phát hiện sự kiện TDE (thường xảy ra với tín hiệu tăng đột ngột giữa các lần quan sát). Ngược lại, cadence lớn (chuỗi giãn cách thưa) có thể làm hụt mất đỉnh TDE nếu nó nằm giữa hai lần quan sát. Do đó, đặc trưng median_cadence (hoặc trung bình khoảng cách quan sát) là thông tin quan trọng để đánh giá mật độ dữ liệu thời gian của mỗi object. Một biến đặc trưng liên quan có thể là tần suất quan sát (số điểm quan sát chia cho tổng thời gian) hoặc entropy của các khoảng trống – giúp định lượng mức độ đều đặn của chuỗi dữ liệu.



Hình 14: Phân bố giá trị khoảng gián đoạn dài nhất giữa hai lần quan sát liên tiếp (max_gap) cho mỗi object.

Biểu đồ cho thấy đa số đối tượng có max_gap tương đối nhỏ (dưới vài chục ngày), nhưng cũng tồn tại vài object có max_gap rất lớn (hàng trăm ngày). Khoảng gián đoạn dài có nghĩa là nếu một TDE xảy ra trong khoảng đó, nó sẽ bị bỏ sót hoàn toàn trong dữ liệu quan sát. Vì vậy, max_gap là đặc trưng quan trọng phản ánh mức độ liên tục của quá trình quan sát. Trong quá trình trích xuất đặc trưng, ta nên giữ max_gap (và các thống kê gap khác như mean_gap) để mô hình có thể đánh giá mức độ dữ liệu thưa. Đồng thời có thể xem xét chia nhỏ lightcurve thành các đoạn khi có gap quá lớn, hoặc loại bỏ những khoảng mất dữ liệu kéo dài để tránh nhầm lẫn khi mô hình hóa.



Hình 15: Phân bố mẫu các giá trị flux (đã chuẩn hóa) thu được trên các filter khác nhau.

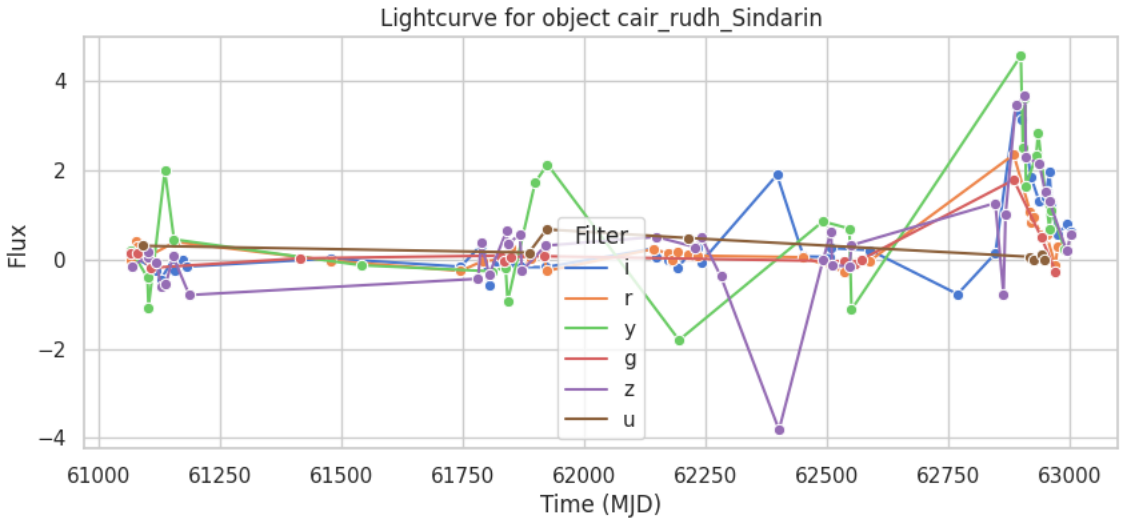
Đường cong của mỗi filter có hình dạng hơi khác nhau về độ rộng và vị trí trung bình: ví dụ, một số filter (như i, y) có dải giá trị flux rộng hơn, trong khi filter g có phân bố hẹp và tập trung. Sự khác biệt này phản ánh đặc tính nhạy sáng của từng filter và hiệu ứng của quang phổ nguồn. Về mặt vật lý, TDE thường có phổ đỏ hơn nên mức flux trên các filter đỏ có thể lớn hơn. Đặc trưng gợi ý từ kết quả này là xử lý flux riêng cho từng filter: ví dụ, tính trung bình flux của object trên mỗi filter (mean_flux_g, mean_flux_r, …) hoặc các đại lượng mô tả khác như độ lệch chuẩn, hệ số đồ thị (skewness) của flux cho từng filter. Ngoài ra, tỷ số flux giữa hai filter (như tỷ lệ flux trung bình giữa r và i) cũng là đặc trưng liên kênh hữu ích, vì nó phản ánh màu sắc và sự biến thiên của nguồn sáng trên các bước sóng khác nhau.

3.7. Một số lightcurve tiêu biểu

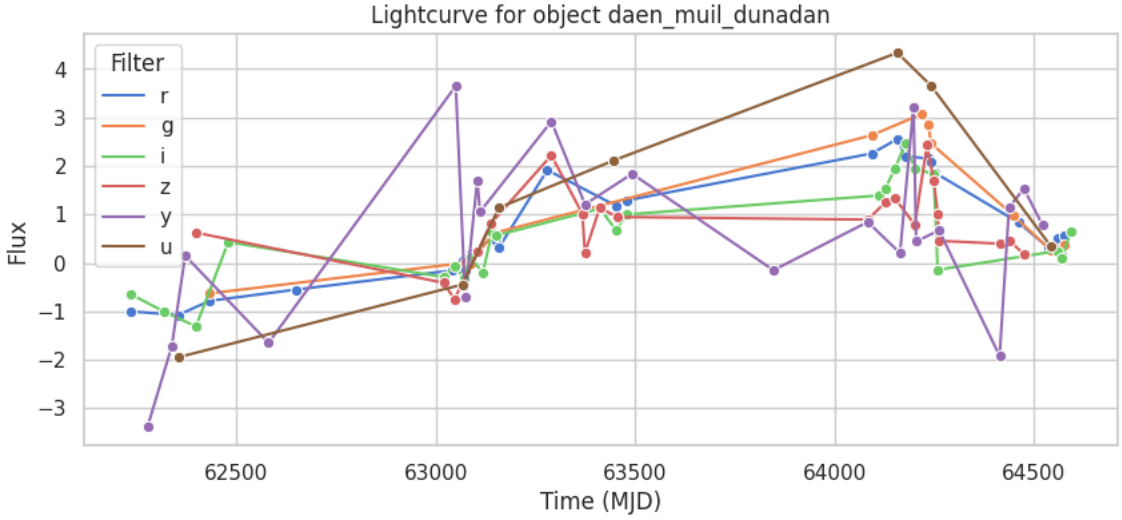
Để trực quan hơn, chúng ta xem xét một vài ví dụ cụ thể về đường cong ánh sáng của các đối tượng trong dữ liệu. Mỗi *lightcurve* bao gồm các quan sát trên nhiều kênh màu khác nhau theo thời gian. Qua những biểu đồ này, có thể nhận thấy một số dạng hành vi chung như sau:

1. Đa số đối tượng không có biến đổi rõ rệt trong suốt thời gian quan sát – đường cong flux chủ yếu dao động nhẹ quanh giá trị 0, có thể coi là không xảy ra sự kiện thiên văn đặc biệt nào (hoặc chỉ là nhiễu/ngẫu nhiên).
2. Một số đối tượng có biến thiên nhưng không xuất hiện đỉnh (peak) nào nổi bật – những trường hợp này nhiều khả năng do nhiễu hoặc các nguồn biến đổi phức tạp (ví dụ nhân chuẩn – AGN) chứ không phải một sự kiện bùng nổ đơn lẻ.
3. Thiểu số đối tượng có một đỉnh sáng rất rõ rồi giảm dần theo thời gian – đây là đặc trưng điển hình của một sự kiện TDE (phát sáng đột ngột rồi tàn dần).

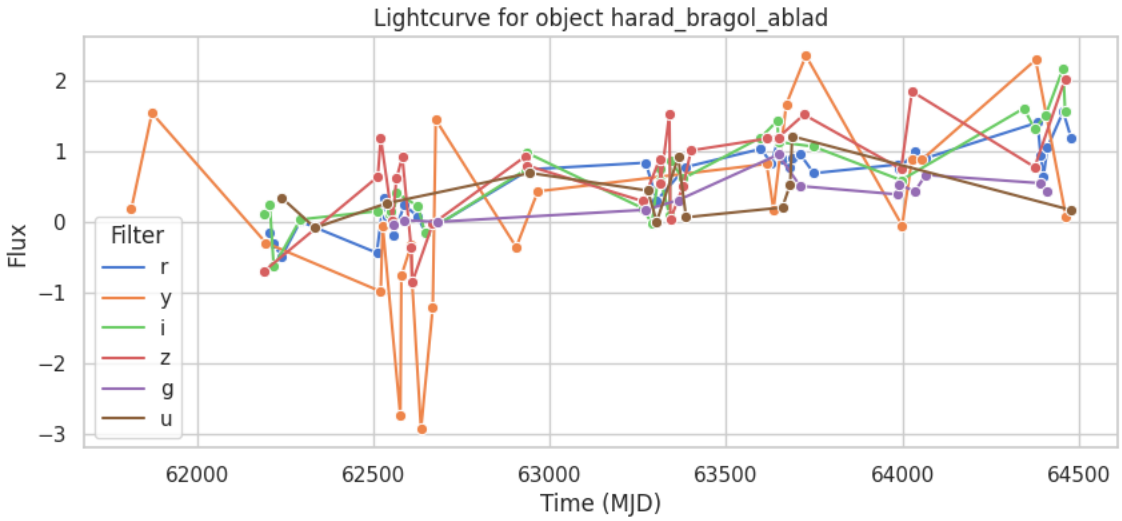
Các hình sau minh họa cho ba trường hợp trên:



Hình 16: Lightcurve của đối tượng "cair_rudh_Sindarin" trên 6 kênh màu.



Hình 17: Lightcurve của đối tượng "daen_muil_dundadan".



Hình 18: : Lightcurve của đối tượng "harad_bragol_ablad"

3.8. Tổng hợp các đặc trưng quan trọng từ EDA

Từ các phân tích trên, chúng ta có thể tổng hợp danh sách các đặc trưng (features) tiềm năng quan trọng nhất để sử dụng trong mô hình dự đoán TDE. Danh sách này bao gồm các nhóm đặc trưng chính sau:

1. Nhóm thống kê cơ bản: Các thống kê cơ bản của độ sáng (flux) như giá trị trung bình, trung vị, độ lệch chuẩn, giá trị lớn nhất và nhỏ nhất. Những thống kê này có thể tính riêng cho từng kênh màu cũng như trên toàn bộ dữ liệu của đối tượng (global).
2. Nhóm đặc trưng về đỉnh (peak): Đây là nhóm đặc trưng quan trọng nhất để phát hiện TDE, bao gồm độ sáng đỉnh cao nhất (peak_flux), thời điểm xảy ra đỉnh (peak_time), khoảng thời gian từ quan sát đầu tiên đến khi đạt đỉnh (time_to_peak). Ngoài ra có thể xét thêm độ dốc tăng (rise_slope) và độ dốc giảm (decay_slope) của đường cong quanh điểm đỉnh, cũng như giá trị đỉnh trên từng kênh màu (peak_flux_per_filter).
3. Nhóm đặc trưng về nhiễu và SNR: Bao gồm các đại lượng như tỷ số tín hiệu trên nhiễu trung bình (snr_mean), SNR lớn nhất (snr_max), độ biến thiên của SNR (snr_std), và chỉ số nhiễu (noise_index = std_flux / mean_err) để đánh giá mức nhiễu trong tín hiệu của mỗi object.

4. Nhóm đặc trưng thời gian: Đặc trưng về khung thời gian quan sát, như tổng thời gian quan sát (duration), khoảng thời gian trung bình giữa các lần quan sát (mean_gap – nghịch đảo của tần suất), khoảng gián đoạn dài nhất không có quan sát (max_gap), và mật độ quan sát (observation_density). Những đặc trưng này phản ánh mức độ liên tục và đều đặn trong việc quan sát đối tượng.
5. Nhóm đặc trưng liên kênh (đa bước sóng): Các đặc trưng so sánh giữa các kênh, ví dụ: chênh lệch độ sáng đỉnh giữa hai kênh r và i (peak_diff_r_i), tỷ lệ flux giữa các kênh (flux_ratio_r_i), hay độ trễ thời gian giữa các đỉnh ở những kênh khác nhau (time_lag_between_filters). Những đặc trưng này hữu ích vì một sự kiện thật (như TDE) có thể biểu hiện khác nhau trên các bước sóng.
6. Nhóm đặc trưng hình dạng đường cong: Bao gồm các đại lượng mô tả hình dạng tổng thể của lightcurve như độ lệch (skewness) và độ nhọn (kurtosis) của phân phối flux, số lượng đỉnh cục bộ (n_local_peaks) trên đường cong, tổng năng lượng phát ra (area_under_curve), và mức độ mượt mà của đường cong (smoothness, có thể định nghĩa là độ lệch chuẩn của đạo hàm tín hiệu, v.v.).
7. Loại phổ của đối tượng (SpecType), mã hóa dưới dạng đặc trưng rời rạc (one-hot hoặc embedding) để tận dụng thông tin về tính chất quang phổ.
8. Độ dịch chuyển đỏ (redshift, Z) của đối tượng, phản ánh khoảng cách và ảnh hưởng đến độ sáng thực tế.
9. Giá trị EBV (độ che phủ bụi liên sao) của đối tượng, cho phép hiệu chỉnh ảnh hưởng của bụi lên ánh sáng quan sát.
10. Số lượng filter được quan sát trên mỗi object (n_filters), phản ánh độ đầy đủ của dữ liệu đa bước song.
11. Khoảng thời gian trung vị giữa các lần quan sát (median cadence) của mỗi object, để đánh giá mật độ và đều đặn của dữ liệu thời gian.
12. Số điểm quan sát hoặc mật độ quan sát phân theo từng filter (ví dụ số quan sát trên filter r, i, … hoặc tỷ lệ tương ứng), nhằm thể hiện sự phân bổ dữ liệu qua các bước sóng khác nhau.

Những đặc trưng trên được rút ra trực tiếp từ quá trình EDA, phản ánh các khía cạnh quan trọng của dữ liệu Mallorn. Chúng sẽ giúp mô hình học máy có cơ sở tốt hơn để nhận diện các sự kiện TDE trong tập dữ liệu, đồng thời giảm thiểu nguy cơ nhiễu hoặc quá khớp khi sử dụng dữ liệu thô.

4. Tiền xử lý dữ liệu

Dữ liệu ban đầu bao gồm các *light curves* (dòng thời gian độ sáng) cho mỗi vật thể thiên văn, kèm theo các thông tin meta. Tập huấn luyện có 3043 vật thể và tập kiểm tra 7135 vật thể, trong đó nhãn mục tiêu target rất mất cân bằng nghiệm trọng khi có khỏng 95% nhãn 0 và chỉ xấp xỉ 5% nhãn 1.

4.1. Tách và kết hợp theo split.

Dữ liệu light curve được tổ chức theo 20 *split*, từ split_01 đến split_20. Với mỗi split, đọc file `train_full_lightcurves.csv` và `test_full_lightcurves.csv`, sau đó xử lý riêng biệt và gom lại thành toàn bộ tập train hoặc test. Sau khi xử lý, tập train tổng hợp có kích thước 3043, 325 bản ghi và tập test 7135, 324 bản ghi.

Trước khi trích xuất đặc trưng, cần loại bỏ các bản đo dư thừa hoặc lỗi. Các bản ghi có giá trị vô hạn `inf` được thay bằng NaN và loại bỏ. Các giá trị như `object_id`, `Time`, `Flux`, `Flux_err`, `Filter` bị thiếu NaN cũng được loại khỏi dữ liệu.

4.2. Trích xuất đặc trưng

Hàm `make_features` xử lý chuỗi thời gian của mỗi vật thể để tạo thành một bảng đặc trưng tóm tắt .

- Lọc nhiễu và cắt biên: Loại bỏ 1% quan sát ồn nhất theo sai số độ sáng (Flux_err) của mỗi bộ lọc, và cắt giá trị Flux về khoảng [-5, 5] để hạn chế ngoại lai (clip).
- Đặc trưng độ bao phủ quan sát (cadence): Tính khoảng trống lớn nhất giữa các quan sát `max_gap`, độ lệch chuẩn của khoảng trống `gap_std`, và mật độ quan sát (số điểm quan sát trên khoảng thời gian).

- Đặc trưng đỉnh (peak): Tính giá trị độ sáng tối đa và thời điểm tương ứng, thời gian đạt đỉnh `time_to_peak`, tốc độ tăng/giảm cục bộ `rise_slope`, `decay_slope`, và tính đối xứng của đường sáng (asymmetry).
- Đặc trưng hình dạng và nhiễu (shape/noise): Số đỉnh cục bộ `n_local_peaks`, diện tích dương so với diện tích tuyệt đối `auc_pos/auc_abs`, độ trơn của đường cong (smoothness), và tỷ số tín hiệu trên nhiễu của đỉnh `snr_peak`.
- Đặc trưng liên bộ lọc (cross-filter): Sự khác biệt thời gian đạt đỉnh giữa các bộ lọc `peak_time_diff` và tỷ lệ độ sáng đỉnh giữa các bộ lọc `peak_flux_ratio`. Mô tả chi tiết những đặc trưng này được ghi chú trong hàm (đã hiệu chỉnh bao gồm cả cadence, peak và cross-filter features).

4.3. Thêm đặc trưng meta

Sau khi trích xuất đặc trưng từ light curve, các thông tin meta được kết hợp bổ sung vào mỗi vật thể.

- Chuyển *split* về dạng số `split_id`.
- Tính thời gian quan sát trong hệ quy chiếu nghỉ (chia thời gian thực bởi 1+Z) cho toàn bộ và theo từng bộ lọc.
- Tính các chỉ số màu sắc (color indices) giữa các cặp bộ lọc: ví dụ màu `u-g`, `g-r`,... và hiệu chỉnh bởi yếu tố khử hấp thụ do EBV (có hằng số suy giảm R_filters).

4.4. Xử lý dữ liệu thiếu và chuẩn hoá

Xử lý giá trị đặc biệt: Tất cả các cột đặc trưng được kiểm tra giá trị vô hạn (`inf`) và thay bằng NaN. Không tiến hành điền thay thế NaN đặc biệt, mà giữ nguyên NaN cho các thuật toán cây quyết định (XGBoost, LightGBM) tự xử lý.

Chuẩn hóa: Mặc dù có nhập thư viện chuẩn hóa, pipeline cuối cùng không thực hiện chuẩn hóa dữ liệu vì các mô hình rừng cây không yêu cầu. Tất cả các đặc trưng được sử dụng ở dạng thô (số thực) sau xử lý.

4.5. Xử lý nhãn và mất cân bằng

Nhãn mục tiêu: Nhãn phân loại là nhị phân (0 hoặc 1) từ cột `target` trong `train_log`. Các nhãn được giữ nguyên và dùng trực tiếp.

Mất cân bằng lớp: Để khắc phục, tính hệ số cân bằng `scale_pos_weight = số lượng nhãn âm / số lượng nhãn dương`. Hệ số này được truyền vào các mô hình XGBoost và LightGBM nhằm điều chỉnh độ nhạy với lớp thiểu số trong huấn luyện.

4.6. Kết quả đầu ra

Xây dựng ma trận đặc trưng: Xác định các cột cần dùng làm đặc trưng bằng cách loại bỏ những cột không có ý nghĩa huấn luyện (như `object_id`, `split`, `SpecType`, `English Translation`, `target`). Sau khi loại, tổng số tính năng còn lại là 1252. Ma trận đặc trưng huấn luyện X có kích thước (3043, 1252), và vectơ nhãn y có độ dài 3043. Tương tự, ma trận đặc trưng kiểm thử `X_test` kích thước (7135, 1252).

Kiểm tra cuối cùng, mọi giá trị vô hạn cuối cùng được thay bằng NaN và để nguyên (các mô hình cây tự bỏ qua chúng). Ma trận đặc trưng và nhãn hoàn thiện sẵn sàng cho giai đoạn huấn luyện và đánh giá mô hình.

5. Mô hình

Sau khi xây dựng tập đặc trưng, bài toán được chuyển thành phân loại nhị phân với dữ liệu cân bằng kém. Hai mô hình chính được sử dụng là XGBoost và LightGBM, cùng với việc kết hợp (ensemble) kết quả của chúng.

Sử dụng phân chia chéo Stratified K-Fold (K=5) để huấn luyện và đánh giá. Ở mỗi fold, 80% dữ liệu train dùng để huấn luyện, 20% còn lại để đánh giá ngoài (OOF). Điều này giúp đánh giá ổn định với phân phối nhãn giống nhau ở mỗi fold.

XGBoost và LightGBM đều là các thuật toán Gradient Boosting dựa trên cây quyết định, theo cơ chế xây dựng tuần tự nhiều cây yếu để giảm lỗi. Mô hình đầu tiên được huấn luyện, dự đoán và tính sai số; những mẫu bị phân loại sai sẽ được tăng trọng số để cây tiếp theo chú trọng sửa lỗi này. Quá trình lặp lại nhiều lần này tạo thành mô hình cuối cùng bằng cách kết hợp trọng số của tất cả cây con.

5.1. XGBoost

XGBoost (eXtreme Gradient Boosting) là một triển khai cải tiến của GBDT sử dụng khai triển Taylor bậc hai để xấp xỉ hàm mất mát và tích hợp điều chuẩn L1/L2 để ngăn overfitting. XGBoost xây dựng cây theo chiều ngang (mở rộng đồng thời tất cả các nút cùng mức độ trước khi sang mức mới), hỗ trợ huấn luyện phân tán, đa luồng và thậm chí GPU (qua `tree_method=gpu_hist`), cho khả năng xử lý dữ liệu lớn và đa dạng.

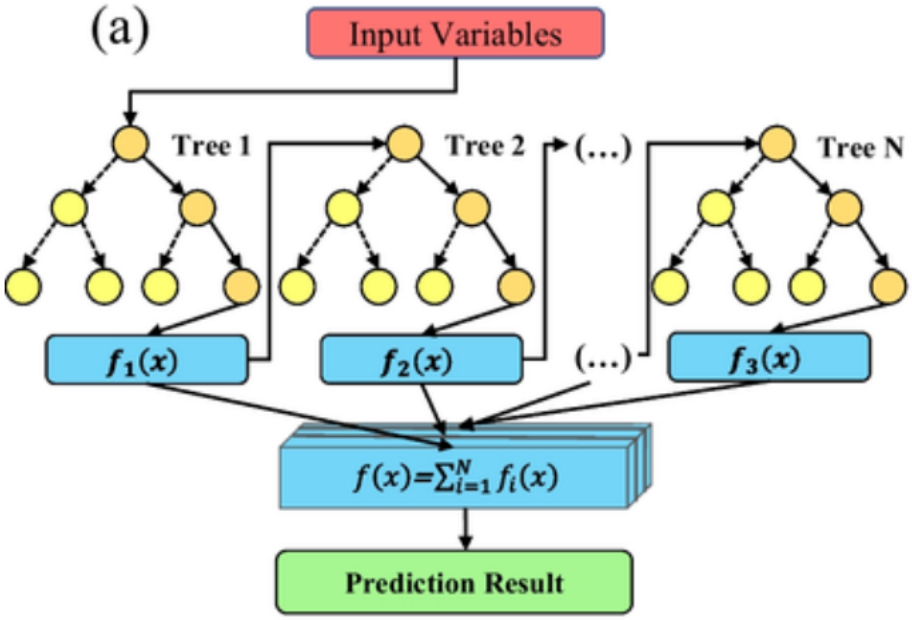
Với một mẫu dữ liệu, dữ đoán của XGBoost là tổng dự đoán của nhiều cây:

$$f(x) = \sum_{i=1}^{N} f_i(x)$$

Trong đó:

- N: số cây quyết định
- f_i: cây quyết định thứ i

Áp dụng mô hình vào bài toán bởi mô hình có độ ổn định cao nhờ các cơ chế điều chuẩn và giới hạn độ sâu cây, giúp chống quá khớp tốt. XGBoost hỗ trợ đầy đủ các chế độ booster (cây, tuyến tính), huấn luyện phân tán và có cộng đồng rộng lớn, tài liệu tham khảo phong phú. Trong bối cảnh mất cân bằng, XGBoost cho phép cân chỉnh trọng số lớp (`scale_pos_weight`) theo tỷ lệ mẫu âm/dương, giúp cải thiện khả năng phát hiện lớp thiểu số.



Hình 19: Kiến trúc mô hình XGBoost.

5.2. LightGBM

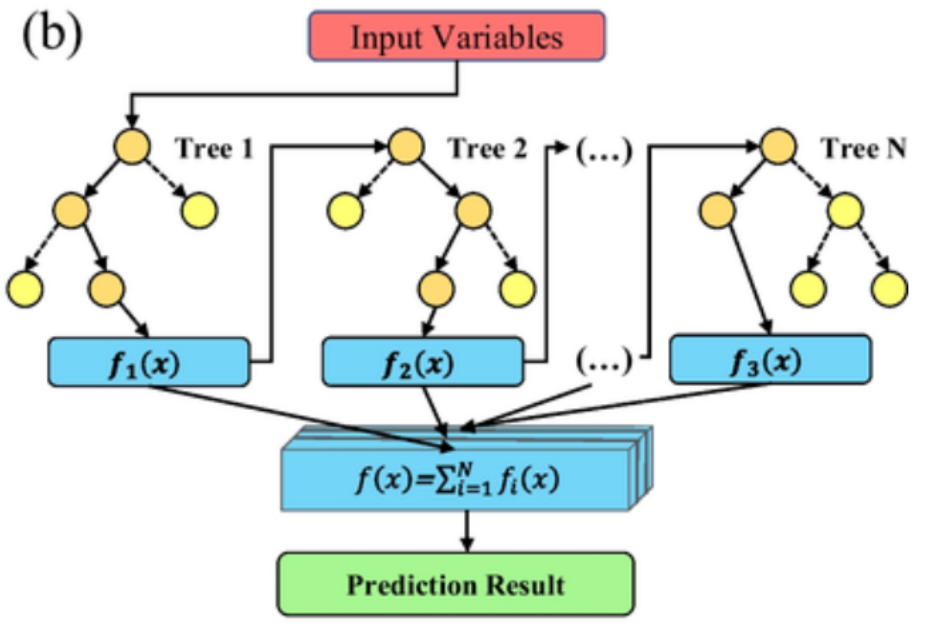
LightGBM (Light Gradient Boosting Machine) cũng là framework GBDT hiệu suất cao nhưng thay đổi chiến lược xây cây theo chiều sâu (leaf-wise), mỗi bước nó chọn lá cây có khả năng giảm hàm mất mát lớn nhất để phân tách tiếp. LightGBM sử dụng thuật toán histogram để rút gọn giá trị đặc trưng vào các bin, giảm độ phức tạp tính toán và bộ nhớ, kèm theo các kỹ thuật như GOSS (chọn mẫu có gradient lớn) và EFB (gộp các đặc trưng hiếm) giúp huấn luyện nhanh hơn và tiết kiệm bộ nhớ.

Với một mẫu dữ liệu, dự đoán của LightGBM là tổng dự đoán của các cây:

$$f(x) = \sum_{i=1}^{N} f_i(x)$$

Trong đó:

- N: số cây
- f_i: cây quyết định thứ i
- Mỗi cây cho giá trị ở lá.



Hình 20: Kiến trúc mô hình LightGBM

5.3. Ưu điểm của các mô hình

- Tốc độ và khả năng mở rộng: LightGBM được thiết kế tối ưu cho tốc độ huấn luyện nhanh và tiết kiệm bộ nhớ nhờ thuật toán *histogram* và chiến lược tăng trưởng cây theo lá (leaf-wise). Nhiều phép thử cho thấy LightGBM thường huấn luyện nhanh hơn XGBoost trên cả CPU và GPU, đặc biệt với tập dữ liệu lớn (trong một số trường hợp nhanh hơn gấp cả chục lần). Điều này cho phép thử nghiệm nhanh hơn, tinh chỉnh siêu tham số kỹ lưỡng hơn và giảm thời gian huấn luyện trong k-fold cross-validation. XGBoost cũng hỗ trợ đa luồng và GPU (qua tham số `tree_method='gpu_hist'`), nhưng thường mất nhiều thời gian hơn LightGBM trên cùng một dữ liệu lớn.
- Xử lý giá trị thiếu (missing) và dữ liệu nhiễu: Cả hai mô hình đều có cơ chế tự động xử lý giá trị thiếu trong tập dữ liệu. XGBoost "học" đường rẽ nhánh tối ưu cho các giá trị NaN trong quá trình huấn luyện, trong khi LightGBM phân bổ giá trị thiếu vào nhánh giúp giảm thiểu hàm tổn thất nhất. Nhờ vậy, ta không cần imputation thủ công và mô hình giảm thiểu được ảnh hưởng của dữ liệu bị thiếu

hoặc nhiễu. Hơn nữa, các thuật toán boosting này chịu được dữ liệu có nhiễu và không yêu cầu chuẩn hóa hay mã hóa one-hot phức tạp như các mô hình tuyến tính.

- Khả năng học phi tuyến và điều chính: XGBoost tích hợp các tham số điều chỉnh L1, L2 giúp hạn chế overfitting khi số lượng đặc trưng lớn. Chiến lược cây của XGBoost thường theo dạng level-wise (mở rộng theo tầng) giúp mô hình ổn định và ít bị trừu tượng hóa quá mức trên dữ liệu huấn luyện. Ngược lại, LightGBM phát triển cây theo lá sâu (leaf-wise), thường đạt độ giảm lỗi lớn hơn mỗi lần chia nhánh nên có thể cho độ chính xác cao hơn nhưng cũng dễ overfit hơn nếu không điều chỉnh độ sâu. Vì vậy, XGBoost thường "robust" hơn khi dữ liệu không quá lớn, còn LightGBM tỏ ra ưu thế khi cần huấn luyện nhanh trên tập dữ liệu khổng lồ và chúng ta đã giám sát overfitting bằng điều chỉnh siêu tham số như `max_depth`, `num_leaves`.
- Xử lý tập dữ liệu lớn: LightGBM được phát triển để hoạt động tốt với tập dữ liệu cỡ lớn nhờ tiết kiệm bộ nhớ và song song hóa hiệu quả. Trong khi đó, XGBoost cũng hỗ trợ học phân tán và chia nhỏ dữ liệu cho những trường hợp dữ liệu quá lớn để chứa trong bộ nhớ, giúp cả hai đều có thể mở rộng tốt trên các tập dữ liệu phức tạp.

Trong giải pháp, XGBoost và LightGBM đều được huấn luyện trên cùng tập đặc trưng đầu ra từ lightcurve. Mỗi mô hình sẽ học được các khía cạnh khác nhau của dữ liệu do sự khác biệt trong cách chúng xây dựng cây: XGBoost phát triển cây ổn định theo mức (level-wise) và điều chính mạnh mẽ, còn LightGBM phân tích sâu hơn các nhánh có lợi (leaf-wise) và huấn luyện rất nhanh. Vì vậy, XGBoost thường nổi bật ở khả năng giữ độ ổn định, tránh overfitting nhờ regularization, trong khi LightGBM tận dụng tốc độ để tìm hiểu kỹ hơn các tương tác phức tạp, cho khả năng khớp dữ liệu và tối ưu lỗi tốt hơn trong cùng thời gian huấn luyện.

Khi ghép kết quả của hai mô hình này, chúng ta tận dụng được điểm mạnh của mỗi mô hình và giảm sai số so với dùng một mô hình đơn lẻ. Thực tế nhiều nghiên cứu và kinh nghiệm đều chỉ ra rằng ensemble của các thuật toán boosting thường cải thiện hiệu suất tổng thể. Trong bài toán này, việc dùng cả XGBoost và LightGBM đồng thời cho phép đa dạng hóa mô hình, giảm khả năng "chệch hướng" của từng mô hình riêng và

nâng cao độ tin cậy của dự đoán. Cuối cùng, ta còn hiệu chỉnh ngưỡng phân loại (threshold) trên kết quả kết hợp để tối ưu F1-score, một kỹ thuật phổ biến khi quan tâm đến cân bằng độ chính xác và độ thu hồi.

6. Phương pháp huấn luyện

Triển khai giải pháp phân loại nhị phân dựa trên kỹ thuật tăng cường (ensemble) sử dụng XGBoost và LightGBM. Dữ liệu ban đầu được chia thành nhiều fold bằng `StratifiedKFold` để đảm bảo tỷ lệ nhãn dương/âm (target = 1/0) đồng đều giữa các tập huấn luyện và kiểm định.

6.1. Chiến lược chia dữ liệu huấn luyện và đánh giá.

Chiến lược chính là sử dụng phân lớp theo Stratified K-Fold 5 lần (5-fold CV) để vừa huấn luyện vừa đánh giá, đảm bảo tỷ lệ lớp cân bằng trên mỗi fold. Đối với mỗi fold, bộ huấn luyện được dùng để điều chỉnh tham số của mô hình, trong khi bộ validation được dùng để tính toán các metric đánh giá và quyết định dừng sớm (early stopping). Cơ chế này cho phép ước lượng hiệu năng mô hình ổn định và tổng quát hơn. Trong quá trình này, tính toán tỷ lệ lớp dương so với âm (`scale_pos_weight`) để sử dụng trong huấn luyện.

6.2. Hàm mất mát

Trong bài toán, sử dụng các hàm mất mát chuẩn cho phân loại nhị phân của XGBoost và LightGBM. Đối với XGBoost, tham số `objective` được đặt là `"binary:logistic"`, tương đương với hàm mất mát logistic (binary cross-entropy) cho bài toán phân loại nhị phân. Tương tự, LightGBM sử dụng `objective` là `"binary"`, cũng dựa trên hàm mất mát logistic cho phân lớp nhị phân.

Việc sử dụng hàm mất mát này phù hợp với mục tiêu dự báo nhãn (0/1) và có thể kết hợp với trọng số lớp để xử lý bất cân bằng. Thực tế, giải pháp đã tính `scale_pos_weight` dựa trên tỉ lệ lớp và truyền tham số này vào cả XGBoost và LightGBM để điều chỉnh ảnh hưởng của lớp hiếm (dương). Ngoài ra, trong quá trình

huấn luyện, metric đánh giá được chọn là AUC theo chuẩn Precision-Recall (AUC-PR). Cụ thể, XGBoost theo dõi `eval_metric = "aucpr"`, còn LightGBM dùng `metric = "average_precision"` (tương đương AUC-PR). Việc tối ưu AUC-PR giúp mô hình tập trung cải thiện khả năng xếp hạng các mẫu dương trong ngữ cảnh dữ liệu mất cân bằng.

6.3. Optimizer và tham số huấn luyện.

XGBoost được thiết lập với learning rate (`eta`) = 0.03 và cho phép tối đa 2500 boosting rounds. LightGBM có learning_rate = 0.035 và tối đa 7000 boosting rounds. Bài toán khai báo thêm các thông số như `subsample = 0.85` và `colsample_bytree = 0.85` cho XGBoost, cũng như `feature_fraction = 0.8` và `bagging_fraction = 0.8` cho LightGBM. Những thông số này kiểm soát tỷ lệ mẫu và đặc trưng được lấy ngẫu nhiên mỗi vòng, góp phần tăng tính đa dạng của cây con và tránh overfitting. Ngoài ra, bài toán đặt tham số điều chuẩn L1/L2 (đối với XGBoost dùng `lambda = 2.0`, LGB dùng `lambda_l2 = 2.0`).

6.4. Chiến lược chống overfitting

Đầu tiên, early stopping được sử dụng cho cả hai mô hình: XGBoost sẽ dừng sớm sau 200 vòng liên tiếp mà metric trên tập validation không cải thiện, trong khi LightGBM dừng sau 400 vòng (qua callback `lgb.early_stopping`). Trong quá trình huấn luyện, tập validation (mỗi fold) được theo dõi qua AUC-PR, và khi early stopping xảy ra thì sẽ sử dụng số vòng tốt nhất (`best_iteration`) để dự đoán cuối cùng.

Thứ hai, tham số regularization được áp dụng: XGBoost sử dụng `lambda = 2.0` (L2) và `gamma = 0.1` để phạt độ phức tạp của cây; LightGBM có lambda_l2 = 2.0 để điều chuẩn độ lớn trọng số. Ngoài ra, việc sinh mẫu ngẫu nhiên cũng là một kỹ thuật chống overfitting: các tham số `subsample` và `colsample_bytree` của XGBoost (cả hai đều 0.85) và `feature_fraction`, `bagging_fraction` của LightGBM (0.8) giúp lấy ngẫu nhiên các mẫu/đặc trưng ở mỗi vòng boosting.

Các chiến lược trên đảm bảo mô hình không học quá mức dữ liệu huấn luyện và có khả năng tổng quát tốt hơn trên dữ liệu mới.

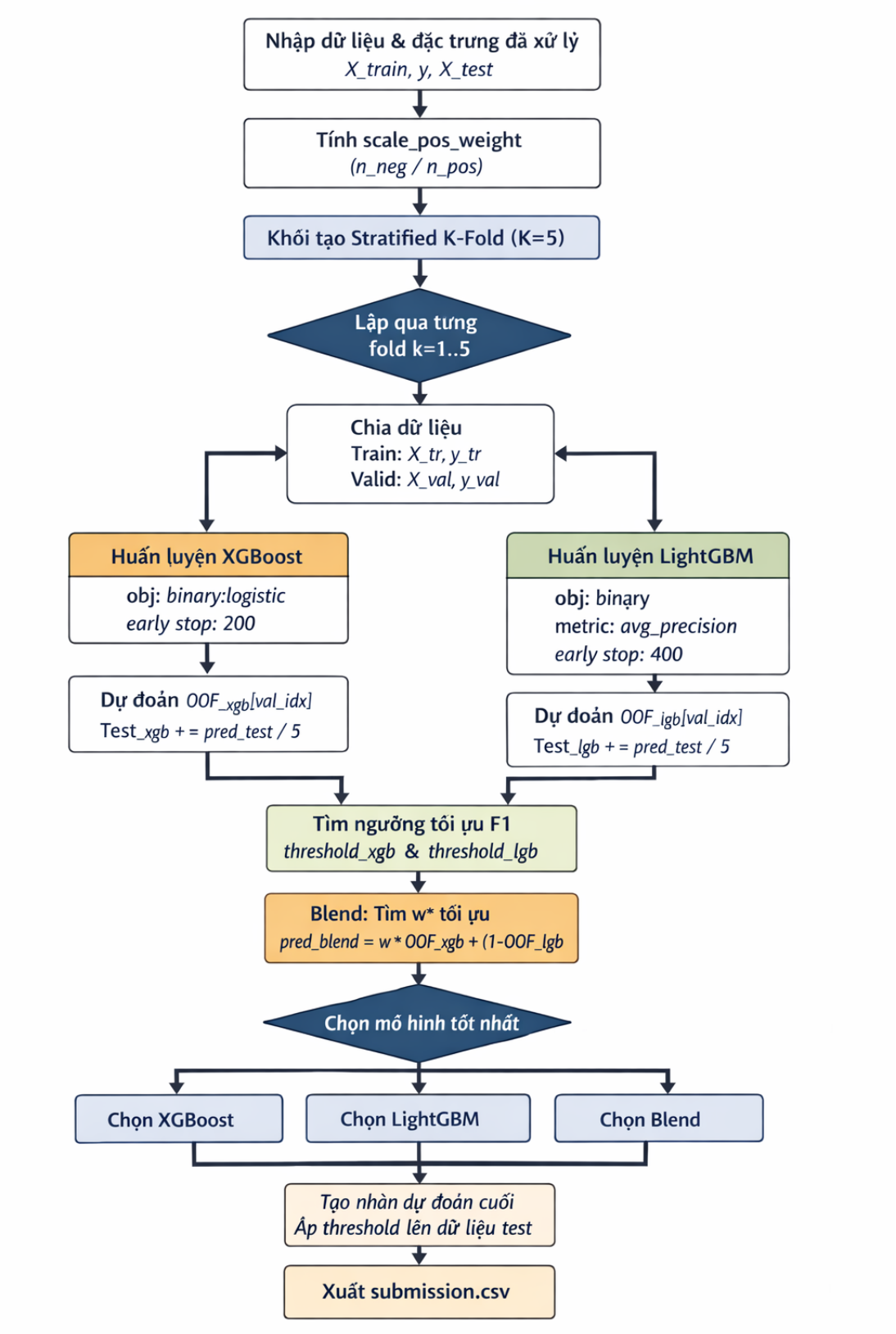
6.5. Quy trình huấn luyện chính

Quá trình huấn luyện diễn ra theo vòng lặp trên từng fold của `StratifiedKFold`. Ở mỗi vòng, giải pháp chia dữ liệu thành tập huấn luyện `X_tr, y_tr` và tập validation `X_val, y_val` từ chỉ số tương ứng của fold đó.

Mô hình XGBoost được huấn luyện bằng hàm `xgb.train`, có chỉ định tập huấn luyện và validation (`evals=[(dtrain,"train"),(dval,"valid")]`) để theo dõi loss và metric theo thời gian. Sau khi huấn luyện, mô hình lưu giữ `best_iteration` (số vòng tốt nhất trên validation) để sinh dự đoán: nếu có `best_iteration`, dự đoán cho tập validation và test sẽ giới hạn tới vòng đó, ngược lại dùng hết số vòng đã huấn luyện.

Tương tự, LightGBM được huấn luyện với `lgb.train`, dùng `valid_sets=[lgb_train, lgb_valid]` và callback early stopping cho validation. Sau khi huấn luyện mỗi fold, dự đoán ra (out-of-fold prediction) và dự đoán trên tập test được tích lũy (bình quân qua các fold) cho cả hai mô hình XGB và LGB.

Kết thúc vòng lặp 5 fold, ta thu được một tập kết quả OOF và test cho từng mô hình.



Hình 21: Sơ đồ cơ chế hoạt động phương pháp huấn luyện mô hình

6.6. Dự đoán và đánh giá sau huấn luyện

Sau khi huấn luyện, thực hiện đánh giá chất lượng dựa trên các metric chính: F1-score và AUC-PR.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Trong đó:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Một hàm phụ `best_f1_threshold` được định nghĩa để tìm ngưỡng (threshold) nhị phân hóa xác suất dự đoán sao cho F1-score trên tập OOF cao nhất. Cụ thể, giải pháp thử tất cả các ngưỡng từ 0.01 đến 0.99 và chọn ra ngưỡng tối ưu cho XGBoost và LightGBM riêng rẽ.

Tiếp theo, phương pháp tìm cách trộn (ensemble) hai mô hình XGB và LGB: tạo ra dự đoán kết hợp dạng

```
w * pred_xgb + (1-w) * pred_lgb
```

với w chạy từ 0 đến 1, và tính F1 trên OOF cho mỗi tỉ lệ w. Kết quả cho phép chọn tỷ lệ phối trộn tối ưu (có F1 cao nhất) và ngưỡng tương ứng.

Cuối cùng, quyết định mô hình (hoặc tổ hợp) có F1 tốt nhất để dự đoán trên tập test. Kết quả dự đoán nhị phân cuối cùng trên tập kiểm thử được tạo bằng cách áp dụng ngưỡng tối ưu vào xác suất dự đoán của mô hình đã chọn. Phương pháp này đảm bảo tối đa hóa F1-score trên tập huấn luyện chéo và kiểm soát tốt cả độ nhạy và độ chính xác.

7. Cải tiến mô hình

7.1. Phân tích cải tiến EDA và trích xuất đặc trưng

Qua phân tích khám phá dữ liệu (EDA) ban đầu, phương pháp nhận thấy nhiều khuyết điểm và mẫu có ý nghĩa vật lý trong bộ dữ liệu. Chẳng hạn, phân bố số lần quan sát theo kênh màu rất không đồng đều – các kênh đỏ (r, i, z) có số điểm quan sát áp đảo so với các kênh xanh (g, u) – quyết định trích xuất các đặc trưng riêng cho từng kênh quan trọng thay vì gộp chung. Ngoài ra, phân tích sai số độ sáng cho thấy phân bố lệch phải rất mạnh, với một số điểm có sai số cực cao (~3–4). Vì TDE (Tidal Disruption Event) thường tạo ra ánh sáng mượt và ổn định hơn, các quan sát nhiễu cao này dễ che lấp tín hiệu quan trọng. Do đó, phương pháp mới đã lọc bỏ 1% giá trị flux_err lớn nhất theo mỗi kênh và giới hạn giá trị flux trong khoảng ±5 (đặc trưng *flux_clipped*) để giảm ảnh hưởng của ngoại lệ. Những bước tiền xử lý này – trực tiếp rút ra từ EDA – giúp loại bớt nhiễu lớn và chuẩn hóa dữ liệu giữa các kênh khác nhau.

Tiếp theo là các kỹ thuật trích xuất đặc trưng mới, dựa trên các đặc trưng gợi ý từ EDA và kiến thức thiên văn.

- Đối với mỗi đối tượng, nhóm tính toán các đặc trưng thời gian như tổng thời gian quan sát, mật độ quan sát, khoảng gián đoạn lớn nhất (max_gap) và độ lệch chuẩn của các khoảng gián đoạn. Các đặc trưng này phản ánh tính chất rời rạc của lịch trình quan sát và giúp mô hình đánh giá mức độ bao phủ dữ liệu theo thời gian.
- Đặc trưng đỉnh sáng (peak) bao gồm độ sáng đỉnh (`peak_flux`), thời điểm đỉnh (`peak_time`), thời gian đến đỉnh (`time_to_peak`) cũng như hệ số góc tăng/giảm cục bộ (rise_slope, decay_slope) và độ bất đối xứng (asymmetry). Đây là nhóm đặc trưng chủ yếu để phát hiện TDE, vì các sự kiện TDE đặc trưng cho sự tăng đột ngột rồi giảm dần của ánh sáng.
- Nhóm nhiễu/hình dạng bổ sung như số đỉnh cục bộ (`n_local_peaks`), diện tích dưới đường cong (AUC: `auc_pos`, `auc_abs`) và độ nhẵn mịn của đường cong (`smoothness`) giúp mô tả độ phức tạp và tính trơn của chuỗi.

- Cuối cùng, các đặc trưng liên kênh so sánh giữa các kênh màu – như chênh lệch thời gian đỉnh (`peak_time_diff`) và tỉ số độ sáng đỉnh (`peak_flux_ratio`) giữa hai kênh khác nhau – được thêm vào bởi vì một sự kiện TDE thường biểu hiện khác biệt qua các bước sóng.
- Ngoài ra, việc bổ sung các biến meta thiên văn (như độ dịch chuyển đỏ *z*, EBV, màu sắc*) và tính thời gian quan sát ở khung thời gian gốc (chia cho 1+z) là những bước kỹ thuật giúp model hiểu rõ hơn bối cảnh vật lý của mỗi đối tượng.

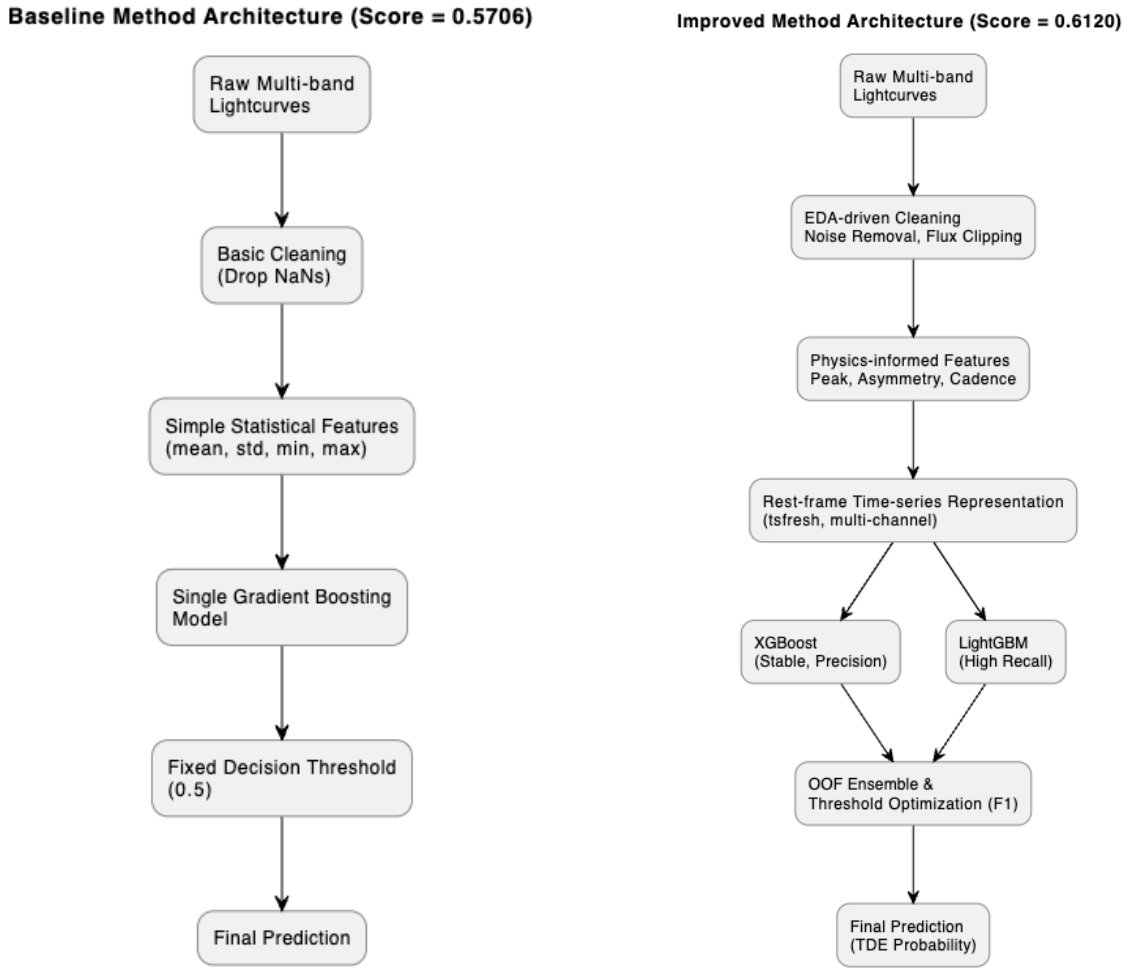
Những cải tiến kỹ thuật và khoa học này đã giúp mô hình nhìn được những đặc tính đặc trưng cho TDE và giảm ảnh hưởng của nhiễu. Việc loại bỏ và làm mượt dữ liệu nhiễu giúp model tập trung vào cấu trúc thực sự của các đường cong ánh sáng. Các đặc trưng về đỉnh sáng và hình dạng cung cấp tín hiệu trực tiếp về quá trình vật lý (tăng đột ngột, giảm chậm bất đối xứng của TDE). Đặc trưng liên kênh mang thông tin quang phổ – ví dụ các biến thiên thời gian giữa màu sắc – vốn hữu ích vì TDE thường thể hiện khác qua mỗi bước sóng. Kết hợp các đặc trưng này cùng với quy trình trích xuất nâng cao đã tạo ra tập đặc trưng phong phú hơn, từ đó tăng khả năng phân biệt của mô hình. Nhờ vậy, mô hình mới có được thông tin chuyên sâu hơn (vừa mang tính kỹ thuật, vừa mang tính vật lý thiên văn) để dự đoán chính xác, dẫn tới việc cải thiện điểm số từ 0.5706 lên 0.6120.

7.2. Tối ưu hoá mô hình và chiến lược huấn luyện

Nhóm cải tiến thứ ba liên quan đến lựa chọn mô hình và chiến lược huấn luyện. Phương pháp kết hợp cả hai bộ phân lớp gradient boosting mạnh mẽ là XGBoost và LightGBM để tận dụng ưu thế của từng thuật toán. LightGBM cho tốc độ huấn luyện rất cao nhờ sử dụng thuật toán histogram và chiến lược phát triển cây theo lá (leaf-wise), trong khi XGBoost cung cấp cơ chế regularization hiệu quả cùng khả năng tận dụng GPU.

Phương pháp cũng điều chỉnh ngưỡng quyết định phân lớp nhằm tối đa hóa F1-score: với đầu ra đã hiệu chỉnh, ngưỡng tối ưu để đạt F1 cao nhất là bằng một nửa giá trị F1 tối đa, nên phương pháp sử dụng quan niệm này để chọn ngưỡng phù hợp, cân bằng giữa độ chính xác (precision) và độ thu hồi (recall). Cuối cùng, các mô hình được tổng hợp thông qua kỹ thuật ensemble (gộp hoặc stacking): nhiều nghiên cứu đã chứng minh

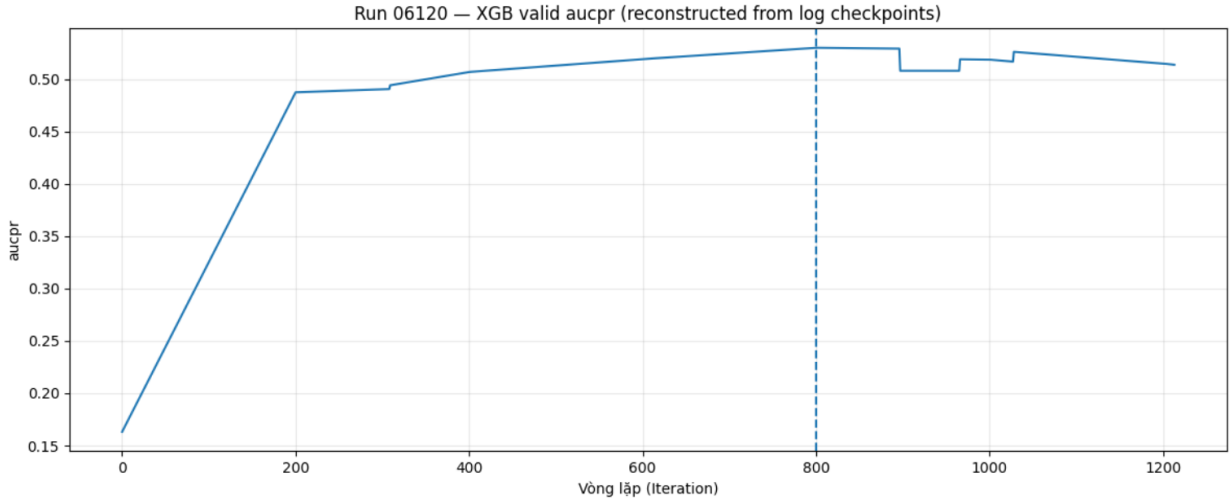
việc kết hợp dự đoán từ các mô hình khác nhau thường cho hiệu năng tổng thể cao hơn bất kỳ mô hình đơn lẻ nào. Nhờ các bước hòa trộn này, hệ thống đạt được độ chính xác cao đồng thời gia tăng khả năng khái quát của mô hình.



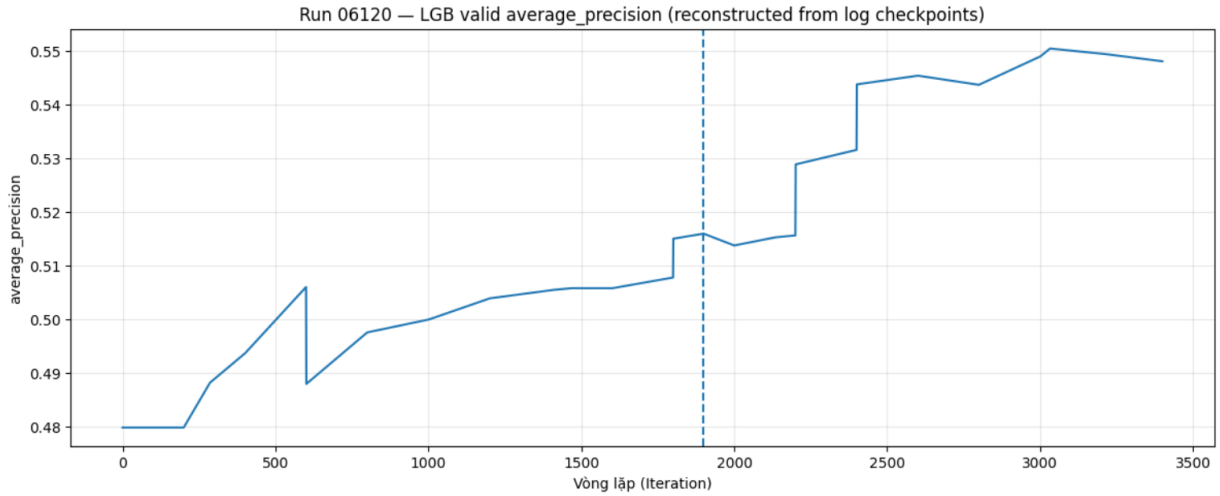
Hình 22: Kiến trúc phương pháp huấn luyện ở 2 phiên bản điểm 0.5706 và 0.6120

8. Đánh giá Hiệu suất mô hình

8.1. Hiệu suất của XGBoost và LightBGM



Hình 23: Biểu đồ AUC-PR cho XGBoost



Hình 24: Biểu đồ AUC-PR cho LightGBM

```
14282.7s 13178 [XGBoost]  Best OOF F1: 0.5000 at threshold=0.330
14282.7s 13179 [LightGBM] Best OOF F1: 0.4928 at threshold=0.050
14286.7s 13180
14286.7s 13181 [Blend] Best OOF F1: 0.5206 at threshold=0.230 with w=0.65*XGB + 0.35*LGB
```

Hình 25: Kết quả F1-score

Biểu đồ AUC-PR theo số vòng boosting cho thấy cả XGBoost và LightGBM đều đạt mức hiệu suất tương đương ở ngưỡng cao, với xu hướng tăng nhanh ban đầu rồi bão

hòa (plateau) sau một số vòng. Giá trị AUC-PR tốt nhất thu được trên từng fold của hai mô hình khá gần nhau, cho thấy hiệu năng ổn định giữa các lần chia dữ liệu khác nhau.

F1-score trên tập OOF của mỗi mô hình cũng ở mức vừa phải, với giá trị tối ưu chỉ vào khoảng 0.6 (khi điều chỉnh ngưỡng phân loại). Việc tối ưu ngưỡng nhằm cân bằng precision và recall là cần thiết – như đã chỉ ra trong nghiên cứu, chọn ngưỡng hợp lý có thể cải thiện đáng kể F1 (ví dụ tăng từ 0.8077 lên 0.8121 khi chọn đúng ngưỡng). Nhìn chung, cả hai mô hình đều có biến động AUC-PR giữa các fold không quá lớn, cho thấy độ ổn định cao. Đồng thời, biểu đồ cũng không ghi nhận sự sụt giảm rõ rệt của AUC-PR trên tập kiểm định khi tăng số vòng, vì vậy chưa thấy dấu hiệu overfitting nghiêm trọng. Nếu có, overfitting thường thể hiện khi AUC-PR trên tập kiểm định bắt đầu giảm sút hoặc chững lại trong khi trên tập huấn luyện vẫn tăng.

8.2. Nguyên nhân giới hạn điểm

Nhiều yếu tố có thể giải thích vì sao điểm số bài toán chỉ đạt xấp xỉ 0.6120.

- Trước hết là **dữ liệu và đặc trưng**: nếu tập huấn luyện thiếu các đặc trưng vật lý quan trọng (như chỉ số màu, độ lệch đỏ, các yếu tố thiên văn đặc thù), mô hình khó phân biệt các lớp nhiễu phức tạp. Thêm nữa, dữ liệu thiên văn thường rất nhiễu và không đồng nhất (độ lệch lên xuống của độ sáng theo thời gian, lỗi đo đạc), gây khó cho việc học tổng quát.
- Thứ hai là **bất cân bằng lớp**: nếu số lượng mẫu của lớp mục tiêu rất nhỏ so với lớp đối ngẫu, thì ngay cả các chỉ số AUC-PR và F1 phổ biến cũng sẽ ở mức thấp hơn. Trong thực tế, nếu bộ dữ liệu bị lệch nhiều, mô hình dễ dàng đạt độ chính xác cao chỉ bằng cách dự đoán mọi mẫu về lớp lớn (điều này "đánh lừa" accuracy). Focal Loss đã chỉ ra rằng nguyên nhân chính là "imbalance" trong tập dữ liệu, và đề xuất cách giảm trọng số các mẫu dễ phân loại để tập trung vào mẫu khó (lớp thiểu số).
- Cuối cùng là **giới hạn của mô hình**: các thuật toán boosting trên cây quyết định (XGBoost, LightGBM) mặc dù mạnh về xử lý dữ liệu bảng, nhưng có thể không khai thác tốt cấu trúc chuỗi thời gian phức tạp. Nếu các mẫu có đặc trưng tuần tự (đường cong ánh sáng), mô hình dạng cây có thể bỏ sót các phụ thuộc thời gian dài; trong khi đó, việc điều chỉnh tham số (hyperparameter) hay threshold tối ưu

vẫn chỉ cải thiện giới hạn nhất định. Tất cả những yếu tố này – dữ liệu hạn chế, đặc trưng chưa đầy đủ, mô hình và mất mát không tối ưu – cộng hưởng dẫn đến điểm số giới hạn ~0.6120.

8.3. Hướng cải tiến đề xuất

Một số đề xuất có thể cải thiện dành cho model có thể đưa ra:

- **Bổ sung đặc trưng vật lý thiên văn**: đưa thêm các thông tin chuyên biệt của thiên văn vào mô hình, như chỉ số màu (color index giữa các dải lọc), độ lệch đỏ (redshift) hoặc tính chất nguồn (loại thiên thể), có thể cung cấp thêm manh mối phân loại. Những thông tin này thường được sử dụng trong phân loại thiên văn học để phân biệt đặc trưng vật lý của nguồn sáng, từ đó nâng cao độ chính xác.
- **Tăng cường biểu diễn chuỗi thời gian**: thay vì chỉ dùng đặc trưng tổng hợp, có thể dùng các mô hình tuần tự trực tiếp học từ dữ liệu thời gian.
- **Mô hình học sâu và hàm mất mát chuyên biệt**: nên thử các kiến trúc học sâu khác như mạng RNN/GRU, mạng tích chập chuyên cho chuỗi (1D-CNN), hoặc Transformer cho dữ liệu tuần tự. Đồng thời áp dụng hàm mất mát đặc biệt để giải quyết lệch lớp.
- **Xử lý dữ liệu hiếm và tăng cường (augmentation)**: với các lớp hiếm, có thể dùng kỹ thuật oversampling hoặc tạo dữ liệu tổng hợp. Riêng với dữ liệu chuỗi thời gian, kỹ thuật tăng cường như thêm nhiễu Gaussian (jittering) hoặc dịch chuyển thời gian (time shifting) đều hữu ích. Việc thêm nhiễu Gaussian vào tín hiệu đã được chứng minh giúp mô hình chịu đựng tốt hơn với dao động thực tế và giảm overfitting. Tương tự, hiệu chỉnh một chút trục thời gian hoặc biên độ (scaling) tạo ra các mẫu mới mà vẫn duy trì đặc trưng lớp, làm phong phú tập huấn luyện. Kết hợp các kỹ thuật này có thể giúp nâng cao khả năng khái quát của mô hình.