Child Mind Institute — Problematic Internet Use Report

Nguyễn Đặng Minh Quan (22521183)
Trần Mạnh Phúc (22521142)
Trần Quang Đạt (22520536)

1.Giới thiệu:

Mục tiêu của cuộc thi là phân tích dự đoán vấn đề sử dụng internet bất thường của trẻ em và thanh thiếu niên. Dựa trên các chỉ số thể chất, thể lực, thói quen, chế độ ăn phổ biến dể dự đoán phát hiện sớm các vấn đề tồn động.

Một số đặc điểm cần được lưu ý trong cuộc thi như:

- Việc quá tập trung vào dữ liệu public có thể khi có sự chênh lệch giữa tập test public và private thì kết quả thu được sau cuộc thi không như mong đợi.
- Các mô hình như XGBoost, CatBoost, LightGBM,... thường được sử dụng trong các cuộc thi Kaggle và thường đạt được kết quả rất tốt.
- Tham khảo cá phương án và giải pháp từ những người có kinh nghiệm từ phần **Discussion** và Public code để có hướng giải quyết dễ dàng hơn. **III. Tổng quan dữ liệu**

Công Thức

1. Hệ số Kappa có trọng số bậc hai (Quadratic Weighted Kappa):
   - Trọng số (W[i, j]):

$$W[i,j] = \frac{(i-j)^2}{(N-1)^2}$$

     - i, j: Các giá trị thực tế và dự đoán.
     - N: Số lượng hạng mục (classes).
   - Kappa:

$$Kappa = 1 - \frac{\sum (W \cdot O)}{\sum (W \cdot E)}$$

     - W: Ma trận trọng số.
     - O: Ma trận quan sát.
     - E: Ma trận kỳ vọng.
   - Ý nghĩa:
     - $Kappa = 1$: Mô hình hoàn hảo.
     - $Kappa = 0$: Dự đoán ngẫu nhiên.
     - $Kappa < 0$: Mô hình tệ hơn dự đoán ngẫu nhiên.

HBN Instruments

Dữ liệu trong train.csv và test.csv gồm các đo lường từ nhiều công cụ khác nhau, được mô tả chi tiết trong data_dictionary.csv. Các công cụ bao gồm:

1. **Demographics**: Thông tin về tuổi và giới tính của người tham gia.
2. **Internet Use**: Số giờ sử dụng máy tính/internet mỗi ngày.
3. **Children's Global Assessment Scale**: Thang điểm đánh giá chức năng tổng quát của trẻ dưới 18 tuổi.
4. **Physical Measures**: Dữ liệu về huyết áp, nhịp tim, chiều cao, cân nặng, vòng eo và hông.
5. **FitnessGram Vitals and Treadmill**: Đánh giá thể lực tim mạch theo giao thức NHANES.
6. **FitnessGram Child**: Đo lường 5 thông số về thể lực liên quan đến sức khỏe (độ bền, linh hoạt, sức mạnh cơ, v.v.).

7. **Bio-electric Impedance Analysis**: Thành phần cơ thể gồm BMI, mỡ, cơ và nước.
8. **Physical Activity Questionnaire**: Mức độ tham gia các hoạt động mạnh trong 7 ngày qua.
9. **Sleep Disturbance Scale**: Phân loại các rối loạn giấc ngủ ở trẻ.
10. **Actigraphy**: Đo lường khách quan hoạt động thể chất bằng thiết bị đeo tay.
11. **Parent-Child Internet Addiction Test (PCIAT)**: Thang đo 20 mục đánh giá nghiện internet. Trường PCIAT_Total là cơ sở tính chỉ số mục tiêu sii:
    a. **0**: Không (None)
    b. **1**: Nhẹ (Mild)
    c. **2**: Vừa (Moderate)
    d. **3**: Nặng (Severe)

Actigraphy Files

Dữ liệu gia tốc được ghi từ thiết bị đeo tay trong tối đa 30 ngày liên tục:

- **Series Files (series_train.parquet và series_test.parquet)**: Dữ liệu từng cá nhân theo ID.
- **Biến số chính**:
    - **X, Y, Z**: Gia tốc đo theo các trục.
    - **enmo**: Norm Euclidean Minus One, biểu thị mức chuyển động (0: không chuyển động). ○ **anglez**: Góc của cánh tay so với mặt phẳng ngang. ○ **non-wear_flag**: 0: thiết bị đang được đeo, 1: thiết bị không được đeo. ○ **light**: Đo ánh sáng môi trường (lux). ○ **battery_voltage**: Điện áp pin (mV).
    - **time_of_day**: Thời gian bắt đầu mỗi khung 5 giây. ○ **weekday**: Ngày trong tuần (1: Thứ Hai, 7: Chủ Nhật).
    - **relative_date_PCIAT**: Số ngày kể từ khi bài kiểm tra PCIAT được thực hiện.

III. Số điểm đạt được trong cuộc thi

Trong đồ án lần này tập trung cải thiện điểm của notebook có sẵn và học tập thêm những kỹ thuật của người đạt giải nhất để xem những điểm khác biệt với notebook hiện tại của nhóm.

Cải thiện private score: 0.445 -> 0.459

Phân tích những điểm vượt trội của notebook top1: 0.482(private score) để học tập cách phân loại nhãn 3 và rút kinh nghiệm cho những cuộc thi sau.

IV. Những cải tiến đã áp dụng trong code để tăng được private score từ các notebook trước đó

1. Các cải tiến đã thử nghiệm

Loại bỏ các cột có tỷ lệ missing với các ngưỡng khác nhau(> 50%, 75%)

Loại bỏ các đặc trưng có độ tương quan dựa trên các ngưỡng khác nhau

Điền các giá trị missing bằng giá trị mean cho biến numerical và mode cho categorical

Sử dụng các encoding khác nhau cho các cột object, category (One-Hot Encoding, Label Encoding, Ordinal Encoding)

Xử lý các giá trị outlier bằng cách thay chúng bằng giá trị trung vị (median) hoặc RobustScalser cho các cột có phấn bố biến động cao

Dùng PCA để giảm chiều dữ liệu

Khai thác thêm thông tin từ các cột quan trọng dựa trên bộ lọc về thời gian (day, night), và trích xuất đặc trưng bằng việc tính toán các giá trị thống kê: mean, std, max, min …

Thử nghiệm với model: LightGBM, XGBoost, Catboost. Kỹ thuật tổng hợp (ensemble) được sử dụng để cải thiện độ chính xác

Sử dụng kappa score(weighted quadratic) được sử đụng để đánh giá hiệu năng của dự đoán tổ hợp. Kèm theo đó là confusion matrix để trực quan hoá phân loại các lớp

2. Cải tiến nổi bật: áp dụng bộ lọc day, night lên các đặc trưng quan trọng để trích xuất thông tin và tạo thêm một số đặc trưng dựa trên những đặc trưng đó. (cải thiện 0.445->0.454 private score)

Dữ liệu time_of_day được biểu diễn dưới dạng %H:%M:%S.%9f, sau khi được chuẩn hoá về dạng giờ trong ngày bằng công thức dưới đây và quy ước lại ta thu được kết quả:

```
df["hours"] = df["time_of_day"] // (3_600 * 1_000_000_000)
```

Thời gian	Quy ước	Điều kiện
Đêm	10 PM - 5 AM	`hours >= 22` hoặc `hours <= 5`
Ngày	7 AM - 8 PM	`7 <= hours <= 20`
Không mặt nạ	Toàn bộ dữ liệu	Không áp dụng điều kiện

Tương tự chúng ta sẽ quy ước đặc trưng last_week để làm bộ lọc cho các giá trị quan trọng như: keys = ["enmo", "anglez", "light", "enmo_weekend", "anglez_weekend"]

Tiếp theo đó sẽ lọc các giá trị emo, anglez, .. trong bảng key theo 4 bộ lọc thời gian là mask = [no_mask, day, nigh, last_week] và mỗi giá trị sẽ được tính trên các thuộc tính sau : mean, std, max, min, kurtosis, skew, diff_mean, diff_std, diff_90th, diff_10th(tổng cộng có 10 giá trị)

Với 5 cột keys, 4 mask và 10 giá trị thống kê khi đó chúng ta sẽ có thêm :
5x4x10 = 200 (đặc trưng tập trung vào khai thác chỉ số enmo, anglez, light)

Chọn các đặc trưng ở keys để lọc: enmo
(Euclidean Norm Minus One)
Mô tả: enmo là một phép đo gia tốc được tính toán từ các giá trị X, Y, và Z thu thập từ gia tốc kế trên đồng hồ đeo tay.

Ý nghĩa:

- enmo đại diện cho cường độ chuyển động của người đeo thiết bị:
- Nếu giá trị enmo = 0, điều này cho thấy không có chuyển động, ví dụ như khi người đeo đứng yên.
- Các giá trị lớn hơn biểu thị mức độ chuyển động tăng dần (chạy, đi bộ, v.v.).

anglez (Angle-Z)

Mô tả: anglez là góc đo được từ accelerometer, được tính toán dựa trên các trục gia tốc X, Y, và Z.

Ý nghĩa:

- anglez đại diện cho góc nghiêng của cánh tay người đeo thiết bị so với mặt phẳng nằm ngang.
- Giá trị anglez nhỏ: Cánh tay hướng về phía ngang.
- Giá trị anglez lớn: Cánh tay dựng đứng hoặc hoạt động theo trục dọc.

Light: Ánh sáng xung quanh cố thể ảnh hưởng đến thói quen ngủ của đối tượng và do đó là một công cụ hữu ích trong việc phân tích nhịp sinh học.

Tại sao áp dụng bộ lọc day, night để trích xuất đặc trưng từ enmo và anglez, light giúp cải thiện mô hình

➔ **Nắm bắt những hoạt động thay đổi theo thời gian:**

Hành vi của người dùng trên internet và hoạt động thể chất thường khác nhau giữa ban ngày và ban đêm.

Ban đêm: Người dùng có xu hướng sử dụng internet để giải trí, xem phim, hoặc chơi game nhiều hơn, điều này có thể liên quan đến việc sử dụng internet một cách có vấn đề(problematic use).

Ban ngày: Hoạt động liên quan đến công việc hoặc học tập chiếm ưu thế hơn.

Việc phân tách này giúp mô hình học được các mẫu đặc trưng từ mỗi thời điểm, làm tăng độ chính xác trong dự đoán.

Ví dụ: một người sử dụng internet vào buổi tối 3-4 tiếng thì có khả năng bị nghiện các trò chơi hơn là người sử dụng vào ban ngày vì đây là thời gian làm việc hoặc đi học.

➔ **Tập trung vào các đặc trưng quan trọng và giảm nhiễu trong dữ liệu:**

Các đặc trưng như ENMO (gia tốc) và AngleZ (góc nghiêng) thay đổi đáng kể giữa ban ngày (khi người dùng hoạt động) và ban đêm (khi nghỉ ngơi). Việc sử dụng bộ lọc Day/Night cho phép mô hình hiểu rõ hơn ngữ cảnh hoạt động, từ đó cải thiện khả năng phân loại.

Ví dụ: mô hình có thể dễ dàng hiểu và học được các mối quan hệ thực tế trong dữ liệu. Enmo cao + anglez dao động mạnh -> chạy hoặc nhảy. Enno thấp + anglez cố định. -> đứng yên hoặc ngồi. Từ đó đưa ra những dự đoán về trạng thái hoạt động tốt hơn.

Khi áp dụng bộ lọc Day/Night, dữ liệu được phân tách thành các khoảng thời gian cụ thể, giảm sự chồng chéo giữa các hành vi không liên quan. Điều này giúp mô hình không bị phân tâm bởi các tín hiệu gây nhiễu trong dữ liệu gốc

Thử nghiệm với không xài bộ lọc day, night thì được kết quả 0.445 (https://www.kaggle.com/code/quannguyn12/0-463-private?scriptVersionId=216621586) -> có bộ lọc day, night thu được kết quả 0.454 (https://www.kaggle.com/code/dattran0509/m-nh-ph-c-quang-tmain?scriptVersionId=199382411)

2. Tiếp tục hiệu chỉnh tham các tham số của mô hình để đạt kết quả cao hơn (từ 0.454-> 0.459 private score)

Hiệu chỉnh các tham số của mô hình:

Lgb_prarams: max_depth: 10 -> 12

Xgb_prarams: max_depth: 4 ->6

Cat_params: max_depth: 8 -> 10

Bới vì dữ liệu ở trong dự án lần rất phức tạp nên lựa chọn việc tăng độ sâu của các mô hình là một lựa chọn hợp lý, nhưng khả năng overfit cũng tăng lên theo đó. Vì vậy, trọng số l2 regularization cũng được tăng ở các mô hình để tránh mô hình bị overfit quá nhiều.

Kết quả cho thấy sau khi sửa đổi mô hình đã học tốt hơn cho kết quả tốt ở trên private test (từ 0.454->0.459)(https://www.kaggle.com/code/dattran0509/m-nh-ph-c-quang-tmain?scriptVersionId=199598465).

Khó khăn gặp phải: có thể đo phần xử lý dữ liệu chưa tốt nên dẫn đến dữ liệu vẫn còn rất nhiều nhiễu và chưa tối ưu được những đặc trưng quan trọng khiến cho mô hình không dự đoán được nhãn 3.

V. Tham khảo notebook của top1 để có thể cải tiến phân loại được nhãn 3 của mô hình.

1. **Data cleaning.**

- Thực hiện loại bỏ những giá trị không hợp lý trong một số cột như là: đối với cột **BIA-BIA_Fat** loại bỏ các giá trị lớn hơn 60 và nhỏ hơn 5 (dựa trên thông tin thực tế) thay bằng giá trị NaN. Tương tự cho các đặc trưng khác như: **BIA-BIA_BMR, BIABIA_DEE,...**

2. **Feature engineering.**

- Đối với time feature thì tương tự với notebook trên thì notebook này vẫn sử dụng lọc theo **day và night (nhưng sử dụng thêm một key khác là battery_volatage)**.
- Điểm khác biệt là note book này tạo ra thêm 2 đặc trưng:
  + **df["non-wear_flag"].mean()**
    => Đặc trưng này giúp tính tỉ lệ phần trăm đồng hồ không được đeo. Vì những điểm dữ liệu ở khoảng thời gian không đeo đồng hồ thì được điền vô chứ không phải dữ liệu thực tế. Việc làm như vậy giúp mô hình học được những điểm dữ liệu này tốt hơn.
  + **df["enmo"][df["enmo"] >= 0.05].sum()**

=> Đặc trưng này tính tổng chuyển động lớn hơn hoặc bằng 0.05. Giá trị **enmo** thể hiện mức độ chuyển động, ngưỡng này giúp loại bỏ những ngưỡng đáng kể.

+ Phần keys dùng để lọc theo masks thì notebook này sử dụng thêm đặc trưng là **battery_voltage.**

=> Việc sử dụng đặc trưng này giúp mô hình học được các hành vi hoặc trạng thái cửa người dùng như là mức pin có thể giảm dần theo thời gian. Điều này có thể cung cấp thông tin ngầm về thời gian hoạt động của người dùng hoặc cách sử dụng thiết bị. Ví dụ: Người dùng có thể không sạc thiết bị khi ngủ hoặc khi không hoạt động.

- Với những đặc trưng khác notebook này loại bỏ các đặc trưng không cần thiết như **Season** và tạo thêm các đặc trưng chuẩn hóa như là **BMI_mean_norm** được tính bằng cách lấy giá trị hiện tại chia cho giá trị BMI trung bình trên nhóm tuổi đó(gồm các nhóm tuổi như là 5, 6, 7, 8,...).Giá trị BMI trung bình trên nhóm tuổi được lấy dựa trên dữ liệu thực tế. Tương tự với các đặc trưng như là **FFM_norm, ICW_ECW_norm, ….**

=> Việc làm như vậy giúp mô hình hiểu được sự chênh lệch giữa cái chỉ số của người được khảo sát với giá trị trung bình. Ví dụ: **DEE_BMR_norm > 1**: Cá nhân này tiêu thụ năng lượng ngoài mức cơ bản cao hơn mức trung bình → Hoạt động thể chất cao hơn bình thường.

- Phân nhóm theo phần vị (quantile binning) được áp dụng cho một phần lớn các đặc trưng để xử lý nhiễu.
- Dùng **PCA** để giảm chiều dữ liệu xuống còn 15. Giúp loại bỏ nhiễu và dư thừa dữ liệu.

3. Imputation.

- Với những đặc trưng đủ điều kiện: Notebook này thay vì điền những giá trị thiếu bằng mean hay mode thì lại dùng và mô hình lasso để dự đoán điền giá trị bị thiếu dựa trên các đặc trưng khác cho các đặc trưng mà đủ các điều kiện như là: Tỷ lệ giá trị khuyết trong đặc trưng đó nhỏ hơn hoặc bằng 40% và số lượng mẫu trong đặc trưng hợp lệ đó phải lớn hơn hoặc bằng 1.

=> Dự đoán giá trị khuyết dựa trên mối quan hệ giữa các đặc trưng, giúp duy trì tính đồng nhất và cấu trúc dữ liệu. Chọn lọc đặc trưng quan trọng, giảm ảnh hưởng của nhiễu nhờ cơ chế L1 regularization của lasso.

- Với những đặc trưng không đủ điều kiện thì điền bằng mean.

4. Model selection.

- Sử dụng 5 model (do có 4 nhãn nên dùng 5 model sẽ hợp lý hơn trong việc vote):

+ LGBMRegressor.

+ 2 model XGBoost Regressors.(Điểm kappa của xgboost không quá cao hay quá thập nằm ở giữa nên việc sử dụng 2 model XGBoost giúp tổng quát quá kết quả dự đoán)

+ CatBoostRegressor.

+ ExtraTreesRegressor.

5. Target, Cross-Validation, Sample Weights.

- **Biến mục tiêu:** Thay vì sử dụng nhãn **sii** gốc, chuyển sang dùng điểm số **PCIATPCIAT_Total** và chuyển đổi kết quả dự đoán về nhãn **sii.** Giúp chuyển bài toán phân loại sang hồi quy -> bài toán hợp lí hơn so với việc sử dụng regression cho bài toán phân loại.
- **Phân phối và trọng số:** Phân phối mục tiêu chứa nhiều giá trị 0, do đó dùng 2 cách tiếp cận: sử dụng trọng số mẫu và các mục tiêu hồi quy thay thế như **Tweedie**. Các giá trị thiểu số được đặt trọng số cao hơn so với các giá trị đa số -> nhờ đây mà có thể dự đoán được nhãn 3.
- **Cross-Validation:** Sử dụng **Stratified KFold** với **10 fold** dựa trên các khoảng (bins), thường xuyên thay đổi seed để đảm bảo ổn định. Kết hợp nhiều seed để đánh giá thêm qua Leaderboard công khai, và tránh tối ưu hóa quá mức cho Leaderboard.

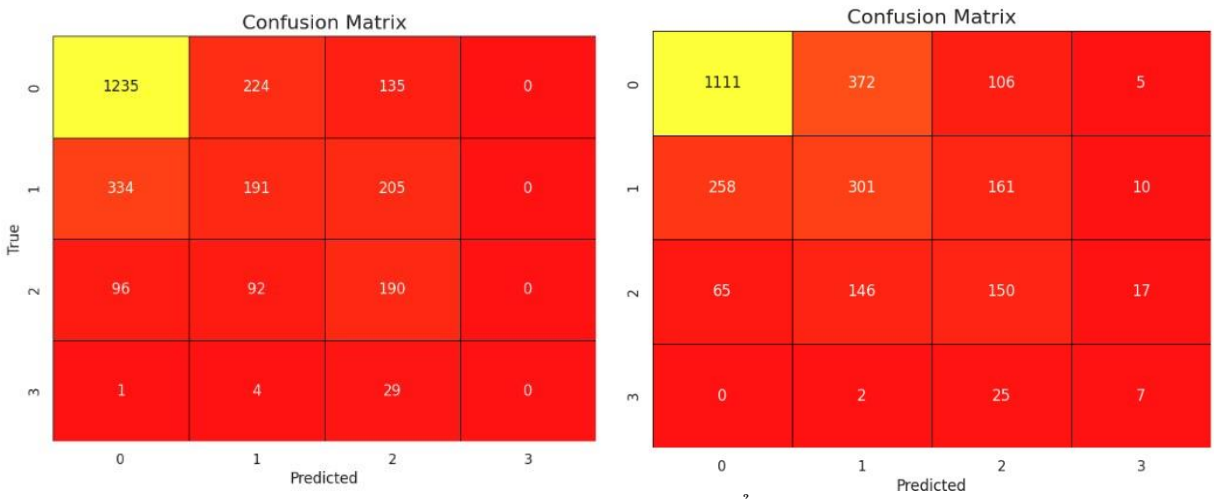
6. Parameter tuning, Feature selection.

- Điểm khá hay ở notebook này nằm ở việc sử dụng **Repeated Stratified KFold** vì việc sử dụng cross-validation thông thường dẫn đến kết quả không ổn định do sự biến động trong dữ liệu và nhiễu. **Repeated Stratified KFold** lặp qua các seed khác nhau và chọn ra các parameter mà có **Kappa Score** cao nhất.Sử dụng **Repeated Stratified KFold**, lặp lại từ **10 đến 20 lần** trong quá trình tối ưu tham số.

=> Mang lại kết quả ổn định và đáng tin cậy hơn bằng cách trung bình hóa qua nhiều lần chia dữ liệu, giảm sự biến động trong điểm số CV.

- Chọn lọc đặc trưng: chọn lọc đặc trưng được thực hiện thủ công dựa trên phân tích tầm quan trọng của đặc trưng (feature importance) do từng mô hình cung cấp.

=> Tập dữ liệu được giảm xuống còn **39 đặc trưng**, giúp đơn giản hóa mô hình và cải thiện khả năng tổng quát hóa.



- Mô hình sau với những kỹ thuật xử lý trên đã tổng quát hoá hơn không bị fit với dữ liệu train quá nhiều dẫn đến không phân loại được các dữ liệu lạ trong tập test.