Child Mind Institute — Problematic Internet Use TEAM: QuynhBupBe

1st Tran Tuan Phong
Machine learning INT3405E
22028081@vnu.edu.vn

2nd Luu Khai Hung
Machine learning INT3405E
22028288@vnu.edu.vn

3rd Nguyen Manh Quynh
Machine learning INT3405E
22028241@vnu.edu.vn

I. Introduction

A. Summary of The Competition

This competition challenges you to develop a predictive model capable of analyzing children's physical activity data to detect early indicators of problematic internet and technology use. This will enable prompt interventions aimed at promoting healthier digital habits.

B. Data Description

The competition data consists of two sources: parquet files containing accelerometer (actigraphy) series and CSV files that include the remaining tabular data. The aim of this competition is to predict a participant's Severity Impairment Index (`sii`), which measures the level of problematic internet use among children and adolescents from 5 to 22 years old, using physical activity data and internet usage behavior data.
The target `sii` is derived from `PCIAT-PCIAT_Total`, the sum of scores from 20 Parent-Child Internet Addiction Test, with each question scored from 0 to 5.
The `sii` is defined as:

- 0: None (`PCIAT-PCIAT_Total` from 0 to 30)
- 1: Mild (`PCIAT-PCIAT_Total` from 31 to 49)
- 2: Moderate (`PCIAT-PCIAT_Total` from 50 to 79)
- 0: Severe (`PCIAT-PCIAT_Total` 80 and more)

Our strategy is to treat the prediction of `sii` as a regression problem. The model predicts the target variable `sii` as a continuous value, which is then rounded to the nearest integer to classify the severity level.

C. Data Exploration

1) Data Preview: The training dataset contains 3960 samples and 80 features, excluding the `id` column and the target variable `sii`. There are 20 test samples provided. According to the documentation, the full test set consists of approximately 3800 instances.

2) Missing Columns: All data from the Parent-Child Internet Addiction Test (PCIAT), including the `PCIAT-PCIAT_Total` feature, are excluded from the test dataset, which is used to predict the target variable `sii`. Therefore, we will directly predict the variable `sii` using all other features, excluding the PCIAT results.

3) Missing Values: All features in the training dataset, except for the `id` and the three basic demographic features contain a significant amount of missing data, as shown in Fig. 1. Since one-third of the data in the target feature is missing, we will make predictions only using entries that have complete data for this feature. After removing the entries with missing values in the feature `id`, we are left with 2,736 entries. In Fig. 2, there are still some PCIAT features with missing value but the entries still contain values for both `PCIAT-PCIAT_Total` and `sii`.

4) Categorical Variables: After excluding all PCIAT columns, there will be 58 columns used for prediction, of which 48 are continuous variables and 10 are categorical variables. Every categorical feature represents the time period during which each data group was collected. In Fig. 3, we can see that the distribution of the participation seasons for each test is relatively balanced.

5) Target Variables: The distribution of `sii` shows a heavily imbalanced dataset, with the majority of values concentrated at `sii` = 0 and a very small amount at `sii` = 3. As the severity level of sii increases, the number of instances decreases, with the lowest count observed at `sii` = 3.

6) Demographic: The amount of male participants is higher than that of female participants in the training dataset. The distribution of SII for both genders is quite similar and aligns closely with the overall `sii` distribution.
The ages of the participants range from 5 to 22 years old. The classification of `sii` by age groups shows that in the younger group (ages 5–10), `sii` = 3 does not occur, and `sii` = 0 accounts for the majority. SII = 3 begins to appear and becomes prominent in the age group 11–17 but gradually decreases in the age group 18–22. In general, `sii` tends to increase as age rises.

7) Children's Global Assessment Scale: The Children's Global Assessment Scale (CGAS) is a 1–100 scale used to assess the overall psychological, social, and behavioral functioning of children and adolescents. This feature has about 14.4% missing values. There is one extreme value outlier (`CGAS-CGAS_Score` = 999), which is obviously an error. However, it was removed when we excluded the entries with missing `sii` values. As the severity of `sii` increases, the

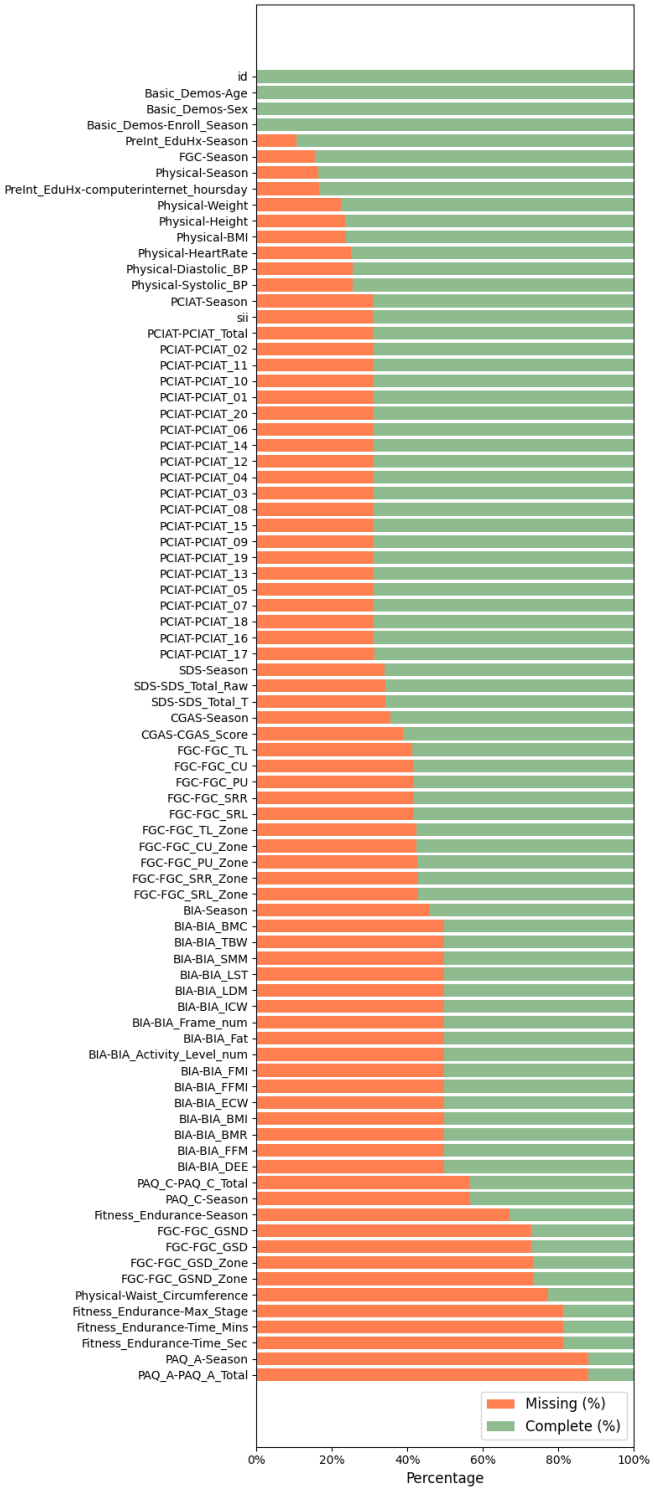

Fig. 1: Missing values over the dataset

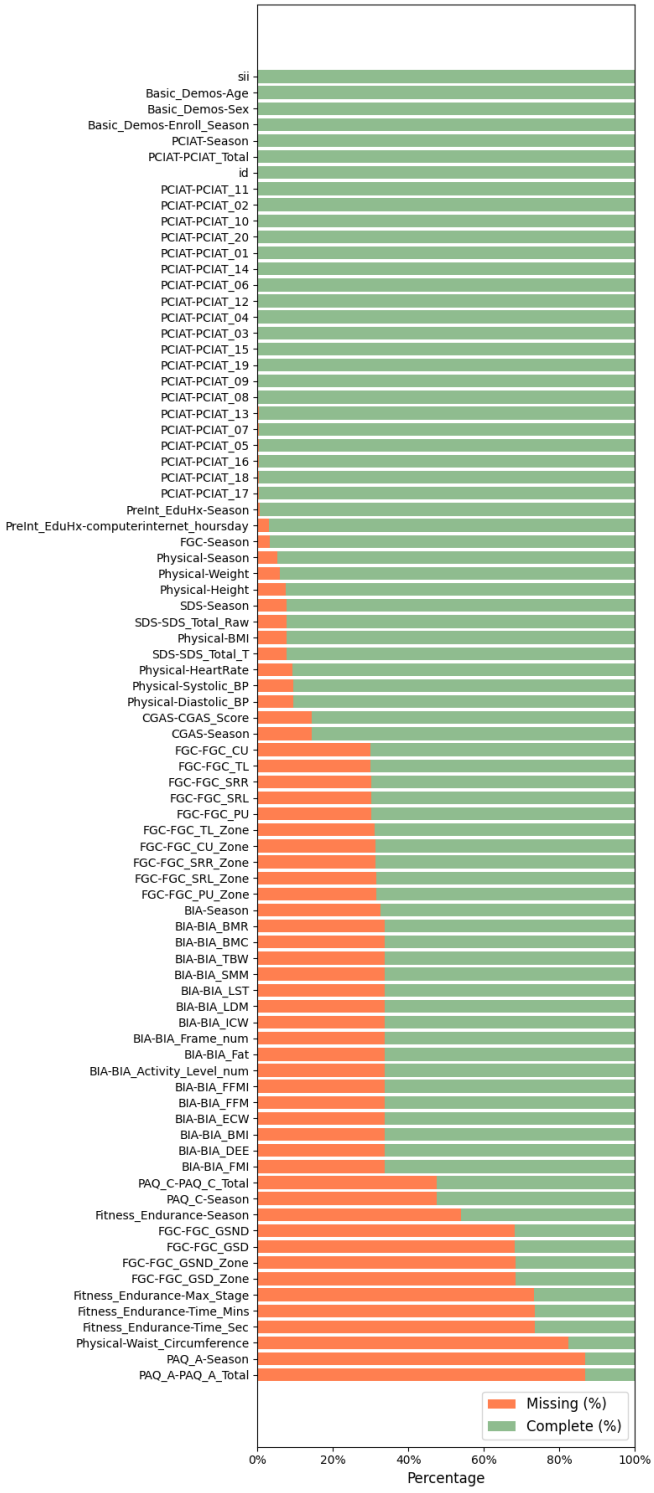

Fig. 2: Missing values after removing missing sii from the dataset

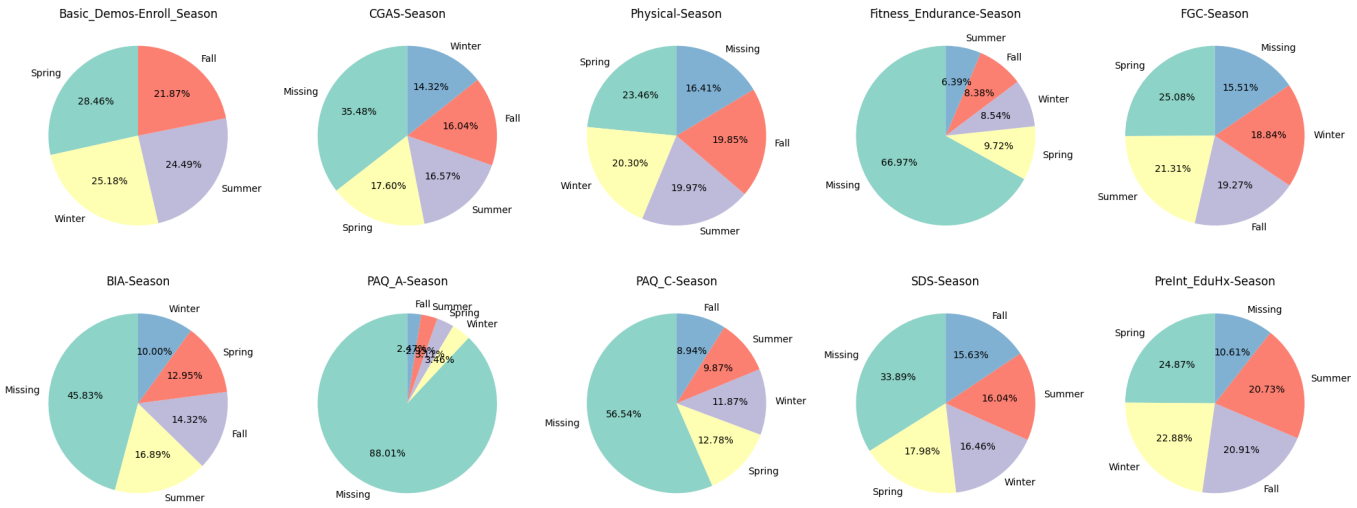

Fig. 3: Seasonal data distribution

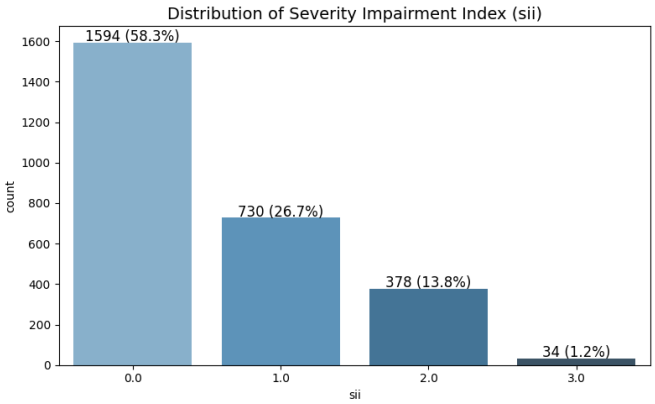

Fig. 4: Distribution of the target variable sii

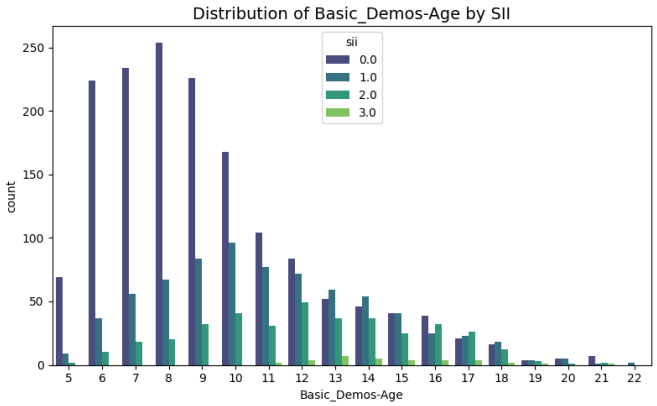

Fig. 6: Distribution of the target variable sii by Age group

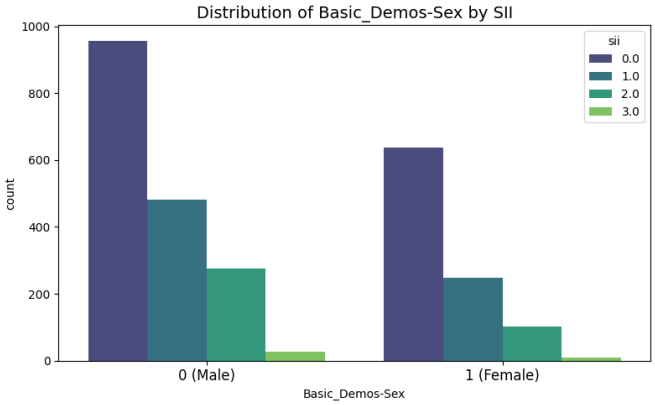

Fig. 5: Distribution of the target variable sii by Sex group

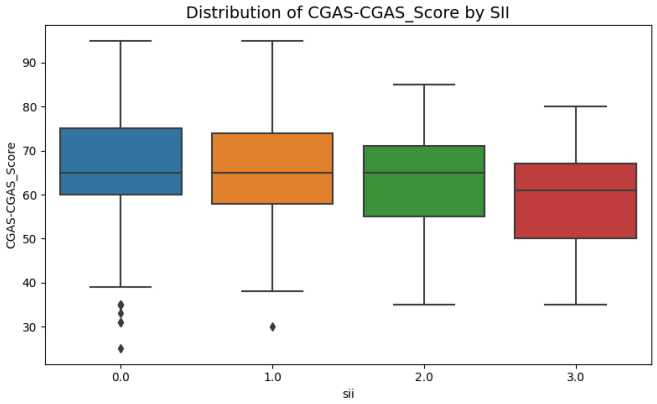

Fig. 7: Distribution of CGAS_Score by sii

mean of `CGAS-CGAS_Score` tends to decrease.

8) Physical Measures: The Physical Waist Circumference feature has the highest missing data in the dataset, with approximately 82% missing values. The other features in this group have missing values ranging from 5.9% to 9.4%.. Some data points have measurement results of 0, and some data have extreme large values. When distributed by age, some data points are not properly distributed, such as children being overweight or over the expected height for their age. Some data points may be errors, but some data points could be due to participants have gigantism or another disorder related to the growth hormone.

With the heart rate data, there are some data points with values lower than the human standard, and some have systolic BP lower than diastolic BP (which should normally be higher).

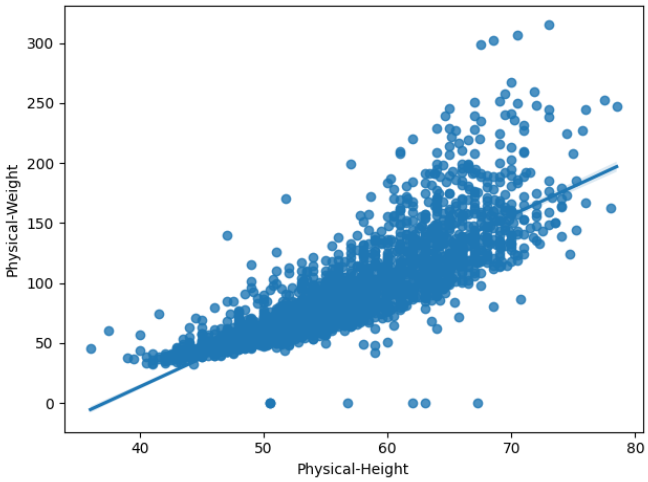

Fig. 8: Correlation between Height and Weight

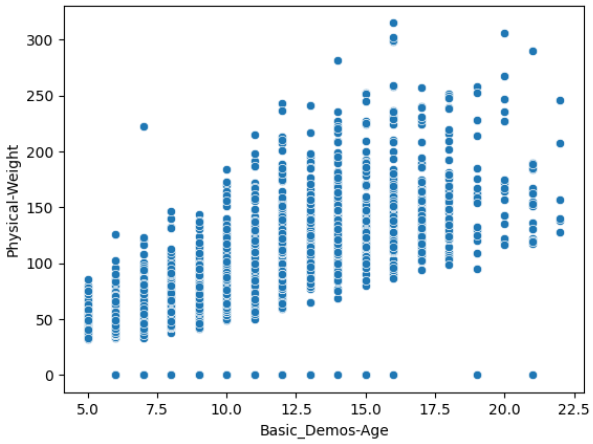

Fig. 9: Weight distribution by Age group

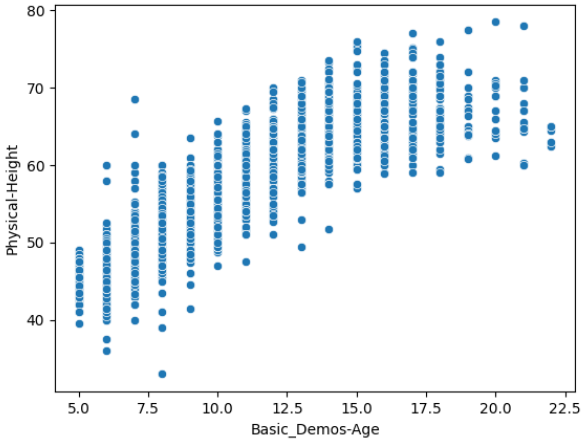

Fig. 10: Height distribution by Age group

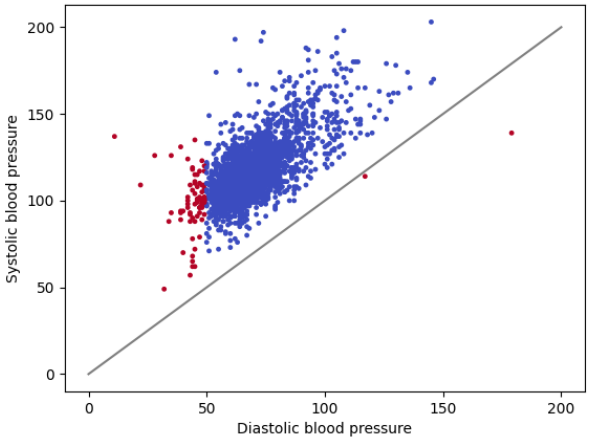

Fig. 11: Correlation between systolic blood pressure and diastolic pressure - Red data points indicate unusual values

9) FitnessGram Vitals and Treadmill: This feature group has approximately 73.3% missing values, which is quite high. Age range for participants from 6 to 12 years old. Participants outside this age range will have no value, which is why there are many missing values. On average, participants reached stage 5 in the endurance test. Some participants failed to complete the first stage (min = 0), possibly due to data errors. A small number of participants aged 7-8 exhibited exceptionally high endurance.

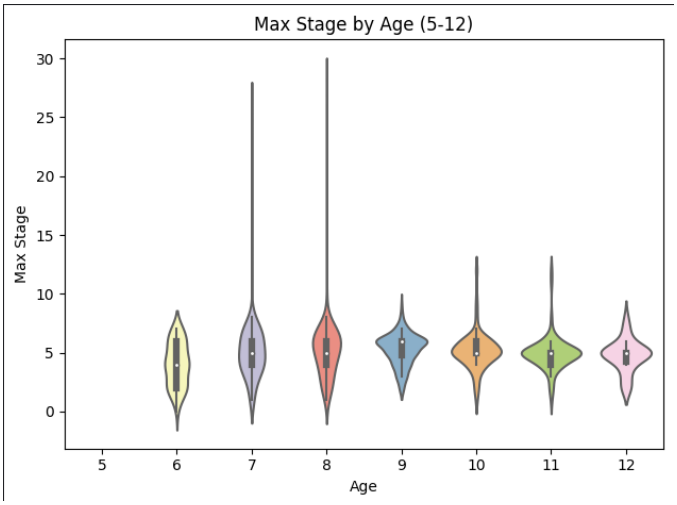

Fig. 12: Distribution of Max Stage across Ages from 5 to 12

10) FitnessGram Child: This feature group has a relatively high percentage of missing values, ranging from 29.8% to 68.4%. There is an overlap between the metrics and their corresponding zones. However, when the distribution is analyzed by gender and age, the separation between zones becomes clearer. This suggests that different age groups and genders may have different zone ranges. However, some noisy data show lower scores being classified into higher zones, while others with higher scores are placed in lower zones. This can clearly seen in the Grip Strength test with its three zone ranges.

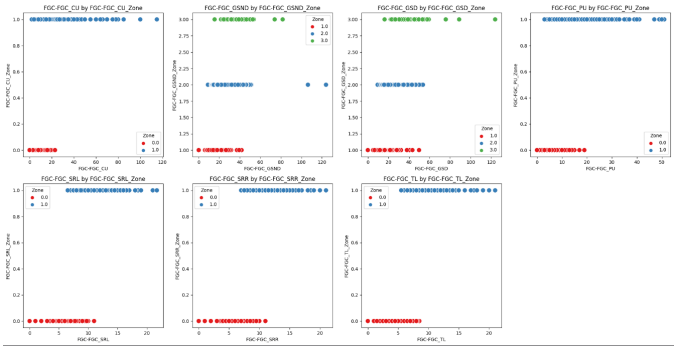

Fig. 13: Distribution of FitnessGram Child by Zone

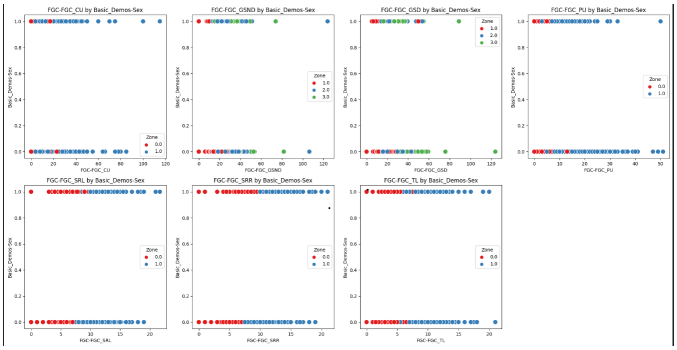

Fig. 14: Distribution of FitnessGram Child by Sex

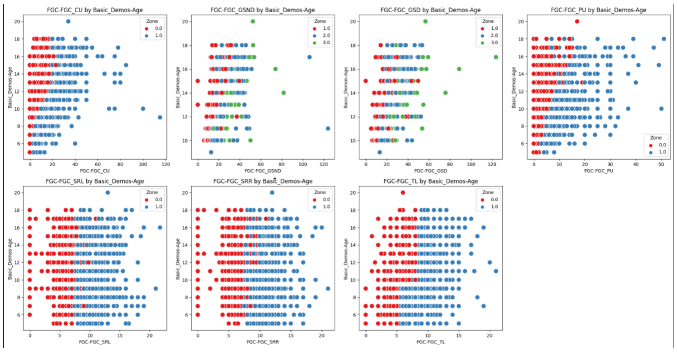

Fig. 15: Distribution of FitnessGram Child by Age

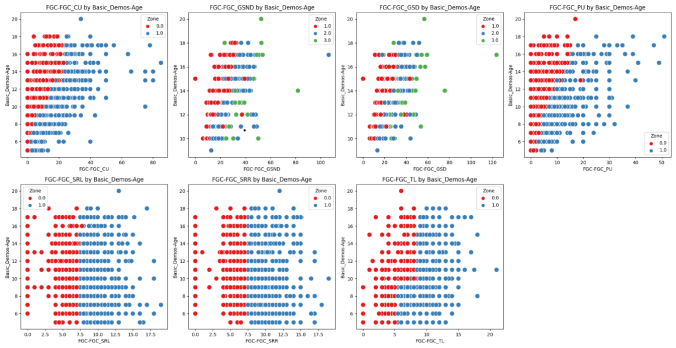

Fig. 16: Distribution of FitnessGram Child by Age of Male Group

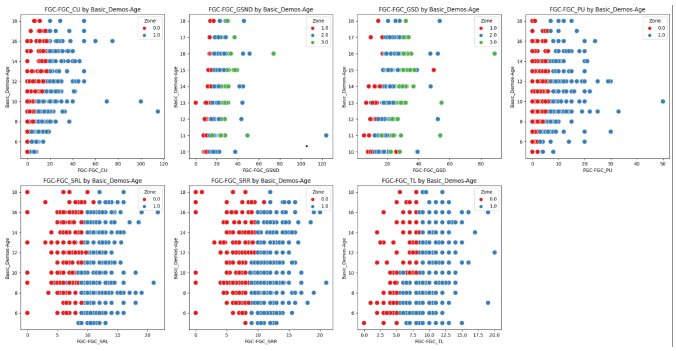

Fig. 17: Distribution of FitnessGram Child by Age of Female Group

11) Bio-electric Impedance Analysis: Each feature in this group has 33.73% missing values. There are a lot of outliers in the numerical BIA group. Almost all features have many data points with extremely high values. There are data in some features such as `BIA-BIA_Fat` and `BIA-BIA_FMI` with negative values, while the `BIA-BIA_BMI` index has excessively small values. These features will not have high reliability in prediction due to the presence of outliers and unreasonable values.

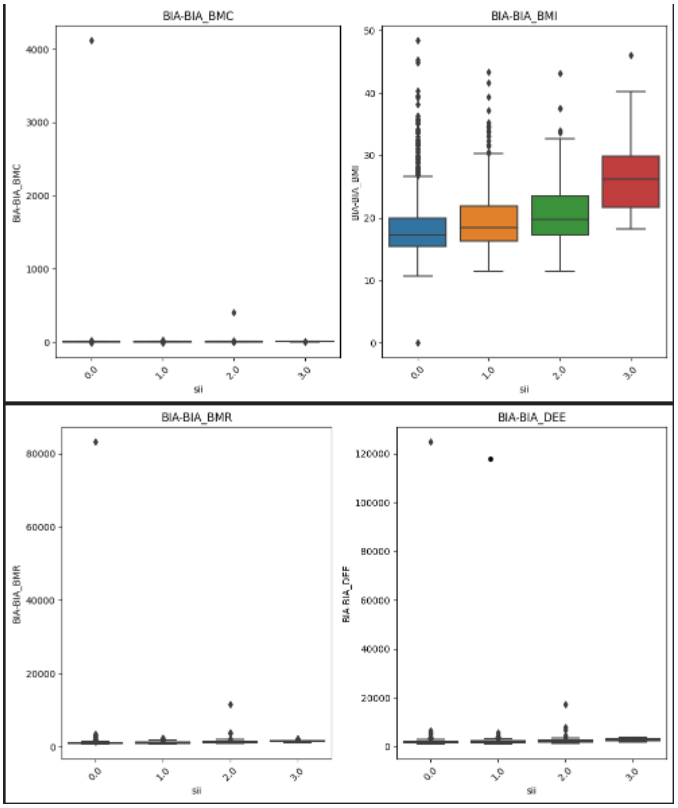

Fig. 18: Distribution of BIA Groups by sii

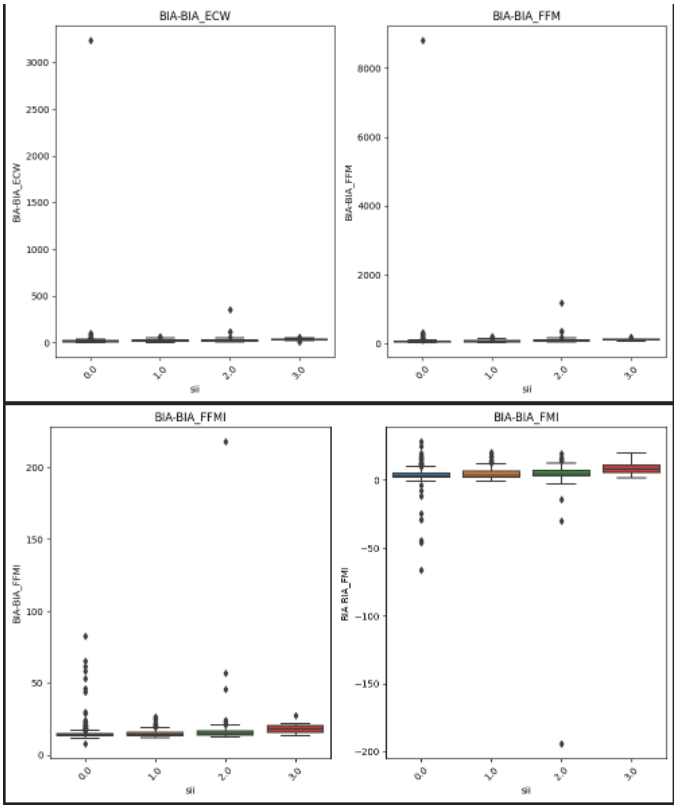

Fig. 19: Distribution of BIA Groups by sii

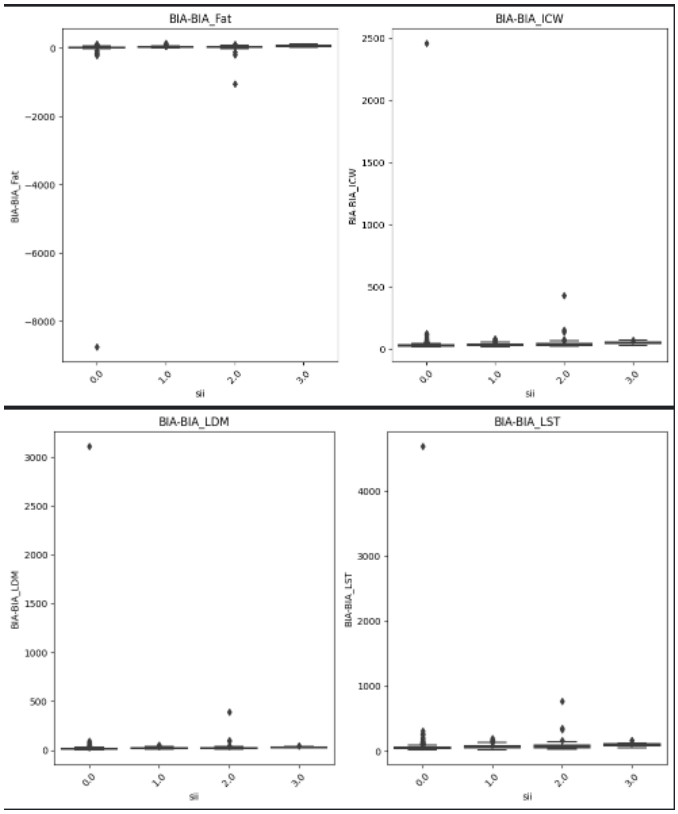

Fig. 20: Distribution of BIA Groups by sii

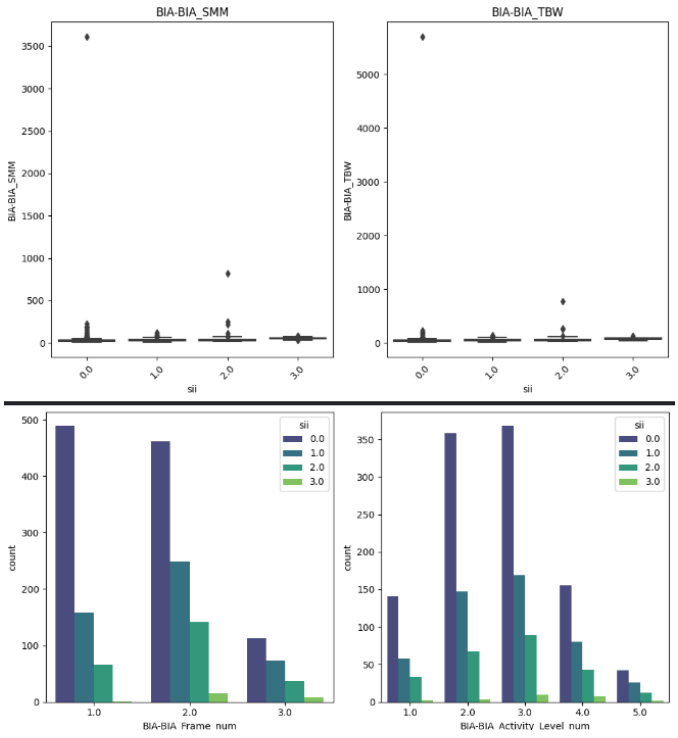

Fig. 21: Distribution of BIA Groups by sii

12) Physical Activity Questionnaire: Since not all participants will answer these questions, the amount of missing data

is relatively high, with 86.7% missing for the Adolescents question and 47.3% for the Children question. The age range for participants in the test for Children is from 7 to 14 years, while the age range for Adolescents is from 13 to 18 years. However, there is only a single entry which contain both values. For Adolescents, the median slightly decreases as the `sii` level increases, particularly noticeable at `sii` = 3. In contrast, the median for Children tends to increase as the `sii` level rises. Perhaps the influence of the internet reduces physical activity time in children more significantly than in adults.

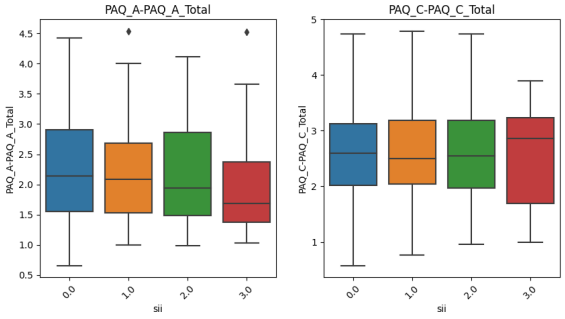

Fig. 22: Distribution of Physical Activity Questionnaire by sii

13) Sleep Disturbance Scale: The Sleep Disturbance Scale questionnaire generates a raw score ranging from 0 to 100, which is subsequently converted into a T-score. The T-score has a higher percentage of missing data than the raw score, with 7.7% missing compared to 7.6%. The severity of sleep disturbances, shows a median increase as `sii` levels rise. This suggests a possible relationship between sleep disturbances and internet usage.

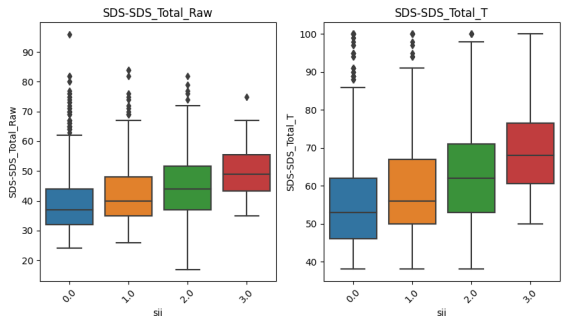

Fig. 23: Distribution of Sleep Disturbance Scale by sii

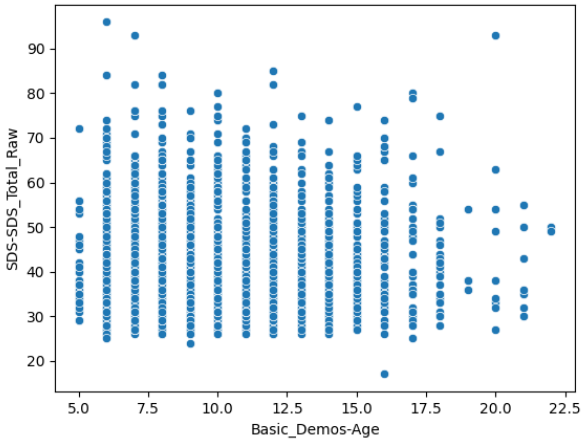

Fig. 24: Distribution of Sleep Disturbance Scale by Age group

14) Internet Use: This feature has a small amount of missing data, with 2.99% missing. This is an important feature directly related to the Problematic Internet Use task. The daily internet usage time increases with age, with the maximum value being more than 3 hours per day. With increasing internet usage time, the age group of internet users becomes more diverse, with a broader range of ages involved. Similarly, as internet usage increases, the `sii` level also tends to rise. `sii` = 0 is dominant among users with low internet usage, while `sii` = 2 and `sii` = 3 seem to appear more frequently among users with higher internet usage, despite a few exceptions.

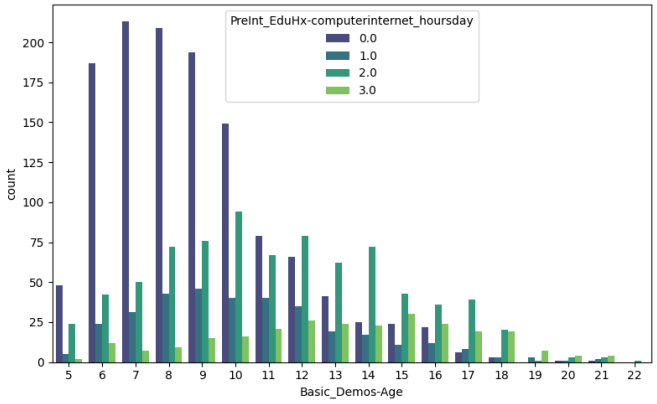

Fig. 25: Distribution of Internet Usage Time by Age group

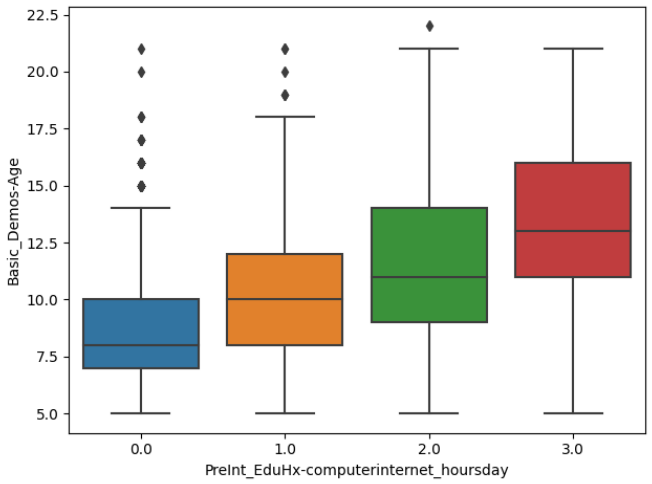

Fig. 26: Distribution of Internet Usage Time by Age group

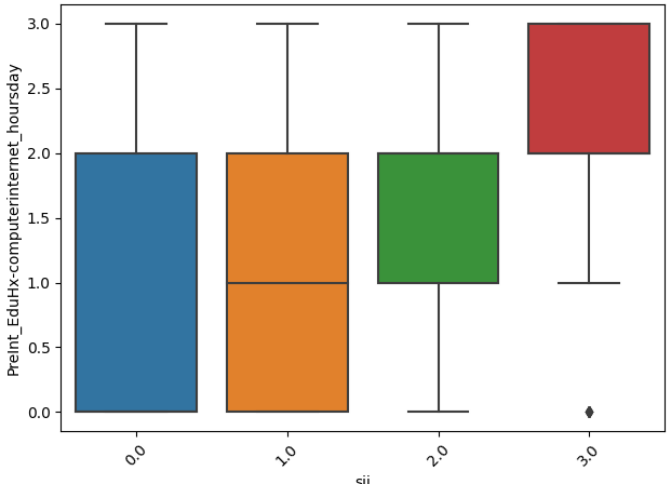

Fig. 27: Distribution of Internet Usage Time by sii

II. Solving the Problem

A. Choosing the Base Model

1) XGBoost OverView: XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for "Extreme Gradient Boosting" and has become one of the most popular and widely used machine learning algorithms due to its ability to handle large data sets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

2) Why XGBoost is chosen?: One of the key features of XGBoost is its efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing. Additionally, XGBoost has built-in support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time.

Data given in tabular form can be trained using tree-based models. Along with LGBM, XGBoost is one of the best tree-based models for working with tabular data, being fast to train, can be explained, and gives very good performance across the datasets.

B. Hyperparameter Tuning

Choosing a Learning Rate: Start by selecting a relatively high learning rate. A learning rate of 0.1 often works well, but generally, a range between 0.05 and 0.3 should be considered, depending on the specific problem. This will help in controlling how much the model adjusts in each boosting step.

Tuning Tree-Specific Parameters: After fixing the learning rate, we can tune various tree-specific parameters:

- `max_depth`: Controls the maximum depth of a tree. Increasing `max_depth` can make the model more complex, but excessive values might lead to overfitting.
- `min_child_weight`: Specifies the minimum sum of instance weights needed in a child node. Higher values of this parameter prevent the model from learning overly specific patterns that could lead to overfitting.
- `gamma`: Determines the minimum loss reduction required to make a split. A higher `gamma` value results in more conservative splits.
- `subsample`: Represents the fraction of samples used to fit each individual tree. Lower values introduce more randomness, which helps in preventing overfitting.
- `colsample_bytree`: Controls the fraction of features used for each tree. Like `subsample`, it introduces randomness to reduce overfitting.

Tuning Regularization Parameters: The regularization parameters `lambda` and `alpha` can be tuned to reduce model complexity and enhance performance:

- `lambda`: L2 regularization term on weights, which helps prevent overfitting by penalizing large coefficients.
- `alpha`: L1 regularization term on weights, which can help in feature selection by driving some coefficients to zero.

Lowering the Learning Rate: Once the parameters are tuned, we can reduce the learning rate further to achieve more precise updates. The optimal parameters from the previous step should be re-checked and adjusted for the new learning rate.

1) Choose learning rate: Initially, our team selects a learning rate of 0.04 and sets the number of estimators to 20,000: Learning Rate = 0.04, n_estimators = 20000

However, due to the high value of `n_estimators`, we reduce it to 2000 and increase the learning rate to 0.05: Learning Rate = 0.05, n_estimators = 2000

In subsequent iterations, we change the learning rate as follows: Learning Rate = 0.03, n_estimators = 2000 and then: Learning Rate = 0.01, n_estimators = 2000

Finally, after further testing, we decide to keep the learning rate at 0.05 and reduce `n_estimators` to 200: Learning Rate = 0.05, n_estimators = 200

2) Tuning tree-specific parameters: Tune max_depth and min_child_weight as they have highest impact on the model outcome At first we choose a larger `max_depth` at 10 and then slowly decrease it to 7 and finally stop at 6. The `subsample` and `colsample_bytree` are set at 0.8 each throughout the project.

3) Tuning regularization parameters: The next step is to apply regularization parameters to reduce overfitting. Higher values of alpha lead to more feature weights being set to zero, resulting in a simpler, more interpretable model. The `reg_alpha` set to 1 is a moderate value, striking a balance between reducing noise and retaining sufficient features for predictive power. The `reg_lambda` set to 5 which is high as the dataset has many features, and some may have large weights due to noisy data. A high `reg_lambda` helps to stabilize the model by reducing these large weights.

C. Data Loading and Preprocessing

1) Loading actigraphy series: Efficiently handling the actigraphy series is critical due to its size and complexity. Two helper functions facilitate this process:

$process\_file(filename, dirname)$

- **Purpose:** Loads a single parquet file in the given directory, computes descriptive statistics (mean, standard deviation, etc.) using `describe()`, and reshapes them into a single row vector. It returns these statistics along with the extracted `id`.

$load\_time\_series(dirname)$

- **Purpose:** Iterates over all IDs in a given directory, using a thread pool for faster parallel processing. For each file, it retrieves summarized statistics. It then combines all results into a single DataFrame with one row per `id` and multiple statistical features.

Advantages:

- **Efficiency:** Parallel processing significantly reduces the time required to process large volumes of time-series data.
- **Scalability:** The approach is scalable to datasets with numerous id directories.
- **Modularity:** Separation of concerns allows for easy maintenance and potential enhancements.

2) Dimensionality Reduction with AutoEncoder: Given the high dimensionality of the aggregated statistical features, dimensionality reduction is employed to capture essential patterns while mitigating computational complexity. Importantly, the AutoEncoder was applied exclusively to the actigraphy series data due to its inherent complexity and high dimensionality compared to other feature sets.

AutoEncoder Architecture:

- **Encoder:** Sequential layers reducing the input dimensionality from `input_dim` to `encoding_dim` through intermediate layers with ReLU activations.
- **Decoder:** Mirrors the encoder architecture, reconstructing the original input from the encoded representation using ReLU and Sigmoid activations.

Perform Training (`perform_autoencoder`):

- **Scaling:** Standardizes the data using `StandardScaler` to ensure efficient training.
- **Conversion to Tensor:** Transforms the scaled data into PyTorch tensors.
- **Training Loop:** Optimizes the AutoEncoder using Mean Squared Error (MSE) loss and the Adam optimizer over a specified number of epochs and batch size.
- **Encoding:** Post-training, the encoder transforms the data into a lower-dimensional representation, which is then converted back to a Pandas DataFrame with appropriately named columns (`Enc_1`, `Enc_2`, ...).

Advantages:

- **Feature Compression:** Reduces dimensionality while preserving critical information, facilitating faster and more efficient model training.
- **Noise Reduction:** Potentially filters out irrelevant variations in the data, enhancing model robustness.

3) Handling Missing Values: Effective handling of missing data is essential to maintain data integrity and model performance.

Imputation Strategy

- **K-Nearest Neighbors (KNN) Imputer:** Utilizes `KNNImputer` with `n_neighbors=5` to fill in missing values based on the similarity of neighboring samples.
- **Numerical Columns Focus:** Applies imputation exclusively to numerical columns (`int32`, `int64`, `float64`) to maintain consistency and relevance.

Post-Imputation Processing

- **Rounding Target Variable:** Ensures that the target variable `sii` is an integer by rounding the imputed values.
- **Preservation of Non-Numerical Columns:** Retains non-numeric columns without imputation, preserving categorical or identifier information.

Data Cleaning

- **Row Filtering:** Drops rows in the training set with excessive missing values (`thresh=10`), maintaining data quality.
- **Infinite Values Handling:** Replaces any infinite values in the training set with `NaN` to prevent computational errors during modeling.

Advantages:

- **Preservation of Data Integrity:** KNN imputation leverages existing data patterns to fill missing values, reducing bias.
- **Consistency Across Datasets:** Ensures that both training and testing data undergo identical preprocessing steps.

D. Training and Validation

A systematic training and evaluation pipeline was established to ensure model reliability and performance optimization.

1) Cross-Validation: Cross-validation is used during training to evaluate the model's performance and ensure that it generalizes well to unseen data. In this approach, we utilize 5-fold cross-validation with a random state of 42 to maintain consistency and reproducibility across different versions.

- **Data Splitting:** Divided the dataset into training and validation sets within each fold.
- **Model Cloning:** Created independent instances of the ensemble model for each fold to prevent data leakage.
- **Training:** Trained the model on the training subset.
- **Prediction:** Generated Out-of-fold (OOF) predictions for validation data and accumulated test set predictions across folds.
- **Performance Logging:** Recorded Quadratic Weighted Kappa (QWK) scores for both training and validation sets per fold.

2) Performance Metrics: The primary metric for evaluation was the Quadratic Weighted Kappa (QWK), a measure suitable for ordinal classification tasks. QWK assesses the agreement between predicted and true classes, accounting for the degree of disagreement.

3) Threshold Optimization: Given that the target variable (sii) is categorical, continuous predictions required thresholding to map them into discrete classes. An optimization process was employed:

- **Initial Thresholds:** Started with predefined thresholds [0.5, 1.5, 2.5].
- **Optimization Objective:** Utilized the Nelder-Mead method to adjust thresholds, maximizing the QWK score.
- **Validation:** Applied the optimized thresholds to both Out-of-Fold (OOF) and test set predictions, enhancing classification accuracy.

4) Results: The training process have the following performance metrics:

- **Mean Training QWK:** 0.9074
- **Mean Validation QWK:** 0.5077
- **Optimized QWK:** 0.544
- **Public score:** 0.437

We can see that the model is over-fitted, a common issue with tree-based models like XGBoost.

E. Improving based model

To improve the base model, we utilize **Ensemble Learning** by combining the strengths of XGBoost with other powerful models such as LightGBM (LGBM), CatBoost, and TabNet. The ensemble leverages the unique capabilities of each model:

- **XGBoost:** Known for its flexibility and ability to handle complex interactions in the data.
- **LightGBM (LGBM):** Efficient and scalable, particularly for large datasets with high dimensionality.
- **CatBoost:** Specialized in handling categorical features and providing robust performance with minimal overfitting.
- **TabNet:** A deep learning model designed for tabular data, employing sequential attention to enhance feature selection and interpretability.

The ensemble is constructed using a weighted `VotingRegressor`, where the predictions of all models are combined to produce a final output. This approach helps to mitigate overfitting by leveraging the diversity of the individual models and improving the overall generalization of the system.

F. Further Enhancements with Alternate Ensembles

The ensemble model also exhibited signs of overfitting with results:

- **Mean Training QWK:** 0.7033
- **Mean Validation QWK:** 0.4458
- **Optimized QWK:** 0.520
- **Public score:** 0.473

To address this, we designed two additional ensemble models with varying voting strategies and preprocessing techniques:

1) Ensemble Model 2:

- **Voting Strategy:** Uses a weighted voting approach combining predictions from **XGBoost**, **LightGBM (LGBM)**, and **CatBoost**. This simpler configuration focuses on balancing speed and accuracy by excluding more complex models like TabNet.
- **Preprocessing Datas:** Only utilize actigraphy and tabular data without doing noise reduction or filling missing data with KNN but fill them with the mean of its feature.

2) Ensemble Model 3:

- **Voting Strategy:** Utilizes a weighted voting scheme incorporating **XGBoost**, **LightGBM (LGBM)**, **CatBoost**, along with two additional models: **Random Forest** and **Gradient Boosting**. The inclusion of these models adds diversity to the ensemble, enhancing robustness against overfitting.
- **Preprocessing Datas:** Only utilize actigraphy and tabular data without doing noise reduction or filling missing data with KNN but fill them with the mean of its feature.

These alternative ensembles aim to reduce overfitting further and improve the generalization of the predictions by experimenting with diverse strategies and techniques.

III. Feature engineering

This section is part of the data preprocessing pipeline for the first ensemble model, but given its level of detail, we believe it deserves to be explained separately.

1) Physical Activity Questionnaire: From the data, we observe that the features `PAQ_C-PAQ_C_Total` and `PAQ_A-PAQ_A_Total` represent similar measurements but are separated by age groups. However, there is only one sample where both features contain data. To consolidate these features into a single column, we propose the following merging strategy:

- For samples where only one of the two features has data, use the available value directly.

- For the single sample where both features have data, prioritize `PAQ_C-PAQ_C_Total` by filling its value into the new column.
- Considering that the two features collectively miss a large part of their data (*86.7% and 47.37%*), this merging is justified to improve data coverage.

2) Physical Activity Questionnaire: As previously noted, the BIA data group often contains a significant portion of missing data (approximately 33%) and exhibits substantial noise. To address this, we have decided that it is best to replace features that can be derived through calculation. These features include `BIA-BIA_Fat`, `BIA-BIA_BMI`, `BIA-BIA_BMR`, and `BIA-BIA_DEE`.

- For `BIA-BIA_Fat`, we drop this column and will be recalculated as:

$$\text{Fat (\%)} = \frac{\text{Fat Weight (kg)}}{\text{Total Weight (kg)}} \times 100$$

- For `BIA-BIA_BMI`, we drop this column as there is another column with better data coverage that already provides BMI information.
- For `BIA-BIA_BMR`, we use the Schofield Equation, calculated as:

$$\text{BMR} = \begin{cases} 88.362 + 13.397 \times \text{Weight (kg)} \\ +4.799 \times \text{Height (cm)} \\ -5 \times \text{Age (years)} & \text{(for males)} \\ 447.593 + 9.247 \times \text{Weight (kg)} \\ +3.098 \times \text{Height (cm)} \\ -5 \times \text{Age (years)} & \text{(for females)} \end{cases}$$

- For `BIA-BIA_DEE`, we calculate it using the Total Daily Energy Expenditure (TDEE) formula:

$$\text{TDEE} = \text{BMR} \times \text{Activity Level}$$

where Activity Level is a multiplier determined by the individual's physical activity:

  - Sedentary (little or no exercise): 1.2
  - Lightly active (light exercise/sports 1–3 days/week): 1.375
  - Moderately active (moderate exercise/sports 3–5 days/week): 1.55
  - Very active (hard exercise/sports 6–7 days/week): 1.725
  - Super active (very hard exercise or physical job): 1.9

This calculation uses the `BMR` obtained from the Schofield Equation and the `BIA-BIA_Activity_Level_num` feature.

3) BMI_Internet_Hours: Formula:
`BMI_Internet_Hours` = `Physical_BMI` × `PreInt_EduHx-computerinternet_hoursday`

Description: This feature represents the interaction between an individual's Body Mass Index (BMI) and the average number of hours they spend using computers or the internet per day.

Purpose: - A higher value of `BMI_Internet_Hours` might indicate an individual with a higher BMI who also spends significant time online, potentially highlighting a link between sedentary habits and body composition.

4) Activity_Level_BMI: Formula:

$$\texttt{Activity\_Level\_BMI} = \frac{\texttt{Physical\_BMI}}{\texttt{BIA-BIA\_Activity\_Level\_num}}$$

Description: This feature represents the Body Mass Index (BMI) normalized by the individual's activity level, as captured by the `BIA-BIA_Activity_Level_num` feature.

Purpose: By dividing BMI by the activity level, this feature aims to provide a measure of BMI adjusted for physical activity. It can help assess whether individuals with higher activity levels tend to have lower BMIs or whether those with lower activity levels have higher BMIs, indicating potential correlations between activity and body composition.

- Higher values of `Activity_Level_BMI` may indicate individuals with a relatively high BMI despite having a moderate or high activity level.
- Lower values may suggest individuals whose BMI is well-regulated relative to their activity level, potentially indicating a healthier balance between physical activity and body composition.

5) BMI with Fat Percentage and Muscle Mass: **Features:**

- `BMI_Muscle`: Calculated as:

$$\texttt{BMI\_Muscle} = \texttt{Physical\_BMI} \times \texttt{BIA-BIA\_SMM}$$

This feature combines the individual's BMI with their Skeletal Muscle Mass (SMM) to highlight the contribution of muscle mass to BMI.

- `BMI_Fat_Percentage`: Calculated as:

$$\texttt{BMI\_Fat} = \texttt{Physical\_BMI} \times \texttt{Fat\_Percentage}$$

This feature combines BMI with body fat percentage to emphasize the influence of fat mass on BMI.

Description: BMI is a common metric for assessing body composition, but it does not distinguish between fat mass and muscle mass. Individuals who engage in regular strength training or physical exercise may develop significant muscle mass, leading to a higher BMI, similar to individuals with high fat mass. These features, `BMI_Muscle` and `BMI_Fat_Percentage`, provide additional context by quantifying the contributions of muscle and fat to overall BMI.

Purpose: By separating the influences of muscle and fat mass on BMI, these features can help differentiate between individuals with a high BMI due to muscle mass (e.g., athletes) and those with a high BMI due to excess fat. This distinction is crucial for more accurate assessments of health and fitness levels.

6) Muscle to Fat Percentage Ratio:

$$\texttt{Muscle\_Fat\_Percentage} = \frac{\texttt{BIA-BIA\_SMM}}{\texttt{Fat\_Percentage}/100}$$

Description: This feature calculates the ratio of Skeletal Muscle Mass (SMM) to Fat Percentage, providing insight into

an individual's body composition. A higher value of this ratio indicates a higher proportion of muscle mass relative to fat percentage.

Purpose: Individuals who engage in regular physical exercise, particularly strength training, often develop significant muscle mass while maintaining a low fat percentage. This feature highlights the balance between these two components, helping to distinguish physically active individuals from those with higher fat percentages and lower muscle mass.

Interpretation:

- A higher `Muscle_Fat_Percentage` suggests that the individual has a greater proportion of muscle mass compared to fat, which is common among those who work out regularly or lead an active lifestyle.
- A lower ratio may indicate a higher fat percentage relative to muscle mass, which could reflect a more sedentary lifestyle or health issues related to body composition.

.

7) Features Combining Internet Usage with Weight, BMI, and Fat Percentage: **Features:**

- `BMI_Internet_Hours`: $\texttt{BMI\_Internet\_Hours} = \texttt{Physical\_BMI} \times \texttt{PreInt\_EduHx-computerinternet\_hoursday}$
  This feature combines Body Mass Index (BMI) with the average daily hours spent using computers or the internet.
- `Fat_Percentage_Internet_Hours`: $\texttt{Fat\_Percentage\_Internet\_Hours} = \texttt{Fat\_Percentage} \times \texttt{PreInt\_EduHx-computerinternet\_hoursday}$
  This feature links body fat percentage to daily internet usage.
- `Weight_Internet_Hours`: $\texttt{Weight\_Internet\_Hours} = \texttt{Physical\_Weight} \times \texttt{PreInt\_EduHx-computerinternet\_hoursday}$
  This feature represents the interaction between body weight and internet usage hours.

Description: These features combine key physical health metrics—BMI, fat percentage, and weight—with the amount of time spent using the internet or computers daily. People with higher values for weight, BMI, or fat percentage who also spend more time online are likely to have correspondingly larger values for these features.

Purpose:

- Individuals with high BMI, fat percentage, or body weight, combined with significant internet usage time, will have large values for these features. This could suggest a link between sedentary behavior and physical health outcomes.
- Conversely, individuals with lower BMI, fat percentage, and weight, or those with limited internet usage, will have smaller feature values, indicating different lifestyle patterns.

8) Features Combining Activity Level with BMI and Fat Percentage:

- `Activity_Level_BMI`: $\texttt{Activity\_Level\_BMI} = \frac{\texttt{Physical\_BMI}}{\texttt{BIA-BIA\_Activity\_Level\_num}}$
  This feature represents the ratio of Body Mass Index (BMI) to the individual's activity level, as measured by `BIA-BIA_Activity_Level_num`.
- `Fat_Percentage_Activity_Level`: $\texttt{Fat\_Percentage\_Activity\_Level} = \frac{\texttt{Fat\_Percentage}}{\texttt{BIA-BIA\_Activity\_Level\_num}}$
  This feature represents the ratio of body fat percentage to the individual's activity level.

Description: These features provide a measure of BMI and fat percentage normalized by activity level. Individuals with lower activity levels (`BIA-BIA_Activity_Level_num`) combined with high BMI or high fat percentage will have disproportionately higher values for these features compared to individuals with a more active lifestyle.

Purpose:

- **Higher Values:** Individuals with low activity levels and high BMI or fat percentage will exhibit higher values for `Activity_Level_BMI` and `Fat_Percentage_Activity_Level`. This may suggest a sedentary lifestyle coupled with potential health concerns related to body composition.
- **Lower Values:** Individuals with higher activity levels and relatively lower BMI or fat percentage will have smaller values for these features, reflecting a healthier balance between activity and body composition.

9) Features Combining Age with Internet Usage and BMI: **Features:**

- `BMI_Age`: $\texttt{BMI\_Age} = \texttt{Basic\_Demos-Age} \times \texttt{Physical-BMI}$
  This feature combines Body Mass Index (BMI) with the age of participants.

- `Internet_Hours_Age`: $\texttt{Internet\_Hours\_Age} = \texttt{Basic\_Demos-Age} \times \texttt{PreInt\_EduHx-computerinternet\_hoursday}$
  This feature combines Internet Usage with the age of participants.

Description: These features show how age influences internet usage and the participant's health.

Purpose:

- Health and internet usage change with age, so combining age with these features will emphasize this relationship.

10) Features Combining CGAS_Score with Age and BMI:

- `CGAS_Age`: $\texttt{CGAS\_Age} = \texttt{CGAS-CGAS\_Score} \times \texttt{Basic\_Demos-Age}$
  This feature combines Children's Global Assessment Scale (CGAS) with the age of participants.

- `CGAS_BMI`: $\texttt{CGAS\_BMI} = \texttt{CGAS-CGAS\_Score} \times \texttt{Physical\_BMI}$
  This feature combines Children's Global Assessment

Scale (CGAS) with the Body Mass Index (BMI).

Description: These features provide clearer insights into children's psychological scales and the impact of mental health on physical health.

Purpose:

- Mental development changes with age, so combining CGAS with age will provide a more accurate picture of the participant's mental health.
- The goal is to examine the relationship between mental and physical health, so combining the CGAS score with BMI can reflect this connection.

IV. Final Result

After combining all the models and data processing techniques, we made a total of 106 submissions. Initially, our goal was to maximize the public score. However, this approach proved to be suboptimal as it led to overfitting on the public test set, resulting in significant underperformance on the private test set.

For our final submission, which achieved the highest public score, we recorded a public score of 0.491. Unfortunately, this submission only achieved a private score of 0.406. Conversely, the submission with the highest private score, which was 0.462, had a relatively lower public score of 0.457. This submission utilized the original unweighted ensemble model consisting of XGBoost, LightGBM (LGBM), and CatBoost. Notably, this version excluded TabNet and did not incorporate noise reduction using AutoEncoders, feature engineering, or advanced imputation techniques; missing data was simply filled using mean values.

An interesting trend emerged during the analysis: submissions that excluded TabNet tended to perform better on the private test set, regardless of whether other ensemble models were included. This observation held true even in scenarios with minimal preprocessing and feature engineering.

To sum up:

- **Best public score**: public score : **0.491**
  private score : **0.406**
- **Best private score**: public score : **0.457**
  private score : **0.462**