Machine Learning Project Report

Dinh Ngoc Duc
QH-2022-CS3 UET-VNU
22028166@vnu.edu.vn

Le Quang Hung
QH-2022-CS3 UET-VNU
22028103@vnu.edu.vn

Pham Duc Lam
QH-2022-CS3 UET-VNU
22028072@vnu.edu.vn

I. Introduction

Problematic Internet use among children and adolescents is becoming an increasing concern in today's digital era. With the rapid development of technology, the Internet has become easily accessible, and this has resulted in many people spending too much time on the Internet, especially children and adolescents, who tend to have a lot of free time. This behavior not only affects their daily routines and social interactions, but is also closely related to mental health challenges such as depression and anxiety. Understanding the causes, impacts, and potential solutions to this issue is essential to foster healthier online habits and ensure the well-being of the younger generation.

Current methods for measuring problematic Internet use in children and adolescents are often complex and require professional evaluations. The procedure may be expensive and time-consuming. In addition, many places may not have experts who can accurately assess the issues. Because of that, problematic Internet use is often not measured directly but is instead inferred from other issues such as depression or anxiety.

On the other hand, physical and fitness measures are highly accessible and require minimal intervention or clinical expertise. Furthermore, excessive use of technology is often associated with noticeable changes in physical habits, such as poor posture, unhealthy eating patterns, and decreased physical activity. As a result, we should try to utilize these easily obtainable physical fitness indicators to identify problematic Internet use, particularly in settings where clinical expertise or appropriate assessment tools are unavailable.

For that reason, Kaggle, the world's largest data science community, has hosted a competition named **Child Mind Institute — Problematic Internet Use**. In this competition, we are given a physical and fitness dataset and the goal is to predict from this data a participant's Severity Impairment Index (sii), a standard measure of problematic internet use. This result can then be used to to devise timely interventions that encourage healthier digital habits.

Thousands of people have participated in this competition, and many models have been proposed. Four approaches are suggested in this notebook [1]:

- Ordinal classification: A classification model with ordinal outputs.
- Multiclass classification: Standard classification, such as logistic regression.
- Regression: Predicting a numerical value representing the severity of the case, then rounding the prediction.
- Custom models: Unique, tailored approaches developed by participants.

Many participants chose the regression method, as this is a relatively simple model to deploy. However, selecting a model alone is not sufficient. The dataset has several issues that reduce model performance, such as missing data, skewed distributions in many fields, and the presence of outliers. To address these issues, various solutions have been proposed. For instance, KNN has been applied to fill in missing values based on the nearest neighbors [2].

To keep up with participants in the competition, we have read other works thoroughly, analyzed and implemented the key ideas that we consider the best. We also researched and thought of new ideas for improvement. The remainder of this report is organized as follows. In Section II, we will talk about the dataset in this competition, explore, analyze the data and point out some key challenges. Section III provides the details of the best method archived in the public leaderboard. In Section IV, the details of our experiments and conclusions about the experiment will be shown. And finally, the report is concluded in Section V.

II. Dataset

The dataset used in this competition is derived from a dataset named The Healthy Brain Network. It contains clinical samples of about five thousand 5-22-year-olds who have undergone both clinical and research screenings. From the dataset, two elements are extracted to be used in this competition: physical activity data and internet usage behavior data. The dataset is split into 2 sources: parquet files containing continuous recordings of accelerometer data for a single subject and csv files containing the remaining tabular data.

From the data table provided on Kaggle's website, we can get some basic information. There are 996 parquet files in series_train folder and 3960 rows in train.csv file, which means the number of times series files present in the training dataset is only a quarter of the total training samples. Analyzing the train.csv file, we can see that there are 82 columns. The 3 columns Basic_Demos-Enroll_Season, Basic_Demos-Age and Basic_Demos-Sex have no missing values. Taking a look at data_dictionaty file, we can see that there are 4 types of data type in the training dataset: categorical int, which indicates categorical data type with an integer number, string, int and float.

This dataset includes a variety of features designed to assess the physical, mental, and behavioral characteristics of children. Below is an overview of the key predictive features:

- **Demographics**: Information about age and sex of participants.
- **Internet Use**: Number of hours of using computer/internet per day.
- **Children's Global Assessment Scale**: Numeric scale used by mental health clinicians to rate the general functioning of youths under the age of 18.
- **Physical Measures**: Collection of blood pressure, heart rate, height, weight and waist, and hip measurements.
- **FitnessGram Vitals and Treadmill**: Measurements of cardiovascular fitness assessed using the NHANES treadmill protocol.
- **FitnessGram Child**: Health related physical fitness assessment measuring five different parameters including aerobic capacity, muscular strength, muscular endurance, flexibility, and body composition.
- **Bio-electric Impedance Analysis**: Measure of key body composition elements, including BMI, fat, muscle, and water content.
- **Physical Activity Questionnaire**: Information about children's participation in vigorous activities over the last 7 days.
- **Sleep Disturbance Scale**: Scale to categorize sleep disorders in children.
- **Actigraphy**: Objective measure of ecological physical activity through a research-grade biotracker. Many values seem to relate to a period after the PCIAT test was carried out.
- **Season**: For each set of measurements there is a 'season' feature which gives the season of the year when the measurements were carried out. These are the only predictive categorical features in the dataset and can be easily preprocessed.

The PCIAT (Parent-Child Internet Addiction Test) dataset comprises 22 features, including responses to 20 behavioral questions (each marked out of 5), the total PCIAT score, and the season in which the test was conducted. The total PCIAT score serves as a key indicator to categorize the level of internet addiction among children, and the dataset provides critical insights into their online habits. We have visualized the PCIAT dataset through various charts for analysis. First chart display two columns – the PCIAT features and their corresponding descriptions – to facilitate a clear interpretation of the behavioral questions and their impact on the total PCIAT score. Additionally, we created a correlation heatmap to further analyze the relationship between individual PCIAT features and the overall PCIAT total score. By computing the correlation coefficients for each feature against the total score and sorting them in descending order, the heatmap highlights the most influential features. This visualization not only helps uncover important features but also clarifies the relationship between each behavioral question and the overall PCIAT score, serving as a strong foundation for developing machine learning models and conducting deeper analyses.

SII (Severity Impairment Index) is a measure used to assess the extent of the impact of excessive internet use on children and adolescents. After conducting analysis, we found that this index is divided into four levels in the dataset:

- **SII = 0**: Unaffected, which included 1,594 cases.
- **SII = 1**: Mild impairment, which included 730 cases.
- **SII = 2**: Moderate impairment, which included 378 cases.
- **SII = 3**: Severe impairment, which included 34 cases.

A notable point is that among the 34 severe cases (SII = 3): 5 cases showed little to no actual internet usage. This raises concerns about potential data inconsistencies and highlights the need for deeper investigation. Furthermore, during the data analysis process, we observed that the test set does not contain the 22 PCIAT features. As a result, during model analysis and evaluation, we temporarily excluded them. However, we developed a model that uses PCIAT-Total to predict sii, yielding a relatively high private score of 0.424, which will be discussed in detail in the Experiment section.

To gain a deeper understanding of the dataset, we have visualized the data through various charts for analysis. First, we analyze percentage of missing data for each features. It can be shown that 78/81 columns have missing values. Some features like Fitness_Endurance-Time_Mins and Fitness_Endurance-Time_Sec have incredibly high missing data rate, therefore, we need to use some data imputation techniques to fully exploit the features. In contrast, features like Basic_Demos-Age or

PreInt_EduHx-computerinternet_hoursday have a high availability of data, so it can be used directly without complex imputation strategies. In general, the dataset still contains a large amount of missing values, therefore, a technique to infer those missing values is essential.

With accelerometer series data type, we have visualized some features of some samples of this type of data and observed that the length of the sample is incredibly long (340584 timesteps). There are two ways to handle this data type. The first approach is we try to train a whole model only using time series data to predict sii and then merge the result with the model trained using the tabular dataset. As this data type only contains information about the accelerometer, we hypothesize that it would not work well. We prefer to use the second approach, where we reduce the data dimensionality. By only retaining some key information in data like mean, median or max value, we have reduced the data dimensionality into a vector with size 96. However, this compact vector still remains relatively large amount of redundant information. Therefore, it still requires other data dimensionality reduction techniques.

To indicate outliers, we have analyzed every single feature by drawing box plots. These charts reveal many interesting facts about the dataset. In Physical-Diastolic_BP, Physical-Systolic_BP, Physical-BMI and BIA-BIA_BMI columns, there are some samples that have a value of 0, but in reality, the value of these columns is impossible to be 0. The value of Physical-Diastolic_BP, following this article [4], often fluctuates around 80, meanwhile, there is a point where its value is around 180. This value has a high chance of appearing due to some error in measurement. With FitnessGram, some columns at children's ages behave abnormally. Some children at the age of 5-12 can somehow curl up to more than 100 times, a feat which even an adult can hardly achieve. The same thing happens with Grip Strength. In Bio-electric Impedance Analysis related columns, the plots reveal even more outliers.

Noticing that there are many large values in the dataset, we have drawn a table to show the range of each feature. The table shows that the range of values varies significantly between columns. The values in the BIA-BIA_DEE column range from 1073.45 to 124728, which is an incredibly large discrepancy. Other columns, like PAQ_A-PAQ_A_Total, has a much smaller range (from 0.66 to 4.71).

III. Methods

A. Data cleaning

1) Handle missing value:

a) K-Nearest Neighbor Imputation: This approach is also referred to as distance-function matching. It involves selecting values from the k-nearest or most similar cases, and the value with the smallest distance is chosen to replace the missing value. Let x_{hk} represent a missing value for the h-th case, where h $\leq$ j, and in the j-th feature, using a straightforward distance function, the missing value can be estimated as follows:

$$x_{hk} = x_{qk} \quad q = \min_{p} d(x_{hk}, x_{pk})$$

The distance function $d(x_{hk}, x_{pk})$ can be defined as:

$$d(x_{hk}, x_{pk}) = \sum_{i \in \text{KNN}(X)} |x_{hk} - x_{ik}|$$

or

$$d(x_{hk}, x_{pk}) = \sqrt{\sum_{i \in \text{KNN}(X)} (x_{hk} - x_{ik})^2}$$

where K-NN (x_{hk}) is the index of the k^{th} closest cases of x_{ij} from the non-missing features. This method is different from others in that an actual value is used to impute and not a value constructed using regression imputation [5]. In [6], the author investigated various imputation methods for addressing missing scores (data) in biometric fusion. The research focused on multi-biometric systems, which combine multiple biometric information sources, as these systems are more effective in recognition than uni-modal systems. The imputation methods examined included K-Nearest Neighbor (KNN), maximum likelihood-based, Bayesian-based, and multiple imputation (MI) techniques. The experiments conducted, which evaluated the imputation of missing scores at different levels of missingness, revealed that the KNN-based method outperformed the other approaches. Similarly, in [7], the author analyzed a range of imputation techniques for handling missing numeric data, including mean, median, predictive mean matching, KNN, Bayesian linear regression, non-Bayesian linear regression, and random sample imputation. This study also identified the KNN imputation method as the most effective among those tested. Therefore, we decided to experiment with the KNN method to handle missing values in this competition.

2) Handle categorical data:

a) Label Encoding: Label Encoding is a technique used to convert categorical data into numerical values, which is essential since machine learning algorithms can only work with numbers. It works by assigning a unique integer to each category in a categorical feature. First, the unique values in the categorical column are identified, and then each unique value is mapped to a unique integer. For example, if a column contains values like ”Winter”, ”Spring”, and ”Fall”, these values may be mapped to integers like 0, 1, and 2 respectively. After mapping, the categorical values in the column are replaced with their corresponding integer values.

This method reduces memory consumption compared to storing strings and is compatible with machine learning models like XGBoost, CatBoost, and LightGBM, which process integer values efficiently.

In this dataset, we experimented with other methods like One-Hot Encoding and Ordinal Encoding, but they were not suitable. One-Hot Encoding would not be efficient because it creates binary columns for each category, leading to sparse data and high memory usage. In the case of categories with many unique values, this can increase dimensionality significantly, which may lead to computational inefficiency. Regarding Ordinal Encoding, it assumes an inherent order between categories, which is not applicable in this case. For example, in columns such as ”Winter”, ”Spring”, and ”Fall”, there is no natural ordering or ranking of these seasons, so applying Ordinal Encoding would be misleading and may introduce incorrect assumptions about the relationships between categories. [3].

Therefore, Label Encoding was chosen as the most appropriate method for this dataset, as it efficiently converts categorical values into numerical representations without introducing unnecessary dimensionality or incorrect ordinal relationships. This method is particularly effective for categorical variables that do not have a natural order and allows machine learning models to handle the data efficiently.

B. Feature engineering

1) Create new feature: It is evident that there is a significant correlation between Body Mass Index (Physical-BMI) and age (Basic_Demos-Age), with the impact of BMI on obesity increasing with age [9]. This suggests that multiplying BMI by age could provide a clearer indication of obesity, which can be linked to excessive internet use. So, we want to emphasize via:

$$BMI\_Age = Physical - BMI^{*} Basic\_Demos - Age \quad (1)$$

The paper [10] also highlights the relationship between Fat-Free Mass Index (BIA-BIA_FFMI) and Body Fat Percentage (BIA-BIA_Fat) with the autonomic nervous system, a factor that can influence behavior. By combining these indices, we can gain deeper insights into signs of internet addiction, as they may reflect changes both physically and behaviorally.

$$FFMI\_BFP = \frac{BIA - BIA\_FFMI}{BIA - BIA\_Fat} \quad (2)$$

In addition, this study mentions that long duration of internet use (PreInt_EduHx-computerinternet_hoursday) negatively affects individuals, with children being more susceptible [11]. By combining internet usage time with age, we can better determine the degree of internet addiction and provide crucial information for assessing the risk of excessive internet use.

$$Internet\_Hours\_Age = Basic\_Demos - Age^{*} PEch \quad (3)$$

$$BMI\_Internet\_Hours = Physical - BMI^{*} PEch \quad (4)$$

The paper [13] emphasizes the connection between BIA-BIA_Fat, BIA-BIA_BMI, and cardiovascular health. By combining these indices, we can create a new metric to assess cardiovascular health, which could potentially aid in diagnosing internet addiction through the influence of physical factors such as obesity and heart health.

$$BFP\_BMI = \frac{BIA - BIA\_Fat}{BIA - BIA\_BMI} \quad (5)$$

The authors in this paper [14] mentioned the use of Fat Mass Index (BIA-BIA_FMI) and Body Fat Percentage (BIA-BIA_Fat) to derive both quantitative and qualitative information about athletes' training status and inherent characteristics. For example, these indices could help identify individual strengths and weaknesses in physical performance. Moreover, the authors discussed the correlation between lean soft tissue (BIA-BIA_LST) and body water percentage (BIA-BIA_TBW) with the musculoskeletal system. We believe that lower physical activity indices and musculoskeletal system metrics may indicate a higher likelihood of internet addiction. Therefore, we aim to clarify this by formulating a specific equation as follows:

$$FMI\_BF = \frac{BIA - BIA\_FMI}{BIA - BIA\_Fat} \quad (6)$$

$$LST\_TBW = \frac{BIA - BIA\_LST}{BIA - BIA\_TBW} \quad (7)$$

From this paper [15], it is evident that body fat percentage (BIA-BIA_Fat) has a significant impact on the basal metabolic rate (BIA-BIA_BMR). By multiplying body fat percentage with BMR, we can form a new metric that not only reflects the body's metabolic rate but is also closely related to obesity and an individual's level of physical activity. Specifically, this calculation helps us better understand the body's energy consumption and the effect of body fat percentage on overall health. The relationship between body fat percentage, daily energy expenditure (BIA-BIA_DEE), and obesity provides important information for assessing the risk of related health conditions. Furthermore, by multiplying body fat percentage with daily energy expenditure, we can create an index directly related to obesity, which has a strong connection with physical activity intensity, therefore helping to identify signs related to problematic internet usage behavior. Based on this, we have developed a formula:

$$BFP\_BMR = BIA - BIA\_Fat^{*} BIA - BIA\_BMR \quad (8)$$

$$BFP\_DEE = BIA - BIA\_Fat^{*}BIA - BIA\_DEE \quad (9)$$

Otherwise, in this page [16], it is evident that Basal Metabolic Rate (BIA-BIA_BMR) and body weight (Physical-Weight) are closely related and both have direct correlations with obesity level. This is also an indicator that may be linked to the sii index. Based on the BMR formula (BMR = (10 x weight in kg) + (6.25 x height in cm) - (5 x age in years) + 5), we can see that after calculation, an index can be determined based on factors such as age and weight. Both weight and age influence an individual's physical activity levels, which in turn affect internet usage behavior. We have:

$$BMR\_Weight = \frac{BIA - BIA\_BMR}{Physical - Weight} \quad (10)$$

In this paper [17], the author demonstrates the use of energy expenditure per kilogram, whereas our dataset provides total energy expenditure (BIA-BIA_DEE) and body weight (Physical-Weight). Therefore, we will apply the formula outlined by the author. Furthermore, we identified a formula to assess the relationship between Skeletal Muscle Mass (BIA-BIA_SMM) and Physical-Height, as described on this page [18]. While the author used the RSMI formula, we simplified it by not squaring the height for better generalization.

$$DEE\_Weight = \frac{BIA - BIA\_DEE}{Physical - Weight} \quad (11)$$

$$SMM\_Height = \frac{BIA - BIA\_SMM}{Physical - Height} \quad (12)$$

This paper [19] also illustrates that the ratio of Fat Mass Index (BIA-BIA_FMI) to Skeletal Muscle Mass (BIA-BIA_SMM) may be a more effective measure of obesity compared to BMI. Additionally, the author in [20] mentioned that total water intake (mL/kg) (BIA-BIA_TBW and Physical-Weight) decreases with age in both genders. Thus, we aim to emphasize these indices more in our dataset.

$$Muscle\_to\_Fat = \frac{BIA - BIA\_SMM}{BIA - BIA\_FMI} \quad (13)$$

$$Hydration\_Status = \frac{BIA - BIA\_TBW}{Physical - Weight} \quad (14)$$

Furthermore, the author in [21] highlights the correlation between Intracellular Water (BIA-BIA_ICW) and Total Body Water (BIA-BIA_TBW) in relation to depression. The observed relationship suggests that depression might play a significant role in diagnosing the Social Internet Index (SII).

$$ICW\_TBW = \frac{BIA - BIA\_ICW}{BIA - BIA\_TBW} \quad (15)$$

2) AutoEncoder: An Autoencoder is a method that compresses a multidimensional sequence (e.g., a windowed time series consisting of multiple values from sensors, clicks, etc.) into a single vector representing the entire sequence. With a well-designed encoder and decoder, the latent vector can be utilized as an input to a multilayer perceptron (MLP) or as an additional set of features in a larger multi-head network. This approach has demonstrated its efficiency in dimensionality reduction, optimizing data processing and analysis [22].

C. Model

1) VotingRegressor: Voting Regressor is an ensemble learning method that combines multiple base regression models to create a robust final prediction. By averaging the outputs of several regressors, it reduces the risk of overfitting and improves overall accuracy. This method is easy to implement and can leverage a variety of base models. However, its performance heavily relies on the choice of base models.

2) XGBoost: XGBoost is a machine learning algorithm developed based on the Gradient Boosting framework, optimized for efficiently handling datasets ranging from medium to large sizes. One of its key advantages lies in its ability to handle missing data without requiring additional pre-processing steps, simplifying the preparation of input data. This feature has made XGBoost a popular tool in machine learning competitions, especially on the Kaggle platform, including the current competition we are participating in. [24] However, a major limitation of XGBoost is its slower training speed compared to LightGBM, primarily due to the complexity of its numerous hyperparameters, which require advanced skills to optimize effectively.

3) LightGBM: LightGBM (LGBM) is a machine learning algorithm based on the Gradient Boosting framework, designed to construct decision trees using a leaf-wise strategy to optimize both performance and training speed. By employing leaf-wise splitting combined with a histogram-based approach, LGBM efficiently handles large-scale datasets. Furthermore, the algorithm supports parallel computation, GPU optimization, and the conversion of categorical features into integer representations, which reduces storage overhead. These advantages make LGBM the baseline algorithm selected for our study [23]. However, due to its sensitivity to small datasets (prone to overfitting) and the lack of built-in mechanisms for handling missing data, we will address these limitations during the competition.

4) CatBoost: CatBoost, similar to LightGBM and XGBoost, is a robust Gradient Boosting algorithm with unique features that make it stand out in specific scenarios. Its most prominent advantage lies in its superior handling of categorical features, including missing values, without requiring complex preprocessing steps. Additionally, the number of hyperparameters needing optimization is significantly reduced, simplifying its implementation process. CatBoost's greatest strength is its high performance on mixed datasets, making it highly suitable for the dataset provided in this competition. [25] However, the algorithm's drawbacks include higher hardware resource consumption compared to LightGBM and XGBoost, as well as slower training times compared to LightGBM.

5) TabNet: TabNet is a deep learning model specifically designed for tabular data, and it is one of the few deep learning architectures capable of outperforming Gradient Boosting algorithms such as LightGBM, XGBoost, and CatBoost in certain scenarios. By leveraging an advanced Attention mechanism, End-to-End learning (operating directly on raw data without complex preprocessing), the ability to handle mixed-type data, and efficient GPU optimization, TabNet emerges as a promising solution for addressing the challenges posed by this Kaggle competition [26]. However, as a deep learning model, TabNet demands significantly more hardware resources compared to traditional methods. Additionally, its performance on small datasets is typically inferior to Gradient Boosting algorithms, which is a crucial consideration during the model training process.

IV. Experiment

In this section, we detail the simulation experiments conducted to evaluate the effectiveness of various methods integrated into the model. We developed three versions to study and produce results for the competition. The first version employs a single model, XGBoost, to calculate the SII based on PCIAT. The second version uses Voting Regression (LightGBM + XGBoost + CatBoost) with three submissions. In the third version, the approach is similar to the second, but an additional model, TabNet, is incorporated. Finally, based on the analysis and evaluation results, we propose the most viable approach identified through our research.

In all of our experiments, the data will be sourced from the Child Mind Institute — Problematic Internet Use competition. The methods will be evaluated based on two key factors: Public score and Optimized QWK score. Public score plays an important role in model evaluation and is calculated using approximately 38% of the test data, with the final results derived from the remaining 62% of the hidden test data. This factor serves to assess the model's ability to accurately predict unlabeled data. Another crucial factor is the Optimized QWK score, which aligns with the evaluation method proposed in the competition. This score is based on the quadratic weighted kappa (QWK), which quantifies the level of agreement between two outcomes. The QWK metric typically ranges from 0 (indicating random agreement) to 1 (indicating perfect agreement). However, in certain cases, when the observed agreement is lower than expected by chance, the QWK score may fall below 0.

A. Version 1: Use XGBoost with 1 submission

1) Baseline: In our baseline approach, we started by loading and processing the input data. The tabular data contains two main types of features: numerical and categorical. To facilitate the training process, we converted the categorical features into numerical form using Label Encoding. For the training model, we selected the XGBoost regression model to predict the PCIAT-PCIAT_Total variable. Basic hyperparameters such as max_depth, n_estimators, learning_rate, subsample, and colsample_bytree, along with the k-fold cross-validation method, were utilized. The average Kappa score was calculated on the train/validation splits. From this encouraging baseline, we will explore additional methods in subsequent stages, including advanced data preprocessing techniques, model development, and hyperparameter tuning, as described in the following sections.

2) Data cleaning experiment: For the categorical columns, we replaced missing values with 0 and mapped the seasonal categories ('Spring', 'Summer', 'Fall', 'Winter') to the corresponding numerical values 1, 2, 3, and 4. This transformation was applied to both the training and testing datasets to ensure consistency. After that, we removed rows whose SII value is NaN.

3) Feature engineering experiment: To improve the model, we decided to perform feature selection based on the correlation with PCIAT-PCIAT_Total as the target variable. Given the large number of features, we focused on selecting those with the strongest correlation to the target, while removing the weaker ones. Upon reviewing the data, we noticed that two features, BMI and sleep disturbance, were measured twice, yielding slightly different results. Since these features were closely similar, we decided to drop the duplicates. Specifically, we removed the features Physical-BMI and SDS-SDS_Total_Raw, as they had slightly lower correlations with the target. we then calculated the correlations between all features and PCIAT-PCIAT_Total, and selected those with an absolute correlation greater than 0.1. This resulted in a refined list of features, excluding the target variable, the SII feature, and the redundant features we had identified. In the end, we were left with 19 features.

In handling missing values, we found that the dataset contained many missing entries, with 46 columns having missing data and 8 columns missing more than half of their values. For example, the feature Physical-Waist_Circumference had the majority of its value missing. To address this, we decided to remove any columns that had more than 50% of their values missing, ensuring the integrity of the data. After identifying these columns, we removed them from the list of selected features. This process helped ensure that the remaining data would be reliable for further analysis and model training. Based on the correlation analysis and the proportion of missing data, 16 features were selected to build the model. The feature selection process not only helps reduce noise but also improves the model's performance by focusing on the most important signals. The selected features include: Basic_Demos-Age, Physical-Height, Physical-Weight, PreInt_EduHx-computerinternet_hoursday, FGC-FGC_CU, and so on. Some features with unreasonable min/max values (such as a minimum weight of 0) were checked and validated to ensure the data's usability.

4) Model experiment: For the training model, we created a custom function to map the PCIAT scores into categories. The function works by adjusting the score threshold with a scaling factor of 1.252 to improve the QWK result. This adjustment was necessary because without it, or with a different scaling factor, the QWK result was lower. For example, with a scaling factor of 1.5, the QWK decreased to 0.383. The custom function works as follows: it multiplies the scores by 1.252, then classifies the scores into bins: 0 for scores less than or equal to 30, 1 for scores between 30 and 50, 2 for scores between 50 and 80, and 3 for scores greater than or equal to 80. To evaluate the model's performance, we used the QWK metric. The QWK is calculated using a function that converts both the true and predicted PCIAT scores into categories using the mapping function, and then calculates the quadratic weighted kappa score based on the Cohen kappa metric. For hyperparameter optimization, we used Optuna to optimize key hyperparameters such as max_depth, n_estimators, learning_rate, subsample, and colsample_bytree. We performed cross-validation using StratifiedKFold with 5 datasets. Through this process, we identified the optimal hyperparameters: max_depth = 3, n_estimators = 92, learning_rate = 0.063, subsample = 0.659, and colsample_bytree = 0.905. We then trained the model using these optimized parameters with 10-fold cross-validation, achieving an average QWK score of 0.451. Finally, we assessed the importance of the features using XGBoost's built-in method and visualized the results with a chart. After the above steps, we trained the XGBoost regression model on our training dataset and used the trained model to make predictions on the test dataset. Then, we used the convert function to map the predicted scores to the corresponding SII categories in order to generate the final submission file.

5) Result: This version has an optimized QWK score of 0.451, with the public score being the lowest among the three versions at 0.451. However, it achieved the highest private score of 0.424. This discrepancy can be explained by the risk of overfitting when combining models like LightGBM, XGBoost, CatBoost, or even TabNet. The ensemble approach may become too focused on specific patterns in the public test data, which may not generalize well to the private test data. By using XGBoost Regression with a simpler design, the model is less prone to overfitting, which allows it to generalize better on the private test set. Furthermore, although powerful, combining multiple models such as TabNet and XGBoost can introduce noise or conflicts in predictions, leading to poorer performance on the private data.

B. Version 2: Use Voting Regression (LightGBM + XGBoost + CatBoost) with 3 submission

1) Baseline: We started by loading and processing the input data. We handled the time series data by implementing a parallel method to efficiently extract summary statistics from parquet files. The process_file function reads each parquet file, removes the step column, and reshapes the summary statistics (from describe()) into a single array. This process is applied to all files in the directory using the load_time_series function. By utilizing ThreadPoolExecutor, we process the files in parallel, significantly accelerating the procedure. The extracted statistics are then aggregated into a dataframe, with each row representing a set of features corresponding to a unique sample ID. Converting the time series data into tabular format in this way is crucial for merging it with the main dataset. After processing, we merged the data into the tabular dataset by matching the sample IDs. Additionally, since the PCIAT feature set consists of questions that can directly calculate the target value SII, and these features are not present in the test set, we temporarily excluded the PCIAT features during the training phase.

The tabular data contains two main types of features: numerical and categorical. To facilitate the training process, we converted the categorical features into numerical form using Label Encoding.

For the training model, we chose the Voting Regression model by combining LightGBM, XGBoost, and CatBoost across three submissions to enhance the model's accuracy and stability. By utilizing three different models, we can leverage the strengths of each algorithm, minimizing bias and improving generalization on unseen data.

2) Data cleaning experiment: We implemented the same approach as in version 1: handling categorical data and dropping rows where SII is NaN. This ensures consistency in data pre-processing, reducing noise and focusing on relevant features for model training. We also applied an outlier removal process to clean the data and improve the reliability of our model. The remove_outliers function was used to filter out unrealistic or erroneous data points. We specifically removed records where certain physical measurements were either zero or exceeded plausible thresholds. For example, we excluded samples with Physical-BMI, Physical-Diastolic_BP, and Physical-Systolic_BP values of zero, as well as records where Diastolic_BP exceeded 160. Additionally, for children under 12, we removed entries where FGC-FGC_CU and FGC-FGC_GSND values were greater than 80, as they were considered outliers for this age group. We also addressed extreme values in the BIA features, removing records with implausibly high values in body composition metrics (e.g., BIA-BMI, BIA-BMR) exceeding realistic thresholds. This process helped eliminate noise from the data and allowed the model to focus on more meaningful and accurate patterns. Without this step, the public score would drop to 0.442.

3) Feature engineering experiment: Initially, we conducted the experiment using only the existing features, resulting in a score of 0.445. To improve, we conduct two experiments to evaluate the effectiveness of adding new features to the model. First, we utilize a heatmap score to analyze and compare the correlation between the old features and the newly created features with the target variable (sii) across various feature pairs. The heatmap score serves as a visual tool that helps in easily identifying the relationships between features and the target variable. For the majority of the new features, the correlation with the target variable (sii) is generally higher than one or two of the old features. This suggests that the new features contribute more to the prediction of the target variable compared to the older ones, enabling the model to capture more intricate patterns from the data. This indicates that the newly created features may improve the model's predictive capability, especially when they reflect deeper relationships within the data. For the second experiment, the features selected (feature selection) also included the new ones (15 new features in III.B.1). It is evident that when the new features are incorporated into the model, both Public score and Optimized QWK score are significantly higher compared to the baseline model. This demonstrates that the creation of new features not only enhances the model's accuracy in prediction but also improves the agreement between the predicted outcomes, as reflected by the improvement in the Optimized QWK score. These results provide clear evidence that adding new features to the model can lead to significant improvements in performance, both in terms of prediction accuracy (Public score) and the level of agreement between the predictions (Optimized QWK score).

4) Model experiment: For the training model, we used a series of steps to optimize performance and evaluate predictions. Initially, the training dataset was divided into features (X) and target labels (y). To ensure robust model validation, we applied Stratified K-Fold cross-validation with 5 splits, using StratifiedKFold from scikit-learn. For each fold in the cross-validation, we trained the model on the training set and evaluated its performance on the validation set. Predictions for both the training and validation sets were made, and the non-rounded predictions were stored in oof_non_rounded. These predictions were then rounded to integer values for comparison with the actual target values, and we calculated the Quadratic Weighted Kappa (QWK) score for both the training and validation sets. The model's performance was tracked for each fold, and the mean QWK scores for both the training and validation sets were computed. After completing the cross-validation, we used an optimization process to find the best threshold values for rounding the predictions. This was done by minimizing the QWK score with the minimize function, utilizing the Nelder-Mead method. The optimized thresholds were then used to adjust the predictions and evaluate the model's final performance. Finally, the predictions for the test set were generated by averaging the predictions from each fold, applying the optimized threshold, and creating a submission file with the predicted values. The model and the submission file were then returned for further use. This approach helps improve the model's reliability and accuracy by optimizing the thresholds used for prediction rounding and leveraging cross-validation to reduce overfitting.

In the process of selecting and designing model architectures, we experimented with 3 fundamental and widely recognized models for tabular data: LightGBM, XGBoost, CatBoost. Using hyperparameter configurations referenced from various notebooks, we tested each model individually and observed results were consistent with our initial expectations. Specifically, LightGBM demonstrated the fastest training speed, while the other models exhibited decreasing speeds in the listed order. This observation aligns with the general characteristics of these 3 models when comparing training time on the same dataset. Regarding the Public Score, the performance of the models showed minor differences, indicating that the dataset had already been effectively utilized by each individual model. We further experimented with Voting Regressor, using the 3 aforementioned models as base models. To maximize performance, we made 3 separate submissions using different model combinations. The first submission utilized a Voting Regressor with all three models (LightGBM, XGBoost, and CatBoost) as base models. For the second submission, we opted to use LightGBM alone due to its fast training speed, as mentioned earlier. The third submission involved a Voting Regressor using only XGBoost and CatBoost, excluding LightGBM. This approach allowed us to evaluate the contribution of each model combination to the final performance, providing insights into the strengths of each model and how they complement each other.

5) Result: This version has the lowest QWK score of 0.455 in the second submission and the highest score of 0.460 in the first submission, with a relatively high public score of 0.456. However, it only achieved a private score of 0.398. This indicates that the model may have overfitted to patterns present in the initial part of the test data, leading to a higher public score. However, the model's performance dropped significantly on the remaining portion of the data, suggesting that the patterns learned during training did not generalize well to the private dataset. A potential reason for this could be that the distribution of the private test data differs significantly from the public test data, exposing weaknesses in the model's generalization ability. Additionally, the complexity of combining multiple models may have introduced noise or conflicting signals, reducing performance on unseen data. This highlights the challenge of balancing model complexity with generalization capability, where simpler models like XGBoost Regression can sometimes outperform complex ensemble methods by avoiding overfitting.

C. Version 3: Use Voting Regression (LightGBM + XGBoost + CatBoost + TabNet) with 3 submission

1) Baseline: Similar to version 2, we also loaded and processed the time series data and temporarily excluded the PCIAT features during the training phase. The difference in this version is that in submission 3 with TabNet, we temporarily excluded the categorical data. This adjustment was made because, during the improvement process, we found that when TabNet was added without excluding the categorical data, both the QWK and public score decreased. For the training model, we also used Voting Regression, but the key difference is that we added TabNet in addition to the three existing models.

2) Data cleaning experiment: In addition to the steps implemented in version 2, we also applied a new technique for handling missing values. Specifically, we used KNNImputer with 5 neighbors to impute missing values in the numeric columns of the dataset. After applying the imputation, we rounded the target variable sii and converted it to integer values. We then merged the imputed data with the original dataset, ensuring that all necessary features were preserved. Before using the dataset for model training, we also handled any infinite values by checking if any of the entries in the dataset were infinite. If we found any such values, we replaced them with NaN. This ensured that no invalid data remained in the dataset.

3) Feature engineering experiment: In addition to what was done in version 2, we used AutoEncoder for both the available training and test datasets, with 100 epochs, reducing the time series data dimensionality to 60, and a batch size of 32. In the AutoEncoder model, after each Linear layer in the encoder part, we applied a LeakyReLU activation function with a parameter of 0.2. The use of LeakyReLU helps mitigate the issue of dead neurons, where certain neurons may become ”inactive” and no longer contribute to the training process. Additionally, we chose the Adam optimization method for training. Adam is one of the most popular and effective optimization algorithms, particularly for training neural network models, due to its ability to adjust the learning rate dynamically and optimize efficiently. We observed that adding the AutoEncoder to the baseline model led to significant improvements. Both the Public score and Optimized QWK score increased compared to the original model. This indicates that applying the AutoEncoder enhanced the model's ability to reconstruct and compress data, leading to more accurate predictions. As a result, we have decided to include this component in the final version of the model.

4) Model experiment: In addition to the model training steps in version 2, we further implemented TabNet training using the TabNetWrapper class. This class not only integrates a missing value imputer (SimpleImputer) to handle missing data but also utilizes a custom callback (TabNetPretrainedModelCheckpoint) to save the best-performing TabNet model during training. The TabNetPretrainedModelCheckpoint callback monitors evaluation metrics like val_loss and saves the model when improvements are detected. The model is only saved if its performance on the validation set improves, ensuring that the best version is retained. These adjustments significantly enhanced the model's performance, leading us to include them in the final model version.

In this version, we experimented with 4 fundamental and widely recognized models for tabular data: LightGBM, XGBoost, CatBoost, TabNet and Voting Regressor, using the 4 aforementioned models as base models. To maximize performance, we made 3 separate submissions using different model combinations. The first submission utilized a Voting Regressor with three models (LightGBM, XGBoost, and CatBoost). In the second submission, we chose to use CatBoost because of its strong capability to handle numerical data effectively, even when categorical features are excluded. CatBoost is known for performing well on structured datasets, especially those composed entirely of numerical features. The removal of categorical features and the use of AutoEncoder for dimensionality reduction further highlighted CatBoost's advantage in extracting valuable insights from numerical data, enabling the model to achieve high performance without relying on categorical information. The third submission involved a Voting Regressor using all four models: XGBoost, CatBoost, LightGBM, and TabNet, with CatBoost given a higher weight due to its strong capabilities, as mentioned earlier. This approach allowed us to evaluate the contribution of each model combination to the final performance.

5) Result: This version has the lowest QWK score of 0.451 in the second submission and the highest score of 0.530 in the third submission, with the highest public score of 0.459. However, it only achieved a private score of 0.388. Using TabNet in deep learning models can improve the public score due to its ability to learn complex features, especially when the public data is similar to the training data. However, a high public score does not guarantee good performance on unseen data (private). The decrease in the private score when using TabNet may be due to overfitting on the public data, distributional differences between the public and private datasets, and the complexity of the model that makes generalization more difficult. After the competition ended, we reviewed the private test results of notebooks using TabNet and found that they all had very high public scores (0.49 or higher, equivalent to the gold prize of the competition), but their private scores were only around 0.407, highlighting the instability in the model's generalization ability when faced with different data distributions.

V. Summary

A. Result

Among the three versions mentioned above, we decided to select the last two versions for submission to the competition. This decision was based on the observation that these two versions achieved higher QWK values and public scores compared to the first version, with 0.456 for version 2 and 0.459 for version 3. The results yielded private scores of 0.398 for version 2 and 0.388 for version 3.

B. Future Trajectory

Through this competition and the results obtained, we have identified several future development directions. We aim to improve the predictive model by developing personalized risk assessment models tailored to the specific circumstances of each child, such as age, environment, and Internet usage habits, among other factors. Additionally, we seek to refine machine learning models to achieve better prediction performance, ensuring reliability when tested on diverse and new datasets.

C. Knowledge Gained

After the competition, we gained a deeper understanding of the concepts and mechanisms related to unhealthy Internet use, particularly its effects on the psychology, behavior, and neurological health of children. Furthermore, working with a large dataset provided us with an opportunity to study and analyze data processing techniques. We also learned how to handle EEG data or behavior-related data to analyze the correlation between neurological factors and Internet usage behavior. Moreover, we got introduced to Python programming, utilizing basic machine learning models, and adjusting parameters and hyperparameters to optimize models and enhance their performance.

D. Contributions

Programming: Le Quang Hung, Dinh Ngoc Duc

Report Writing: Pham Duc Lam, Le Quang Hung, Dinh Ngoc Duc

Presentation: Pham Duc Lam

Acknowledgment

We would like to express our heartfelt gratitude to Dr. Ta Viet Cuong, BSc Nguyen Minh Duc and BSc Le Viet Quan for providing valuable knowledge on machine learning and answering our questions, enabling us to complete this competition and report.

We also extend our thanks to the authors of the articles and books we referenced, as they provided a wealth of critical knowledge throughout the competition process.

References

[1] "CMI-PIU: Features EDA", Antonina Dolgorukova, Kaggle Notebook.

[2] "LB0.494 with TabNet", Ichigo_E, Kaggle Notebook.

[3] "Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists", Alice Zheng, Amanda Casari.

[4] "Systolic vs. Diastolic Blood Pressure, why both numbers are important", By Richard N. Fogoros, MD

[5] "A Review of Missing Data Handling Techniques for Machine Learning", Luke Oluwaseye Joel, Wesley Doorsamy, Babu Sena Paul.

[6] "A comparison of imputation methods for handling missing scores in biometric fusion ", Ding Y. and Ross A

[7] "Comparison of performance of data imputation methods for numeric dataset", Jadhav A., Pramod D. and Ramanathan K.

[8] "Missing data imputation using genetic algorithm for supervised learning", Shahzad W., Rehman Q. and Ahmed E. *International Journal of Advanced Computer Science and Applications, 2017*

[9] "The influence of body mass index, age and gender on current illness: A cross-sectional study", Brand Lee Jarret, G J Bloch, D Bennet.*International journal of obesity*

[10] "Correlation of Body Mass Index, Body Fat percentage and Fat Free Mass Index with Autonomic Nervous Function", Swikruti Behera, Debasish Das.

[11] "Age Differences in the Relationship Between Daily Social Media Usage and Affect", Xin Yao Lin , Margie Lachman .

[12] "Association between Times Spent on the Internet and Weight Status in Korean Adolescents", Seong-Ik Baek , Wi-Young So.

[13] ”A comparison of body mass index and body fat percentage for predicting cardiovascular disease risk”,Hosein Sheibani Maryam Saberi-Karimian, Habibollah Esmaily, Mohsen Mouhebati, Mohmoud Reza Azarpazhooh, Ghasemali Divbands,Marzieh Kabirian, Roshanak Ghaffarian, Maryam Tayefi, Gordon A. Ferns,Mohammad Safarian, Majid Ghayour-Mobarhan

[14] ”Regional Lean Soft Tissue and Intracellular Water Are Associated with Changes in Lower-Body Neuromuscular Performance: A Pilot Study in Elite Soccer Players” , Tindaro Bongiovanni, Grant Tinsley, Giulia Martera, Carmine Orlandi, Federico Genovesi, Giuseppe Puleo, Giuseppe Puleo, Athos Trecroci.

[15] ”Impact of body fat mass and percent fat on metabolic rate and thermogenesis in men”, K. R. Segal,I. Lacayanga,A. Dunaif,B. Gutin, andF. X. Pi-Sunyer

[16] ”Metabolic rate after massive weight loss in human obesity.”, Finer N , Swan PC , Mitchell FT.

[17] ”Body weight control and energy expenditure”, Danielle Cristina Fonseca and Priscila Sala and Beatriz de Azevedo Muner Ferreira and Jessica Reis and Raquel Susana Torrinhas and Itai Bendavid and Dan Linetzky Waitzberg

[18] , ”Relationship of body anthropometric measures with skeletal muscle mass and strength in a reference cohort of young Finnish women”, SL Qazi , T Rikkonen , H Kröger , R Honkanen , M Isanejad, O Airaksinen, J Sirola

[19] , ”REDEFINING OBESITY: A RATIO OF FAT AND MUSCLE MASS COMPARED TO BODY MASS INDEX IN OLDER ADULTS”, Kworweinski Lafontant, Ladda Thiamwong, Jeffery Stout, Joon-Hyuk Park, Rui Xie, David Fukuda

[20] , ”Total water intake by kilogram of body weight: Analysis of the Australian 2011 to 2013 National Nutrition and Physical Activity Survey”, Laura J. Mallett Undergraduate MD, Vidhun Premkumar Undergraduate MD, Leanne J. Brown PhD, AdvAPD, Jennifer May PhD, FRACGP, FACRRM, Megan E. Rollo PhD, APD, Tracy L. Schumacher PhD

[21] , ”Decreased Intracellular to Total Body Water Ratio and Depressive Symptoms in Patients with Maintenance Hemodialysis”,Maolu Tian, Zuping Qian , Yanjun Long , Fangfang Yu , Jing Yuan, Yan Zha.

[22] ”Using LSTM Autoencoders on multidimensional time-series data”, Sam Black

[23] Song, Q., Ge, H., Caverlee, J., Hu, X. (2019). Tensor completion algorithms in big data analytics. ACM Transactions on Knowledge Discovery from Data (TKDD), 13(1), 1-48.

[24] Stylos, N., Zwiegelaar, J. (2019). Big data as a game changer: how does it shape business.

[25] Leonelli, S., Tempini, N. (2020). Data journeys in the sciences, 412. Springer Nature.

[26] Arik, S. Ö., Pfister, T. (2021). TabNet: Attentive Interpretable Tabular Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 35(8), 6679-6687.