VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF INFORMATION TECHNOLOGY

FINAL PROJECT REPORT

PYTHON FOR MACHINE LEARNING
CS116

Child Mind Institute - Problematic Internet Use

Professor: PhD. Nguyen Vinh Tiep

Project Team Members

No.	Name	Student ID
1	Tran Nhu Cam Nguyen	22520004
2	Tran Thi Cam Giang	22520361
3	Nguyen Huu Hoang Long	22520817
4	Hoang Ngoc Quan	22521178

Ho Chi Minh City, January 2025

Contents

1 Overview

This report outlines our team's participation in the Child Mind Institute – Problematic Internet Use competition, which sought to predict the Severity Impairment Index (SII) associated with internet addiction utilizing physical and behavioral data. In response to the competition objectives, we implemented a comprehensive machine learning pipeline that included exploratory data analysis (EDA), data pre-processing, feature engineering, and advanced modeling techniques.

Our approach is designed to address significant challenges present in the dataset, such as skewed class distributions. Through this adaptive methodology, we achieved a competitive Quadratic Weighted Kappa (QWK) score, resulting in a commendable rank with a bronze medal. This report highlights the rationale behind our chosen techniques, presents key results, and discusses valuable lessons learned, ultimately underscoring the importance of adaptability and collaboration in effectively addressing real-world machine learning challenges.

2 Introduction

2.1 Competition Overview

The Child Mind Institute - Problematic Internet Use competition challenges participants to predict the **Severity Impairment Index (SII)** for internet addiction using physical and behavioral data. The levels - *None, Mild, Moderate, and Severe* - derived from the Parent-Child Internet Addiction Test (PCIAT). The primary objective is to use machine learning to identify early signs of problematic internet use, allowing the promotion of healthier digital habits.

2.2 Dataset

The competition dataset comprises two main types of data:

- **Time Series Data:** Dynamic measurements of physical activity collected over time and stored in Parquet files. These measurements reflect participants' physical movements and behaviors.
- **Tabular Data:** Static demographic, health, and behavioral characteristics stored in CSV files. These features include information such as age, gender, and self-reported patterns of Internet use.

Target Variable: The target variable, **Severity Impairment Index (SII)**, is calculated from the PCIAT questionnaire, which includes 20 items rated on

a scale of 0 to 5. The total score ($PCIAT\_Total$) categorizes participants into one of the following SII levels:

- **0: None** ($PCIAT\_Total = 0 - 30$)
- **1: Mild** ($PCIAT\_Total = 31 - 49$)
- **2: Moderate** ($PCIAT\_Total = 50 - 79$)
- **3: Severe** ($PCIAT\_Total = 80 - 100$)

2.3 Evaluation Metric

Performance is assessed using the **Quadratic Weighted Kappa (QWK)** metric. QWK evaluates the agreement between predicted and actual SII levels while accounting for the magnitude of misclassifications. The metric is particularly effective for ordinal regression tasks.

- $Score = 1$**:** Perfect agreement between predictions and ground truth.
- $Score = 0$**:** The agreement is purely random.
- $Score < 0$**:** Agreement is worse than random.

To calculate the **Quadratic Weighted Kappa**, three matrices are constructed: O, W, and E, where N represents the number of distinct labels.

- **Matrix** O**:** This is an $N \times N$ histogram matrix. Each element $O_{i,j}$ represents the count of instances where the actual label is i and the predicted label is j.
- **Matrix** W**:** This is an $N \times N$ weight matrix, calculated based on the squared difference between the actual and predicted labels. The weight for each pair (i, j) is computed as:

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2}.$$

  This weighting emphasizes larger penalties for greater deviations between the actual and predicted values.

- **Matrix** E**:** This is an $N \times N$ histogram matrix of expected outcomes, calculated under the assumption that there is no correlation between actual and predicted labels. It is derived as the outer product of the histogram vectors of actual and predicted outcomes, normalized so that E and O have the same sum.

The **Quadratic Weighted Kappa** is then calculated as:

$$\kappa = 1 - \frac{\sum_{i,j} W_{i,j} \cdot O_{i,j}}{\sum_{i,j} W_{i,j} \cdot E_{i,j}}.$$

Interpretation: The metric reflects how much better the agreement between the actual and predicted values is compared to random predictions. A higher κ value indicates better predictive performance. It is especially useful in ordinal classification problems where the relative distance between labels carries significance.

3 EDA - Exploratory Data Analysis

3.1 Data Summary

The training dataset consists of two distinct data types:

- **Tabular data:** 3,960 rows and 82 features.
- **Time series data:** 996 rows, corresponding to 996 participants equipped with wrist-worn activity trackers.

3.1.1 Tabular Data

The tabular data contains a variety of features spanning different domains, summarized below:

- `Demographics:` Includes information on the age and sex of participants.
- `Internet Use:` Number of hours spent using computers or the internet daily.
- `Children's Global Assessment Scale:` A numeric scale rating the general functioning of youths under 18.
- `Physical Measures:` Measurements such as blood pressure, heart rate, height, weight, waist, and hip circumferences.
- `FitnessGram Vitals and Treadmill:` Cardiovascular fitness metrics using the NHANES treadmill protocol.
- `FitnessGram Child:` Physical fitness parameters include aerobic capacity, muscular strength, endurance, flexibility, and body composition.
- `Bio-electric Impedance Analysis:` Body composition measures like BMI, fat percentage, muscle mass, and water content.

- `Physical Activity Questionnaire`: Information on vigorous physical activities performed by children in the past 7 days.
- `Sleep Disturbance Scale`: Categorization of sleep disorders in children.
- `Actigraphy`: Objective measures of physical activity using a research-grade biotracker.
- `Parent-Child Internet Addiction Test (PCIAT)`: A 20-item scale evaluating compulsive internet use, escapism, and dependency behaviors. The primary target field, PCIAT-PCIAT_Total, is the basis for deriving the competition's target variable SII.

Each participant is identified uniquely by the id field. Figure 1 is a summary of the key statistics for the tabular training data features.

```
display(train.head())
print(f"Train shape: {train.shape}")
```

	id	Basic_Demos-Enroll_Season	Basic_Demos-Age	Basic_Demos-Sex	CGAS-Season	CGAS-CGAS_Score	Physical-Season	Physical-BMI	Physical-Height	Physical-Weight	...	PCIAT-PCIAT_18	PCIAT-PCIAT_19	PCIAT-PCIAT_20	PCIAT-PCIAT_Total	SDS-Season	SDS-SDS_Total_Raw	SDS-SDS_Total_T	PreInt_EduHx-Season
0	00008ff9	Fall	5	0	Winter	51.0	Fall	16.877316	46.0	50.8	...	4.0	2.0	4.0	55.0	NaN	NaN	NaN	Fall
1	000fd460	Summer	9	0	NaN	NaN	Fall	14.035590	48.0	46.0	...	0.0	0.0	0.0	0.0	Fall	46.0	64.0	Summer
2	00105258	Summer	10	1	Fall	71.0	Fall	16.648696	56.5	75.6	...	2.0	1.0	1.0	28.0	Fall	38.0	54.0	Summer
3	00115b9f	Winter	9	0	Fall	71.0	Summer	18.292347	56.0	81.6	...	3.0	4.0	1.0	44.0	Summer	31.0	45.0	Winter
4	0016bb22	Spring	18	1	Summer	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 82 columns

```
Train shape: (3960, 82)
```

Figure 1: Summary statistics of tabular training data features

3.1.2 Time Series Data

The time series data provides temporal observations for each participant, with features recorded at regular intervals. Key features include:

- `id`: Identifier linking the time series to participants in the tabular data.
- `step`: Timestep for each observation.
- `X, Y, Z`: Accelerometer readings (in g) along the three standard axes.
- `enmo`: Euclidean Norm Minus One of the accelerometer signals, indicative of motion levels.
- `anglez`: Derived angle of the arm relative to the horizontal plane.
- `non-wear_flag`: Indicates whether the wristwatch was worn (0) or removed (1), based on GGIR definitions.
- `light`: Ambient light level in lux.
- `battery_voltage`: Battery voltage of the device (in mV).

- `time_of_day`: Start time of a 5-second sampling window, formatted as
- `weekday`: Day of the week (1 = Monday, 7 = Sunday).
- `quarter`: Quarter of the year (1 to 4).
- `relative_date_PCIAT`: Number of days since the PCIAT test was administered, with negative values for data collected before the test.

3.2 Target Distribution Analysis

To begin, we examine the distribution of the **Severity Impairment Index (SII)**. The data revealed two significant issues:

1. **Missing Values:** A substantial portion of the data is incomplete.
2. **Class Imbalance:** Over 70% of the records belong to the "None" (class 0) category or are missing (`NaN`), while the "Severe" (class 3) category constitutes less than 1% of the dataset.

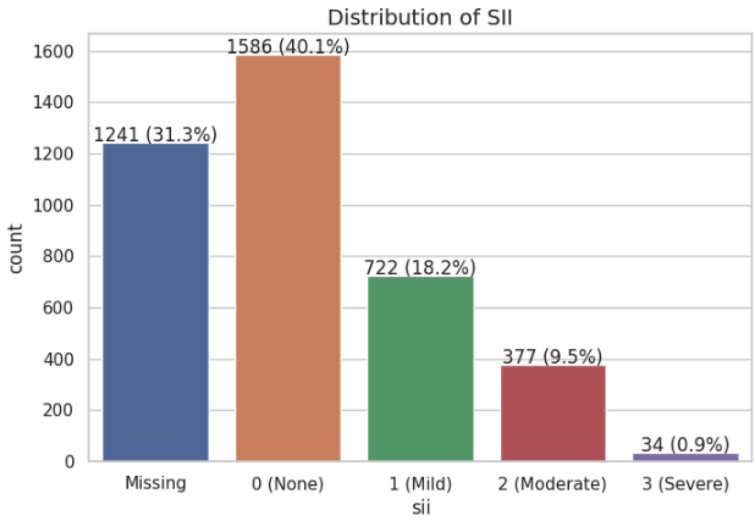

Figure 2: SII Distribution

The SII is calculated using the `PCIAT_Total` score, derived from responses to 20 questions in the PCIAT questionnaire. However, some participants did not provide answers to all questions, resulting in inaccurate or incomplete `PCIAT_Total` calculations. For instance, participant *id 93* (shown in Figure 3) left all questions unanswered, leading to an `SII` classification of 0 ("None"). However, this classification may not accurately reflect the participant's internet use behavior.

Handling Uncertain Records

To address these uncertainties, we proposed the following approach:

1. **Assume Maximum Scores for Missing Answers:** Missing responses were assigned the maximum score of 5 for each unanswered question.
2. **Recalculate the SII:** Using the adjusted scores, we recalculated the `PCIAT_Total` and determined the new SII category.
3. **Flag Inconsistent Results:** If the recalculated SII differed from the original SII, the record was marked as **uncertain**.
4. **Set Uncertain Records to Missing:** All flagged uncertain records had their SII values updated to `NaN` to avoid bias from unreliable data.

	PCIAT-PCIAT_01	PCIAT-PCIAT_02	PCIAT-PCIAT_03	PCIAT-PCIAT_04	PCIAT-PCIAT_05	PCIAT-PCIAT_06	PCIAT-PCIAT_07	PCIAT-PCIAT_08	PCIAT-PCIAT_09	PCIAT-PCIAT_10	PCIAT-PCIAT_11	PCIAT-PCIAT_12	PCIAT-PCIAT_13	PCIAT-PCIAT_14	PCIAT-PCIAT_15	PCIAT-PCIAT_16	PCIAT-PCIAT_17	PCIAT-PCIAT_18	PCIAT-PCIAT_19	PCIAT-PCIAT_20	PCIAT-PCIAT_Total	sii
24	2.000000	2.000000	3.000000	1.000000	2.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000	1.000000	2.000000	1.000000	2.000000	nan	2.000000	30.000000	0.000000
93	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	0.000000	0.000000
104	5.000000	2.000000	4.000000	2.000000	nan	2.000000	2.000000	2.000000	1.000000	2.000000	1.000000	1.000000	2.000000	2.000000	3.000000	3.000000	3.000000	3.000000	3.000000	2.000000	45.000000	1.000000
141	1.000000	2.000000	4.000000	2.000000	2.000000	2.000000	1.000000	3.000000	1.000000	1.000000	2.000000	0.000000	0.000000	0.000000	3.000000	0.000000	nan	0.000000	2.000000	0.000000	26.000000	0.000000
142	2.000000	2.000000	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000	1.000000	1.000000	3.000000	0.000000	2.000000	1.000000	1.000000	1.000000	1.000000	1.000000	nan	1.000000	26.000000	0.000000
270	3.000000	3.000000	4.000000	2.000000	4.000000	2.000000	1.000000	3.000000	2.000000	2.000000	4.000000	0.000000	2.000000	1.000000	4.000000	nan	2.000000	3.000000	4.000000	2.000000	48.000000	1.000000
368	2.000000	3.000000	4.000000	2.000000	5.000000	1.000000	2.000000	nan	nan	nan	2.000000	1.000000	1.000000	2.000000	2.000000	1.000000	2.000000	1.000000	nan	nan	31.000000	1.000000
592	3.000000	0.000000	3.000000	0.000000	3.000000	1.000000	0.000000	1.000000	1.000000	1.000000	2.000000	0.000000	1.000000	nan	nan	1.000000	2.000000	1.000000	1.000000	0.000000	21.000000	0.000000
724	3.000000	2.000000	4.000000	2.000000	2.000000	1.000000	0.000000	nan	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2.000000	1.000000	1.000000	3.000000	0.000000	1.000000	29.000000	0.000000
877	5.000000	5.000000	5.000000	4.000000	5.000000	0.000000	5.000000	5.000000	5.000000	5.000000	4.000000	nan	4.000000	5.000000	5.000000	1.000000	5.000000	0.000000	5.000000	5.000000	78.000000	2.000000
1706	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1911	nan	5.000000	5.000000	nan	3.000000	4.000000	0.000000	1.000000	5.000000	0.000000	4.000000	0.000000	0.000000	0.000000	5.000000	3.000000	nan	3.000000	0.000000	0.000000	38.000000	1.000000
2285	2.000000	2.000000	2.000000	1.000000	2.000000	1.000000	0.000000	2.000000	2.000000	2.000000	2.000000	0.000000	2.000000	2.000000	nan	1.000000	2.000000	nan	1.000000	1.000000	27.000000	0.000000
3037	3.000000	4.000000	4.000000	0.000000	4.000000	0.000000	0.000000	4.000000	2.000000	2.000000	4.000000	0.000000	4.000000	1.000000	4.000000	4.000000	4.000000	nan	nan	1.000000	45.000000	1.000000
3500	3.000000	3.000000	2.000000	3.000000	4.000000	4.000000	4.000000	2.000000	3.000000	3.000000	2.000000	1.000000	2.000000	2.000000	2.000000	nan	2.000000	2.000000	2.000000	1.000000	47.000000	1.000000
3672	5.000000	5.000000	1.000000	0.000000	nan	1.000000	0.000000	0.000000	0.000000	0.000000	5.000000	0.000000	5.000000	1.000000	5.000000	5.000000	3.000000	5.000000	5.000000	3.000000	49.000000	1.000000
3757	2.000000	2.000000	4.000000	0.000000	5.000000	1.000000	nan	5.000000	nan	0.000000	0.000000	0.000000	0.000000	4.000000	4.000000	5.000000	2.000000	4.000000	0.000000	4.000000	42.000000	1.000000

Figure 3: Examples of Uncertain Records with Missing Data

This process ensures that the dataset used for further analysis and modeling is more reliable by mitigating the effects of missing values and addressing inconsistencies in the SII computation.

4 EDA - Multivariable Relationship Analysis

4.1 Age, Gender, and Target Relationship

4.1.1 SII by Age

Figure 4 shows that the Severity Impairment Index (SII) increases with age, indicating that older participants tend to exhibit higher internet addiction severity. Notably, outliers are present across age groups, particularly in the "None" and "Mild" categories, suggesting atypical internet usage among some younger participants.

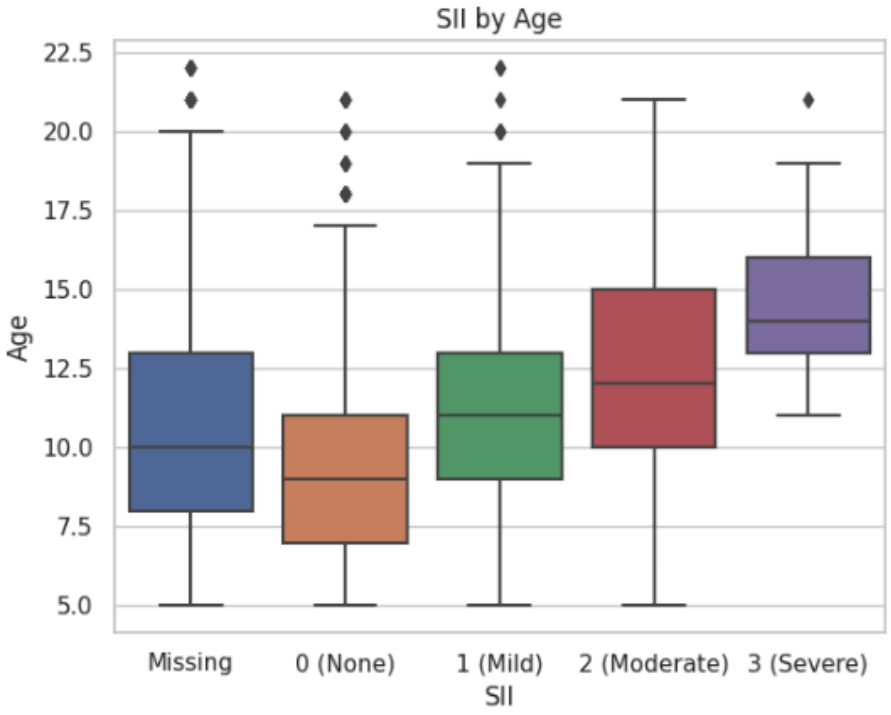

Figure 4: SII Distribution by Age

Figure 5 further indicates that children predominantly fall into the "None" and "Mild" categories, while adults are more likely to fall into higher severity categories. However, missing data is significantly higher among adults (60.2%) than children (29.4%), likely due to a smaller adult sample, which may affect the generalizability of the results.

sii Age Group	Missing	0 (None)	1 (Mild)	2 (Moderate)	3 (Severe)
Children (5-12)	858 (29.4%)	1359 (46.6%)	493 (16.9%)	203 (7.0%)	6 (0.2%)
Adolescents (13-18)	330 (34.6%)	211 (22.1%)	217 (22.8%)	169 (17.7%)	26 (2.7%)
Adults (19-22)	53 (60.2%)	16 (18.2%)	12 (13.6%)	5 (5.7%)	2 (2.3%)

Figure 5: SII Distribution by Age Group

Figure 6 shows the proportional distribution of SII severity by age group, reinforcing the trend that internet addiction severity increases with age. However, significant variability suggests individual differences.

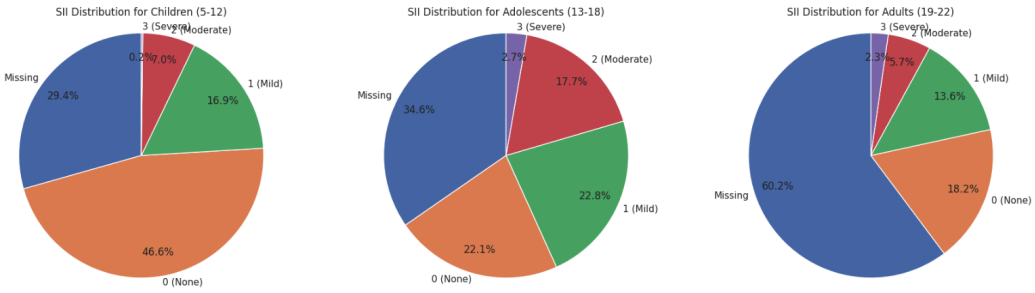

Figure 6: Proportional Distribution of SII by Age Group

4.1.2 SII by Gender

Figure 7 reveals no significant gender differences in internet addiction severity, with both males and females exhibiting similar SII distribution patterns.

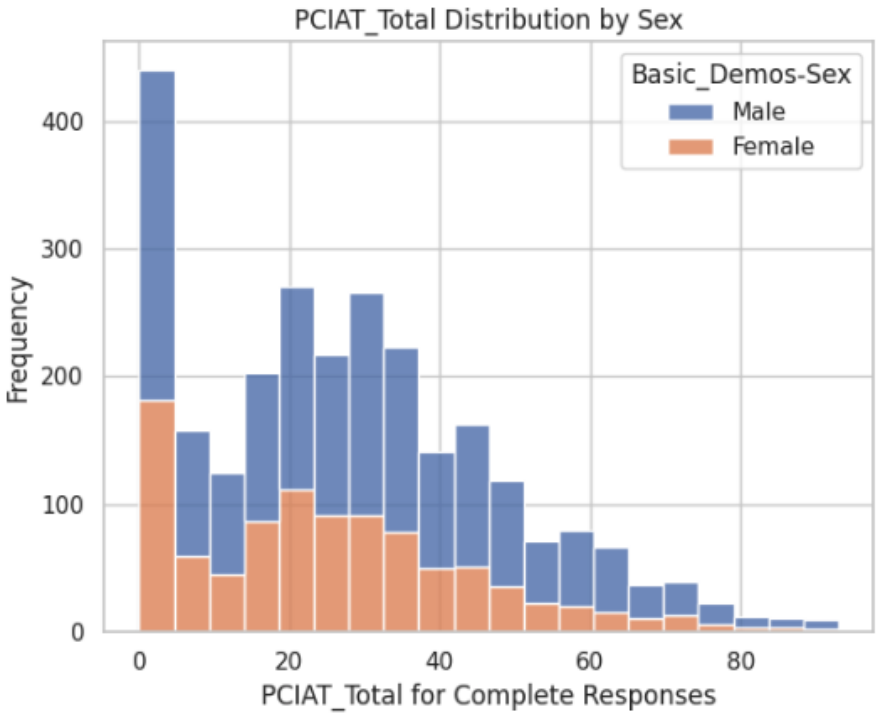

Figure 7: SII Distribution by Gender

4.2 Internet Use

The duration of internet use is a key predictor of internet addiction severity. Figure 8 illustrates that most participants use the internet for less than 1 hour

daily. Age-wise trends, shown in Figure 9, indicate that children typically use the internet for 0-2 hours, while adults report 2 or more hours per day.

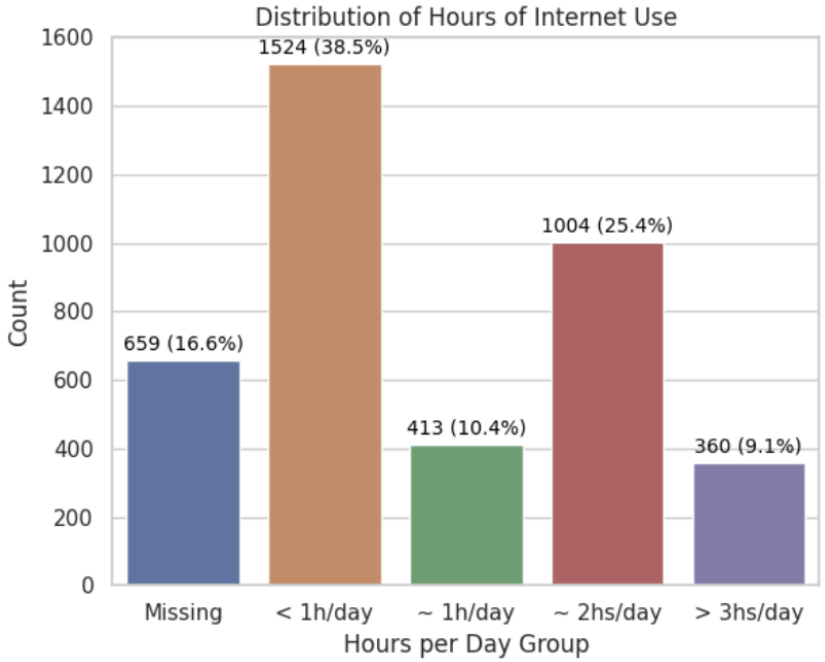

Figure 8: Distribution of Internet Usage Hours

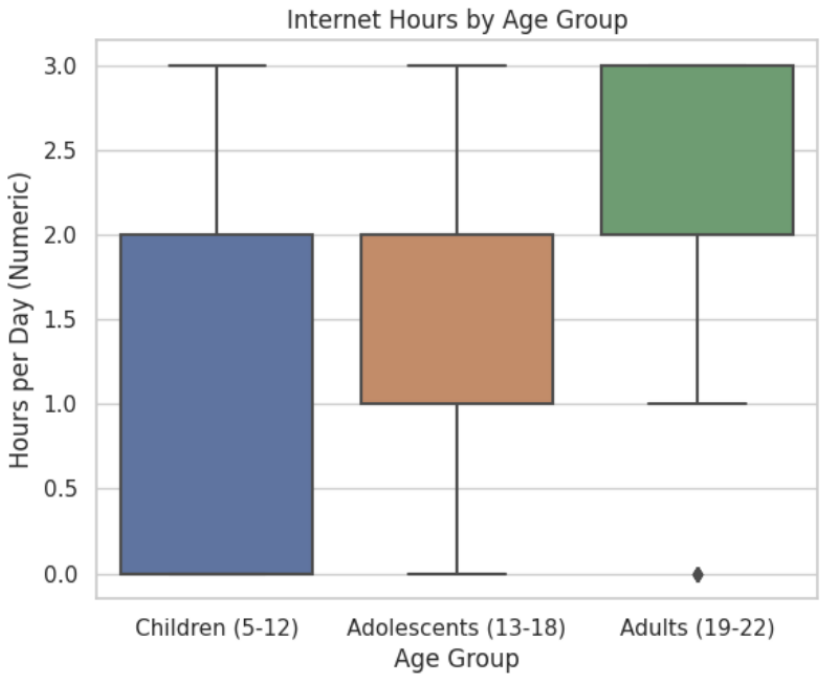

Figure 9: Internet Usage by Age Group

Figure 10 reveals a positive correlation between longer internet use and higher SII scores. However, missing data and a limited sample in severe addiction categories (SII level 3) may affect the reliability of this trend.

sii	0 (None)	1 (Mild)	2 (Moderate)	3 (Severe)
internet_use_encoded				
Missing	52 (63.4%)	15 (18.3%)	15 (18.3%)	0 (0.0%)
< 1h/day	933 (73.9%)	247 (19.6%)	78 (6.2%)	5 (0.4%)
~ 1h/day	160 (47.2%)	123 (36.3%)	54 (15.9%)	2 (0.6%)
~ 2hs/day	366 (47.2%)	251 (32.3%)	147 (18.9%)	12 (1.5%)
> 3hs/day	75 (29.0%)	86 (33.2%)	83 (32.0%)	15 (5.8%)

Figure 10: Internet Usage vs. Internet Addiction Severity

4.3 Physical Measures

4.3.1 Weight and Height

Figure 11 demonstrates that weight and height generally increase with age. However, outliers, such as a 7-year-old weighing 100 kg, may indicate data entry errors or unusual cases (e.g., obesity), which complicate data interpretation.

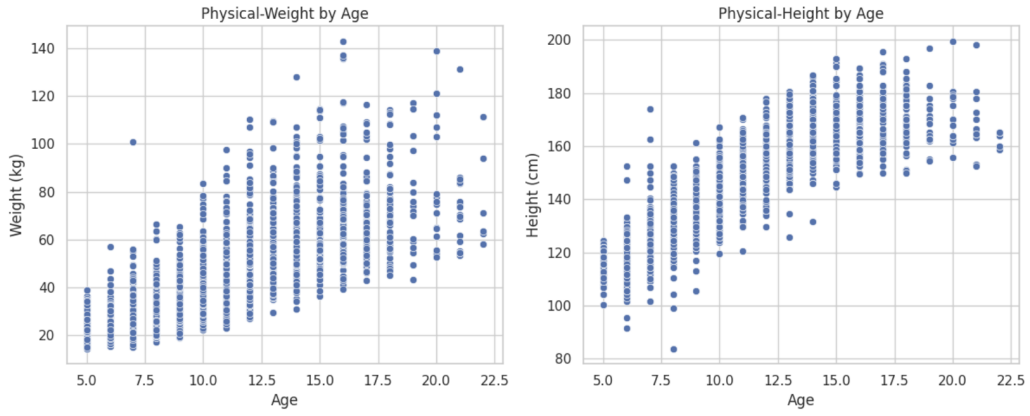

Figure 11: Relationship Between Age, Weight, and Height

4.3.2 Blood Pressure and Heart Rate

Blood pressure (BP) and heart rate data contain numerous outliers, as shown in Figure 12. For example, heart rates below 40 bpm and BP values outside the normal range (e.g., below 20 mmHg or above 200 mmHg) suggest data inaccuracies. These anomalies complicate the reliability of the measurements.

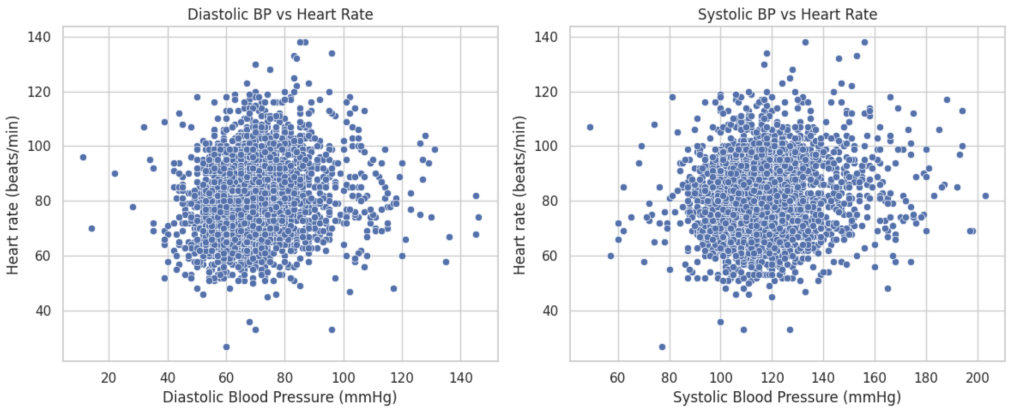

Figure 12: Blood Pressure and Heart Rate Data

Figure 13 highlights correlations between physical measurements. Height, weight, and waist circumference are positively correlated with SII, suggesting that taller, heavier individuals with larger waist circumferences may exhibit higher SII scores. Cardiovascular metrics (BP and heart rate) show high variability and weak correlations with SII.

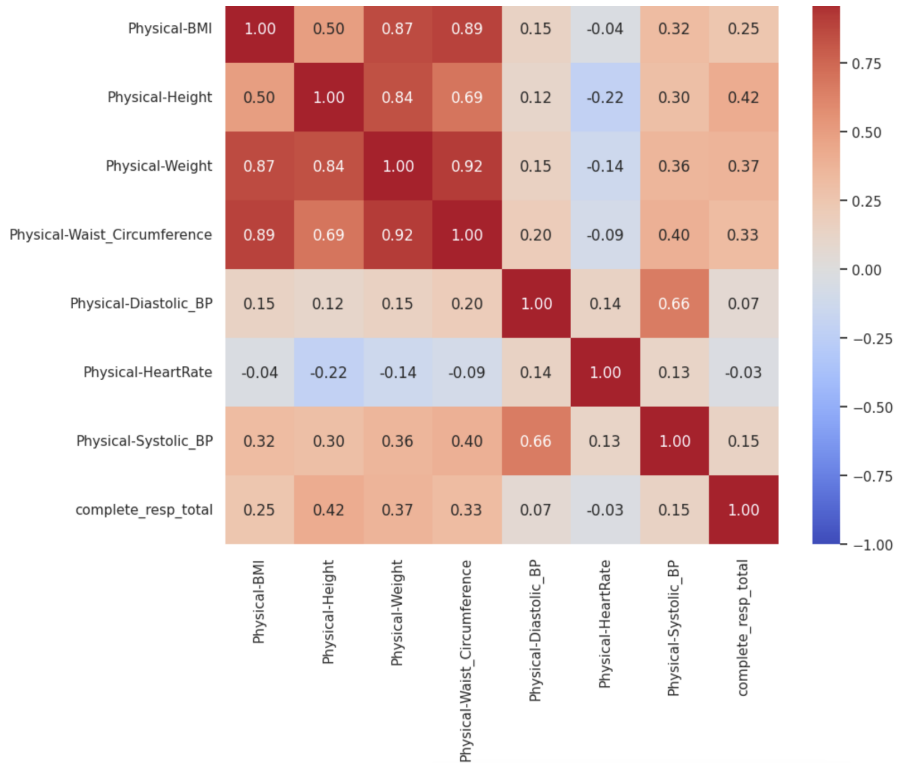

Figure 13: Correlation Matrix of Physical Measurements

4.4 Fitness Measures

Fitness metrics, such as handgrip strength and flexibility, show internal correlations, as shown in Figure 14. Notably, better performance in physical tests (e.g., sit-ups, push-ups) correlates moderately with higher PIU severity. This correlation may reflect age effects rather than a direct causal relationship between fitness and internet addiction. Inconsistent test standardization across participants may contribute to variability.

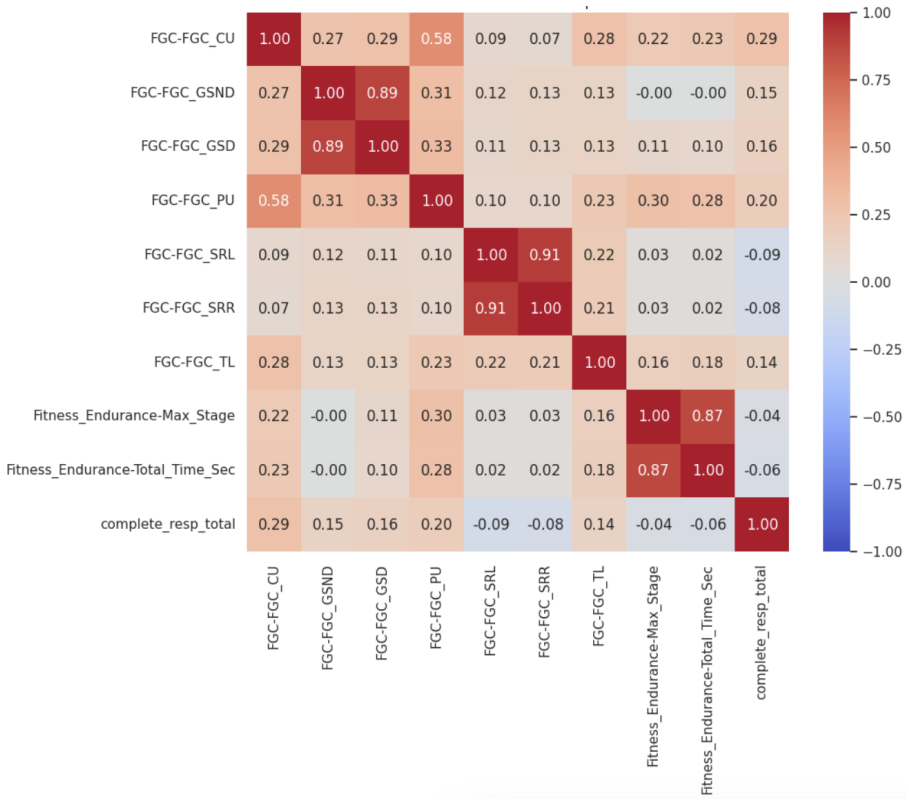

Figure 14: Correlation Matrix of Fitness Measurements

5 Data Preprocessing and Feature Engineering

5.1 Data Preprocessing

5.1.1 Processing Time-Series Data

Time-series data is loaded from Parquet files and standardized using the StandardScaler. Standardization transforms features to follow a standard normal distribution, ensuring consistency, stability, and suitability for downstream analysis. The standardized data is then converted into PyTorch tensors for compatibility with deep learning workflows.

An autoencoder, illustrated in Figure 15, is used to compress the data into low-dimensional feature vectors of size $d = 60$, effectively reducing dimensionality while retaining key information. The encoded features are re-integrated into the original dataset using their respective IDs, enriching it with additional features that enhance downstream task performance.

```python
class AutoEncoder(nn.Module):
    def __init__(self, input_dim, encoding_dim):
        super(AutoEncoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(input_dim, encoding_dim*3),
            nn.ReLU(),
            nn.Linear(encoding_dim*3, encoding_dim*2),
            nn.ReLU(),
            nn.Linear(encoding_dim*2, encoding_dim),
            nn.ReLU()
        )
        self.decoder = nn.Sequential(
            nn.Linear(encoding_dim, input_dim*2),
            nn.ReLU(),
            nn.Linear(input_dim*2, input_dim*3),
            nn.ReLU(),
            nn.Linear(input_dim*3, input_dim),
            nn.Sigmoid()
        )

    def forward(self, x):
        encoded = self.encoder(x)
        decoded = self.decoder(encoded)
        return decoded
```

Figure 15: Autoencoder encoding time-series data into feature vectors

5.1.2 Handling Missing Data

To ensure data quality and minimize model bias, a two-step approach is applied to handle missing data:

1. **Contextual Imputation with KNN:** Missing values in numerical columns are imputed using the K-Nearest Neighbors (KNN) algorithm. Each missing value is replaced by the mean value of its 5 most similar rows (nearest

neighbors), determined by feature similarity. This method preserves contextual integrity and avoids introducing arbitrary values that may distort the dataset.

2. **Exclusion of Rows with Extensive Missing Values:** Rows with a high proportion of missing values or missing target labels are removed. Retaining such rows could increase noise, introduce biases, and degrade model reliability. Removing them ensures a cleaner dataset with fewer uncertainties, leading to improved downstream results.

This dual approach combines the benefits of filling plausible values using KNN-based imputation with the elimination of heavily incomplete rows that could undermine model accuracy. By prioritizing data quality, this strategy enhances both the robustness and performance of subsequent modeling tasks.

5.2 Feature Engineering

5.2.1 Feature Redundancy Removal

Features with the suffix `"season"` are excluded, as they do not contribute significantly to predictive performance. Removing redundant features improves computational efficiency and reduces dimensionality.

5.2.2 Feature Creation

New features are generated by combining existing variables to enhance the dataset's richness and predictive power. These engineered features capture meaningful relationships and add deeper contextual insights, which can help improve model performance.

1. **Interplay Between BMI and Other Factors:**
   - **BMI_Age:** Combines BMI and age, reflecting age-related effects on body mass.
   - **Internet_Hours_Age:** Captures the interaction between internet usage and age, offering insights into behavioral patterns.
   - **BMI_Internet_Hours:** Links BMI to internet usage, exploring potential correlations between physical health and digital habits.
2. **Body Composition Ratios:**
   - **BFP_BMI:** Ratio of body fat percentage to BMI, indicating fat distribution relative to body weight.

- **FFMI_BFP:** Ratio of fat-free mass index to body fat percentage, emphasizing lean mass contributions.
- **FMI_BFP:** Fat mass index relative to body fat percentage, focusing on fat-specific metrics.

3. **Balance and Element Distribution:**

- **LST_TBW:** Ratio of lean soft tissue to total body water, reflecting water distribution in lean mass.
- **BFP_BMR:** Body fat's contribution to basal metabolic rate, showing fat's energy requirements.
- **BFP_DEE:** Contribution of body fat to daily energy expenditure, useful for assessing energy demands.
- **BMR_Weight:** Basal metabolic rate scaled by weight, analyzing metabolic efficiency.
- **DEE_Weight:** Daily energy expenditure relative to weight, reflecting energy use efficiency.

4. **Muscle, Fat, and Hydration Metrics:**

- **SMM_Height:** Skeletal muscle mass normalized by height, indicating muscle density.
- **Muscle_to_Fat:** The ratio of muscle to fat, highlighting the balance between these body components.
- **Hydration_Status:** Ratio of total body water to weight, reflecting hydration levels.
- **ICW_TBW:** Fraction of intracellular water in total body water, assessing water distribution.

5. **BMI and Cardiovascular Fitness:**

- **BMI_PHR:** Combines BMI with pulse heart rate to evaluate correlations between body composition and cardiovascular health.

```
df['BMI_Age'] = df['Physical-BMI'] * df['Basic_Demos-Age']
df['Internet_Hours_Age'] = df['PreInt_EduHx-computerinternet_hoursday'] * df['Basic_Demos-Age']
df['BMI_Internet_Hours'] = df['Physical-BMI'] * df['PreInt_EduHx-computerinternet_hoursday']
df['BFP_BMI'] = df['BIA-BIA_Fat'] / df['BIA-BIA_BMI']
df['FFMI_BFP'] = df['BIA-BIA_FFMI'] / df['BIA-BIA_Fat']
df['FMI_BFP'] = df['BIA-BIA_FMI'] / df['BIA-BIA_Fat']
df['LST_TBW'] = df['BIA-BIA_LST'] / df['BIA-BIA_TBW']
df['BFP_BMR'] = df['BIA-BIA_Fat'] * df['BIA-BIA_BMR']
df['BFP_DEE'] = df['BIA-BIA_Fat'] * df['BIA-BIA_DEE']
df['BMR_Weight'] = df['BIA-BIA_BMR'] / df['Physical-Weight']
df['DEE_Weight'] = df['BIA-BIA_DEE'] / df['Physical-Weight']
df['SMM_Height'] = df['BIA-BIA_SMM'] / df['Physical-Height']
df['Muscle_to_Fat'] = df['BIA-BIA_SMM'] / df['BIA-BIA_FMI']
df['Hydration_Status'] = df['BIA-BIA_TBW'] / df['Physical-Weight']
df['ICW_TBW'] = df['BIA-BIA_ICW'] / df['BIA-BIA_TBW']
df['BMI_PHR'] = df['Physical-BMI'] * df['Physical-HeartRate']
```

Figure 16: New feature creation to enrich the dataset

Creating new features, shown in Figure 16 improves the model's ability to discern complex patterns in the data. Features like ratios and interactions reveal underlying relationships, while dimensionality reduction through a selection of relevant columns eliminates noise, enhancing computational efficiency and predictive accuracy.

5.2.3 Feature Refiltering

To streamline the dataset and enhance model performance, feature filtering is applied. This involves:

1. **Correlation Analysis:** Features with strong correlations to the target variable are prioritized for inclusion.
2. **Domain Knowledge Integration:** Domain expertise ensures relevant features are selected.
3. **Noise Reduction via Feature Selection:** Irrelevant features are removed, simplifying the dataset and improving model performance.

This filtering method uncovers meaningful patterns and accelerates model training without introducing unnecessary complexity.

6 Modeling

6.1 Model Types

The models used in this study are as follows:

1. **LightGBM**
   - Designed for regression tasks with fast and efficient handling of large datasets.
   - Selected for its ability to optimize performance and leverage GPU acceleration effectively.
2. **XGBoost (Extreme Gradient Boosting)**
   - A high-performance boosting framework known for its computational efficiency.
   - Features sparsity-aware handling of missing values and mitigates overfitting through regularization.
3. **CatBoost (Categorical Boosting)**
   - Specifically optimized for handling categorical data without the need for one-hot encoding, preventing dimensionality explosion.
   - Offers automatic handling of missing values and adaptability to diverse datasets with minimal tuning.
4. **TabNet**
   - A deep learning architecture using attention mechanisms, designed for tabular data.
   - Balances accuracy and interpretability by optimizing key variables through feature selection in each layer.
5. **Random Forest**
   - An ensemble learning algorithm based on decision trees. Predictions are made using the average (or voting) across independent trees.
   - Robust against overfitting and performs well with both missing data and non-linear relationships.
6. **Gradient Boosting**
   - An iterative boosting method focused on reducing model errors by sequentially constructing decision trees.

- Applicable for regression and classification problems, with customizable loss functions for problem-specific optimization.

7. **Voting Regressor**
   - Combines predictions from multiple models (e.g., LightGBM, XGBoost, CatBoost, TabNet, Random Forest, and Gradient Boosting).
   - Uses averaging or majority voting for final predictions, leveraging strengths of individual models.

This diverse model selection and combination leverage strengths from both traditional machine learning methods and advanced architectures, balancing interpretability, flexibility, and predictive power to achieve robust results.

6.2 Evaluation Protocol

6.2.1 Metric

The evaluation relies on the **Quadratic Weighted Kappa (QWK)**, an agreement metric tailored for ordinal classification tasks. QWK effectively penalizes discrepancies based on the distance between predicted and true labels, making it highly suitable for assessing severity predictions.

6.2.2 Cross-Validation

Stratified K-Folds cross-validation ensures the data is divided into training and validation sets with balanced class distributions in each fold. This protocol helps prevent overfitting and provides reliable out-of-fold (OOF) predictions. Key aspects include:

- **Stratified K-Fold Cross-Validation:** The dataset is split into `n_splits` folds while maintaining class distributions in each fold.
- **Out-of-Fold (OOF) Predictions:** Generated to provide an unbiased estimate of model performance on unseen data.
- **Fold-wise Analysis:** Validation scores are averaged across folds, with individual fold scores tracked to detect performance variability.

6.3 Threshold Optimization

To map continuous model outputs to discrete ordinal categories (*None*, *Mild*, *Moderate*, *Severe*), threshold optimization is performed:

- The `minimize` function from `scipy.optimize` is used to find optimal decision thresholds.

- Optimization aligns model predictions with the QWK metric, improving ordinal classification accuracy.
- Final tuned thresholds are applied to OOF predictions and test predictions.

6.4 Hyperparameter Tuning

Fine-tuning model hyperparameters is conducted to balance bias and variance:

LightGBM

Key parameters tuned:

- `learning_rate`, `max_depth`, and `num_leaves`: Control learning speed and model complexity.
- `feature_fraction` and `bagging_fraction`: Introduce randomness to improve generalization.
- `lambda_l1` and `lambda_l2`: Provide regularization for optimal bias-variance trade-off.

XGBoost

Key parameters tuned:

- `learning_rate`, `max_depth`, and `n_estimators`: Control iterative learning and model depth.
- `subsample` and `colsample_bytree`: Enhance randomness to reduce overfitting.
- `reg_alpha` and `reg_lambda`: Manage regularization to stabilize training.

CatBoost

Key parameters tuned:

- `learning_rate`, `depth`, and `iterations`: Optimize learning progression and model size.
- `l2_leaf_reg`: Regularization factor to prevent overfitting.
- `task_type`: Configured to maximize efficiency through GPU or CPU training.

TabNet

Key parameters tuned:

- `n_d` and `n_a`: Widths of decision and attention layers.
- `lambda_sparse`: Promotes feature sparsity for interpretability.
- `mask_type`: Defines the attention sparsity mechanism.
- `learning rate schedule`: Handles adaptive learning with early stopping mechanisms.

6.5 Ensemble of Ensembles Learning and Final Submission

To combine the strengths of individual models, an ensemble of ensembles approach is employed:

- **Weighted Voting Regressor:** Combines predictions from LightGBM, XGBoost, CatBoost, and TabNet using weighted averaging.
- **Kappa Optimizer:** Fine-tunes prediction thresholds to align outputs with the QWK metric.
- **Submission Preparation:** Test set predictions are formatted into the required CSV layout, including `id` and `sii` columns.

This integrated workflow ensures robust performance, emphasizing consistency and accuracy across all modeling stages.

7 Qualitative Hypothesis to Win Contest

The baseline we presented is a robust and theoretically effective machine learning framework. It thoroughly incorporates the essential components of a comprehensive **Machine Learning Pipeline**, including:

- **Data Processing**: Utilizing time-series techniques and balanced imputation strategies for processing.
- **Feature Engineering**: Creating, filtering, and refining features for meaningful model input.
- **Cross-Validation and Out-of-Fold Evaluation**: Ensuring model reliability and unbiased assessment.
- **Ensemble of Ensembles**: Combining predictions from multiple models for improved accuracy.
- **Hyperparameter Tuning**: Balancing bias-variance trade-offs to optimize model performance.
- **Thresholding Technique**: Mapping continuous outputs to discrete ordinal categories effectively.

Our baseline achieved a remarkably high quantitative result, with a QWK score of **0.498** on the public test, placing us in a competitive rank during the public phase of the contest. However, we quickly recognized significant limitations in the dataset and evaluation protocol, necessitating a nuanced and adaptive approach for further improvement.

7.1 Observed Challenges and Limitations

Despite its success, our baseline revealed potential vulnerabilities stemming from the limited size and skewed distribution of the competition's test datasets:

1. **Test Case Scarcity**: The public test set contained only 20 samples, and the private test set had 52 samples. This limited sample size restricts robust evaluation and makes it challenging to generalize conclusions.
2. **Severe Class Imbalance**: Class 3 comprised merely 1% of the overall dataset. As a result, the test sets would likely contain few samples from class 2 and none from class 3. This contradicts the competition's primary goal to classify severity levels of Internet addiction effectively.

3. **Limited Practical Significance of Easy Predictions**: Distinguishing between non-addicted (class 0) and addicted (class 1) is relatively straightforward and feasible using manual observations. The true challenge lies in accurately identifying severity levels (class 1 vs. class 2 vs. class 3), a task requiring advanced machine learning techniques.

7.2 Hypothesis and Adaptive Strategies

Given these observations, we hypothesize potential discrepancies between the training set, public test set, and private test set distributions. This hypothesis prompted the development of a *backup baseline*, mirroring the original pipeline but prioritizing the accurate prediction of classes 2 and 3 over overall accuracy.

7.3 Blending Baselines for Robust Predictions

Given the uncertainty surrounding our hypothesis, we adopt a conservative strategy by blending the predictions of the original baseline and the backup baseline:

- The **original baseline** exhibited a left-skewed prediction bias, excelling at distinguishing between class 0 and class 1.
- The **backup baseline** demonstrated a right-skewed prediction bias, better-identifying class 2 and class 3.

The blended baseline combined predictions from both approaches by averaging their outputs, creating a balanced prediction spectrum. While this method is not perfectly aligned with either extreme, it minimized prediction error and maximized robustness under the competition's **ambiguous evaluation conditions**. This approach sought to reduce discrepancies across all classes, thereby optimizing the QWK score.

7.4 Results and Reflections

The blended baseline demonstrated its effectiveness by achieving a QWK score of **0.443** on the private test, securing a high rank and earning our team a **bronze medal** in the competition that first our Kaggle competition. While the score fell slightly short of our expectations, it highlighted the following strengths of our process:

- **Baseline Updates**: The iterative enhancement of our pipeline ensured the continuous improvement of our result.

Figure 17: Certificate of competition

- **Quantitative and Qualitative Balance**: Combining empirical evaluation with strategic qualitative hypothesis yielded an unexpectedly competitive result.
- **Team Effort and Guidance**: Our collaborative teamwork and the invaluable insights PhD Tiep Vinh Nguyen provided informed critical decisions and ensured success.

7.5 Conclusion and Acknowledgements

The success of our adaptive and comprehensive baseline highlights the significance of combining **Quantitative Evaluation with Qualitative Hypothesis**. Our ability to effectively adapt challenging dataset conditions underscores our *team's expertise, experience, and analytical ability.*

We extend our heartfelt gratitude to PhD Tiep Vinh Nguyen for his mentorship, whose guidance and keywords provided a diverse perspective and informed critical decisions. We also express appreciation to all team members for their tireless dedication and collaborative spirit. This competition has been an invaluable learning experience, motivating us to strive for excellence in future endeavors.