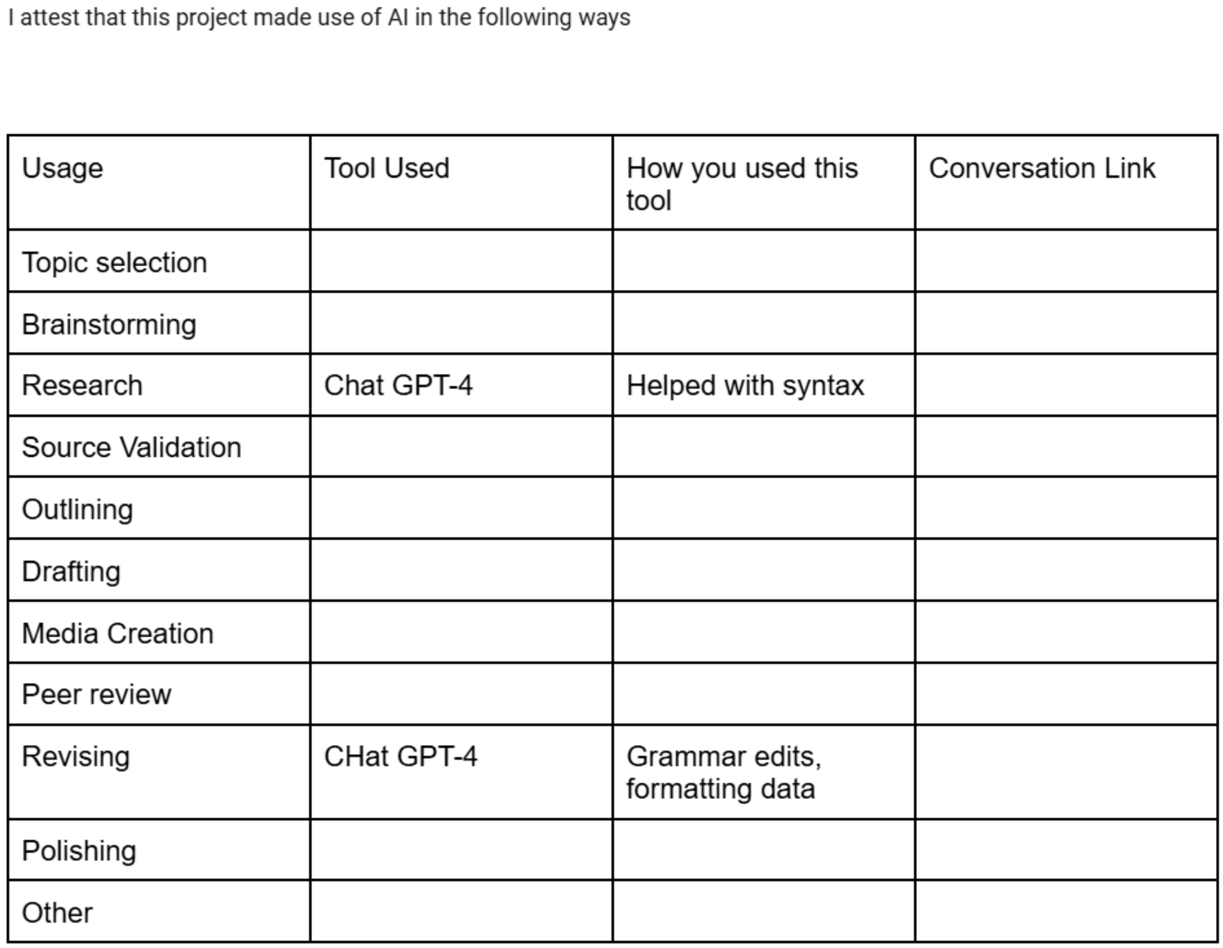
# Predicting Problematic Internet Use

Group 1: Connor McManus, Sean Shen, Victoria Uchman, Barrett White

## Abstract

Problematic internet use (PIU) has emerged as a critical concern in contemporary society, particularly among youth, due to its adverse implications for mental health and learning. This study utilizes data from the Healthy Brain Network (HBN), a comprehensive dataset of approximately 5,000 participants aged 5–22, to predict the Severity Impairment Index (SII), a categorical measure of PIU severity ranging from None to Severe. Leveraging physical activity metrics, demographic variables, and behavioral markers, we employ machine learning techniques such as multinomial logistic regression, random forests, and XGBoost for classification. We address challenges related to high-dimensional and incomplete data through imputation and feature selection techniques, ultimately identifying key predictors of PIU severity, such as symptom severity scores, physical activity levels, and internet usage behavior. The results provide insights for clinicians and policymakers, with potential applications in designing targeted interventions to mitigate the impacts of PIU.



I attest that this project made use of AI in the following ways

| Usage | Tool Used | How you used this tool | Conversation Link |
|---|---|---|---|
| Topic selection | | | |
| Brainstorming | | | |
| Research | Chat GPT-4 | Helped with syntax | |
| Source Validation | | | |
| Outlining | | | |
| Drafting | | | |
| Media Creation | | | |
| Peer review | | | |
| Revising | CHat GPT-4 | Grammar edits, formatting data | |
| Polishing | | | |
| Other | | | |

### Statement Approving All Member Contribution

All members in Team 1 contributed to the final project, report, and presentation in equal portions. Members met weekly and corresponded regularly to ensure work was distributed accordingly and everyone had a strong understanding of the project's concept and reproducibility. Portions were divided and agreed upon before completion and discussed in meetings to approve work quality.

## Introduction

The digital age has transformed how individuals, particularly youth, engage with the internet, leading to significant concerns about problematic internet use (PIU). PIU is characterized by behaviors such as compulsivity, escapism, and dependency, and has been linked to adverse mental health outcomes and hindered learning. Understanding the predictors of PIU is crucial for early identification and intervention, particularly among vulnerable populations. This study uses the Healthy Brain Network (HBN) dataset, which integrates clinical, physical activity, and internet usage data from participants aged 5–22, to predict the Severity Impairment Index (SII). The SII categorizes PIU into four levels: None, Mild, Moderate, and Severe. Our objectives are to identify key demographic, physical, and behavioral predictors of PIU and to provide interpretable insights into their relationships with mental health outcomes. To achieve this, we employ machine learning models tailored for ordinal classification, addressing challenges such as missing data and high-dimensionality through advanced imputation and feature selection techniques. The findings are anticipated to advance our understanding of PIU, informing clinical practices and public health strategies aimed at mitigating its impacts.

## Data Description

The Healthy Brain Network (HBN) dataset consists of approximately 5,000 participants aged 5–22, providing a rich source of data on clinical, demographic, and behavioral measures. The dataset includes two primary components:

1. Tabular Data: Contains demographic information, physical activity metrics, and internet usage behavior.
    - Notable instruments include:
        - Demographics: Age, sex.
        - Physical Measures: Height, weight, BMI, and cardiovascular fitness.
        - Internet Use: Daily hours of internet/computer usage and scores from the Parent-Child Internet Addiction Test (PCIAT).
            - To understand question types in PCIAT or to take the test yourself: https://www.healthyplace.com/psychological-tests/parent-child-internet-addiction-test
        - Sleep and Activity: Sleep disturbances and vigorous physical activity participation.
    - The target variable, SII, is derived from the PCIAT_Total score and categorizes participants into four PIU levels (0 = None, 1 = Mild, 2 = Moderate, 3 = Severe).
2. Time-Series Data: Includes accelerometer (actigraphy) data from wrist-worn devices, capturing physical activity through measures such as ENMO and ambient light levels. Challenges in this dataset include substantial missing data, particularly in the tabular component, and the high dimensionality of features. Missing values are addressed using imputation methods like KNN and MICE, while feature selection is applied to identify significant predictors.

## Quadratic Weighted Kappa

Quadratic Weighted Kappa(QWK) was used as the primary evaluation metric in this project. QWK is a measure of the agreement between two ordinally scaled samples. This metric varies from 0 (random agreement) to 1 (complete agreement). In the event that there is less agreement than expected by chance, the metric may go below 0. This metric was used because it is ideal for ordinal variables and it takes into account the distance between the predicted and actual value when evaluating accuracy. This is ideal for our prediction needs since we would rather predict a 2 when it is actually 3 than predict 0. It can be calculated with the formula below,

$$QWK = 1 - \frac{\Sigma_{i,j} W_{i,j} O_{i,j}}{\Sigma_{i,j} W_{i,j} E_{i,j}}$$

$O$ is the n x n observed agreement matrix, which records the actual frequency of occurrences for each combination of predicted and true ratings. $W$ is the n x n weight matrix, which assigns penalties to disagreements based on their distance from perfect agreement. $E$ is the n x n expected agreement matrix, which represents the agreement expected by chance based on the marginal distributions of the ratings.

## Imputations

### Median/Mode

Median/mode imputation considers only the predictor in which a missing value occurs. If this predictor is numerical, then the median of the predictor is imputed. If it is categorical, then the mode is imputed.

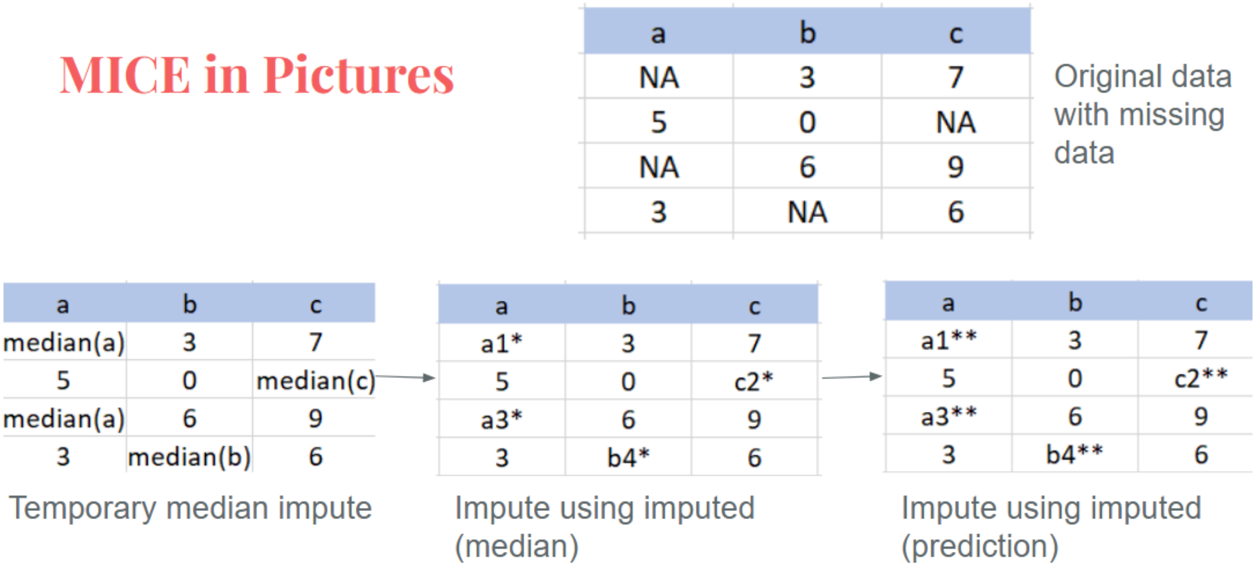
### KNN Imputation

In KNN imputation, for each missing value in the dataset, the algorithm calculates the distance between the incomplete observation and all other observations that are not missing. It then imputes a a value based on the points it is “nearest” to in terms of Euclidean Distance. For numeric variables, the mean of the values of the k nearest neighbors is imputed, and for categorical variables the mode is imputed. The main difference between KNN imputation and simple median/mode imputation is that median/mode imputation only takes the predictor of interest (where the missing value occurs) into account, while KNN imputation considers all other predictors as well when calculating distance.

### MICE imputation

Multivariate Imputation with Chained Equations, also known as MICE, is an imputation technique that integrates predictive modeling to impute missing values in a dataset. Given a dataset with missing values, missing values of a column will be imputed using predicted values from a model built on other columns. In the first iteration, the algorithm temporarily imputes all the missing values with a simple estimate such as the mean/median for continuous columns, or mode for categorical columns. By doing so, there will not be issues with missing values when creating prediction models. Then where there were missing values, replace the simple estimates (mean, median, mode) with the predicted values for those missing values by the model created from the other columns. Repeat this imputation with modeling for every other column. Supposedly, the data is better than before since the algorithm takes into the influences of other columns in the imputation of an arbitrary column. This counts as the end of the first iteration. The algorithm then continues with more iterations using the imputed data and predictive modeling to re-predict the missing values until they converge or stabilize, meaning that there are no significant changes in the imputed data between the last iteration and the current iteration. Different choices for predictive modeling such as random forest and predictive mean matching can used for imputation.

Strengths of this method of imputation is that it is quite robust since it utilizes predictive modeling to fill in missing data and allows different modeling methods to do so. A weakness of this method of imputation is that it is computationally very expensive because of the many predictive models made and many iterations of the imputation. Another weakness of this algorithm is that there is no guarantee of convergence for the imputed missing data after a certain number of iterations.



MICE in Pictures

| a | b | c |
|---|---|---|
| NA | 3 | 7 |
| 5 | 0 | NA |
| NA | 6 | 9 |
| 3 | NA | 6 |

Original data with missing data

| a | b | c |
|---|---|---|
| median(a) | 3 | 7 |
| 5 | 0 | median(c) |
| median(a) | 6 | 9 |
| 3 | median(b) | 6 |

Temporary median impute

| a | b | c |
|---|---|---|
| a1* | 3 | 7 |
| 5 | 0 | c2* |
| a3* | 6 | 9 |
| 3 | b4* | 6 |

Impute using imputed (median)

| a | b | c |
|---|---|---|
| a1** | 3 | 7 |
| 5 | 0 | c2** |
| a3** | 6 | 9 |
| 3 | b4** | 6 |

Impute using imputed (prediction)

This is an intuitive illustration of the MICE imputation. The * means the iteration number of prediction for the given missing data. The more *s there are, the more times the same missing values has been calculated using predictive modeling.

Our group used Random Forest as the method for predictive modeling in MICE. The default parameters for this is 10 trees and $\sqrt{p}$ for the subset of predictors.

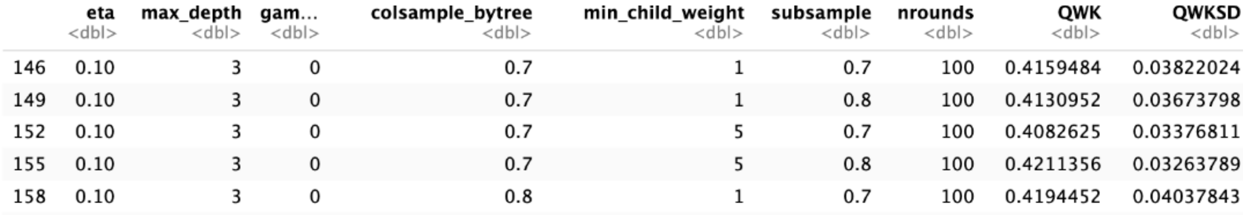
# Models

## XGBoost

One of the models we produced was XGBoost. XG Boosting is an extension of gradient boosting, which in itself is an extension of traditional boosting. Traditional boosting aggregates trees based on a modified version of the same, original data - in other words, trees are grown sequentially using information from previously grown trees. Traditional boosting involves fitting a tree to the residuals of the previous model (to improve upon where the previous tree lacks). Gradient boosting instead focuses on the pseudo-residuals, which are the gradients of the loss function with respect to the predictions made by the previous tree. The pseudo-residual represents the direction in which the model's predictions need to be adjusted. XGBoost implements regularization, in our cases ridge regression, to control the complexity of the trees.

Similar to traditional boosting, we focus on tuning the number of trees, the shrinkage parameter (learning rate, ideally slow), and the number of splits at each tree (depth). In XGBoost, it is common to also tune the minimum loss reduction required to split a tree, how many features are used to build each tree, and the weights of each observation needed to be in a node. We used 5 fold cross validation to test all different combinations of all of these parameters. The maximum QWK found via this cross validation had 100 trees, a learning rate of 0.1, a depth of 3, a gamma (minimum loss reduction) of 0, a weight of 0.7 features to build each tree, and a minimum weight of 5 to be in each node. Each parameter had three possible values - any more, and the model would have been too much computationally.

XGBoost using MICE imputation produced the highest QWK of .383. Using median/mode and KNN imputations, XGBoost produced QWKs of .317 and .379, respectively.



| | eta | max_depth | gam... | colsample_bytree | min_child_weight | subsample | nrounds | QWK | QWKSD |
|---|---|---|---|---|---|---|---|---|---|
| | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> |
| 146 | 0.10 | 3 | 0 | 0.7 | 1 | 0.7 | 100 | 0.4159484 | 0.03822024 |
| 149 | 0.10 | 3 | 0 | 0.7 | 1 | 0.8 | 100 | 0.4130952 | 0.03673798 |
| 152 | 0.10 | 3 | 0 | 0.7 | 5 | 0.7 | 100 | 0.4082625 | 0.03376811 |
| 155 | 0.10 | 3 | 0 | 0.7 | 5 | 0.8 | 100 | 0.4211356 | 0.03263789 |
| 158 | 0.10 | 3 | 0 | 0.8 | 1 | 0.7 | 100 | 0.4194452 | 0.04037843 |

```r
#Create function for QWK optimization in the cross validation
set.seed(565)
qwk_summary = function(data, lev = NULL, model = NULL){
  predicted = data$pred
  actual = data$obs

  qwk_result = kappa2(data.frame(actual, predicted), weight = "squared")

  out = c(QWK = qwk_result$value)
  return(out)
}


#Matrix of possible parameter values for XGBoost
param_grid = expand.grid(
  nrounds = c(50, 100, 200),
  eta = c(0.01, 0.1, 0.3),
  max_depth = c(3, 6, 9),
  gamma = c(0, 1),
  colsample_bytree = c(0.7, 0.8),
  min_child_weight = c(1, 5),
  subsample = c(0.7, 0.8))

#Setting up the cross validation to optimize QWK
cv_control = trainControl(
  method = "cv",
  number = 5,
  savePredictions = "all",
  summaryFunction = qwk_summary,
  classProbs = TRUE)

#Creating the model
xgb_tuned_model = train(
  x = xgb_data,
  y = xgb_label,
  method = "xgbTree",
  trControl = cv_control,
  tuneGrid = param_grid,
  metric = "QWK")

#Final model using optimized parameter values
```

```r
final_model = xgboost(
  data = as.matrix(train[, -c(which(names(train) == "sii"), which(names(train) == "sii_numeric"))]),
  label = as.numeric(train$sii) - 1,
  objective = "multi:softmax",
  num_class = length(levels(train$sii)),
  nrounds = 100,
  eta = 0.1,
  max_depth = 3,
  gamma = 0,
  colsample_bytree = 0.7,
  min_child_weight = 5,
  subsample = 0.8)
```

## Random Forest

Random forest is an ensemble method that builds multiple decision trees using random subsets of variables and combines their outputs to achieve better predictive performance. We chose to attempt this method for a couple reasons. It can provide feature importance information, it is flexible in high dimensions, and can be optimized for Quadratic Weighted Kappa. Some aspects of this method we had to account for in prediction include that it treats ordinal variables as categorical and it can suffer if ordinal classes are imbalanced. To treat these weaknesses we optimized for QWK to help treat the categorical issue and we used SMOTE to deal with the imbalanced ordinal classes in our data. SMOTE (Synthetic Minority Oversampling Technique) is a method to address class imbalance by generating synthetic samples for the minority class through interpolation between existing minority class instances and their nearest neighbors. This technique increases the representation of the minority class in the dataset, improving model performance without simply duplicating existing cases. Randomized Search Cross Validation was then used to tune hyperparameters with the goal of maximizing Quadratic Weighted Kappa. By leveraging cross-validation, the process ensured an efficient and robust selection of hyperparameters for optimal model performance based on QWK. The best hyperparameters selected were: model__n_estimators: 200, model__min_samples_split: 5, model__min_samples_leaf: 4, model__max_depth: None, model__bootstrap: True}

This yielded a 5 fold cross validation mean QWK of 0.4204

## Stepwise QWK

We also created a stepwise QWK algorithm that performs variable selection for polr() which is a proportional odds logistic regression model. Proportional odds logistic regression is perfect for the multinomial ordinal regression since the response variable of interest (sii) is multinomial meaning it has multiple levels and ordinal meaning that it has innate ordering between the levels. polr() has several inputs: the data frame, a formula that includes the response and the predictors, and the cumulative link function such as a logistic link function. There are 5 link functions in total. The cumulative link function is used to find the cumulative probabilities of the class and the classes below that certain class ($P(Y \leq k)$). So individual probabilities would be found by subtracting the cumulative probability of the last class from the cumulative distribution of the current class ($P(Y = 2) = P(Y \leq 2) - P(Y \leq 1)$).

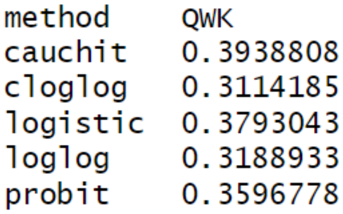
The algorithm uses QWK as the direct optimizing criterion instead of AIC and BIC. But unlike AIC and BIC, QWK does not balance model fit on data and model complexity which is why there is a k-fold cross-validation at every step. The stepwise QWK algorithm starts with no predictors in a set, then tries to add all of the predictors and remove all the predictors that are in the set. When a predictor is added or removed from the set, a polr() model is created and k-fold cross-validated to check the QWK of all 5 link functions. The QWK of all 5 link functions is then averaged which will be the cross-validated QWK. If the cross-validated QWK improves, then this addition or removal of a predictor will be kept; otherwise, if there is no improvement in the QWK, no changes will be made to the current set of predictors and the algorithm will move on to the next predictor. This addition and removal cycle continues until there is no improvement after a cycle which signals the termination of the algorithm.

The strength of this algorithm is that it is very interpretable as it performs variable selection directly, the algorithm is intuitive to explain and implement, and it is a flexible bidirectional algorithm because predictors that were added early on can be removed from the set if it is not useful for prediction later. The weaknesses of this algorithm are that it is computationally heavy because of the k-fold cross-validation at every step and that there is not a direct parameter such as $\lambda$ in LASSO to control the model complexity.

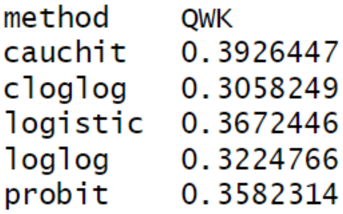
There are some potential improvements to make to this algorithm. In the QWK calculation, a penalization term can be added to the complexity so that the optimization criterion will also take the model complexity into account. This algorithm can also be improved by optimizing a single link function at a time instead of the average of all five link functions. PCA and PLS can be implemented to produce new features that can potentially improve the model prediction accuracy, but this will trade off the interpretability of the results.

The following is the pseudocode for the stepwise QWK function.

```
#pseudocode for stepwise qwk

stepwise_qwk = function(data, response variable, k-fold){

  #start with an empty set of predictors
  set_pred

  #indicator variable for while loop continuation, if there is improvement
  #repeat more loops
  improved=T

  while("there is improvement from the last iteration"){
    #set improved=F

    for ("all predictors not in set_pred"){
      #add in a predictor that is not in set_pred

      #make a polr() model with the predictors in set_pred with all 5
      #cumulative link functions

      #do k-fold cross validation; if there is improvement to the average
      #QWK of all 5 link functions, keep the addition; otherwise, go to the
      #next predictor without set_pred and repeat

      #given there is improvement to the average QWK of 5 cumulative link
      #functions, set improved=T

      #record the current best set of predictors (set_pred), best model, and
      #current best QWK
    }

    for ("all predictors in set_pred"){
      #remove a predictor from current best set of predictors (set_pred)
      #from the loop that adds predictors

      #make a polr() model with the predictors in set_pred with all 5
      #cumulative link functions

      #do k-fold cross validation; if there is improvement to the average
      #QWK of all 5 link functions, keep the removal; otherwise, go to the
      #next predictor without change to set_pred and repeat

      #given there is improvement to the average QWK of 5 cumulative link
      #functions, set improved=T

      #record the current best set of predictors (set_pred), best model, and
      #current best QWK
    }

    #if there is no improvement such that improved=F, the loop stops
  }
  #return the best set of predictors (set_pred), best model, and the best QWK
}
```



```
method     QWK
cauchit    0.3938808
cloglog    0.3114185
logistic   0.3793043
loglog     0.3188933
probit     0.3596778
```

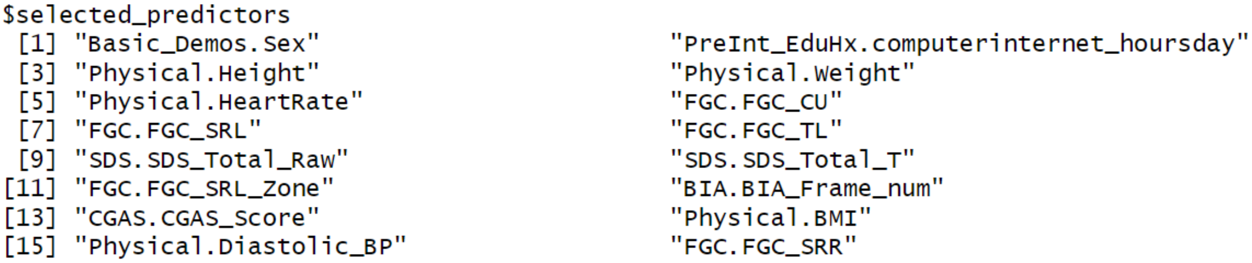


```
method     QWK
cauchit    0.3926447
cloglog    0.3058249
logistic   0.3672446
loglog     0.3224766
probit     0.3582314
```

The left table shows the training QWK by each of the cumulative link functions. The performance is evaluated by 5-fold cross-validation on the training dataset. In other words, this is the best QWK output from the Stepwise QWK function earlier.

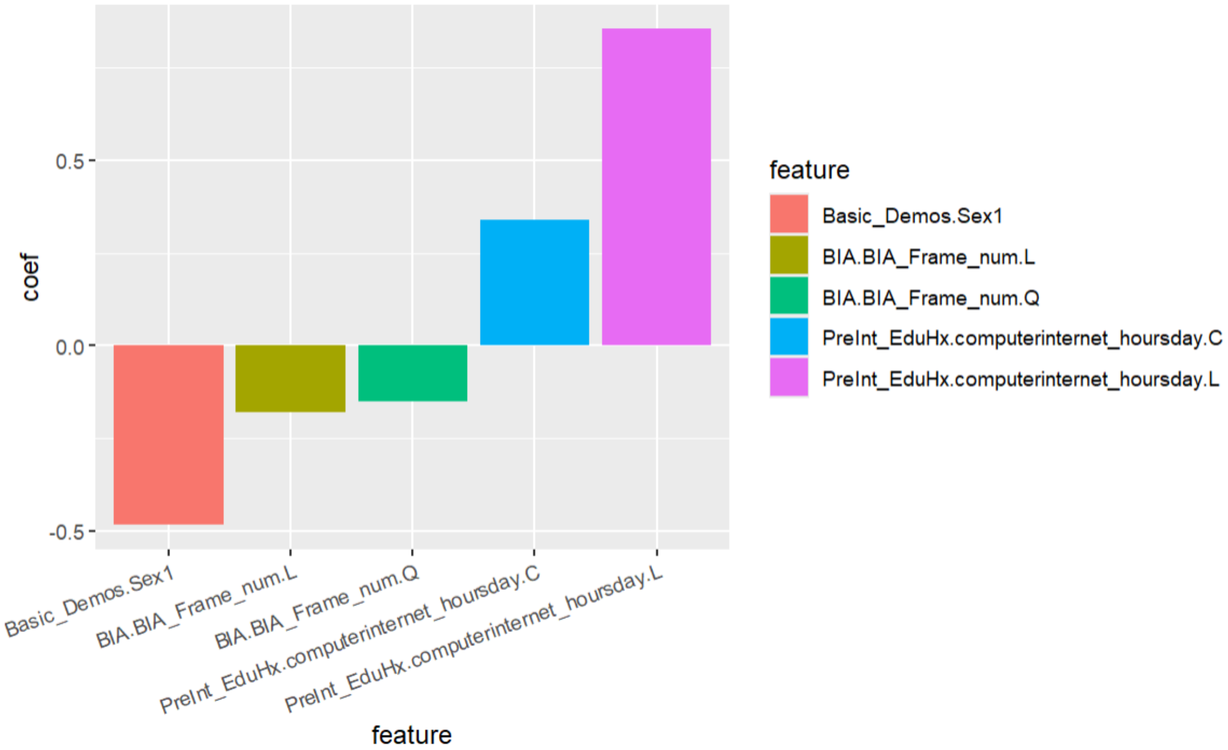
The right table shows the QWK by each of the cumulative link functions. The performance is evaluated by 5-fold cross-validation on the full dataset using the predictors selected by the stepwise QWK.

Using the validation set approach with a 9:1 split and using the formula (predictors selected by Stepwise QWK), the testing QWK is 0.3793324 with a “cauchit” link function, 0.3374896 with a “cloglog” link function, 0.3791268 with a “probit” link function, 0.3632534 with a “loglog” link function, and a 0.394014 with a “logistic” link function.



```
$selected_predictors
 [1] "Basic_Demos.Sex"                              "PreInt_EduHx.computerinternet_hoursday"
 [3] "Physical.Height"                              "Physical.Weight"
 [5] "Physical.HeartRate"                           "FGC.FGC_CU"
 [7] "FGC.FGC_SRL"                                  "FGC.FGC_TL"
 [9] "SDS.SDS_Total_Raw"                            "SDS.SDS_Total_T"
[11] "FGC.FGC_SRL_Zone"                             "BIA.BIA_Frame_num"
[13] "CGAS.CGAS_Score"                              "Physical.BMI"
[15] "Physical.Diastolic_BP"                        "FGC.FGC_SRR"
```

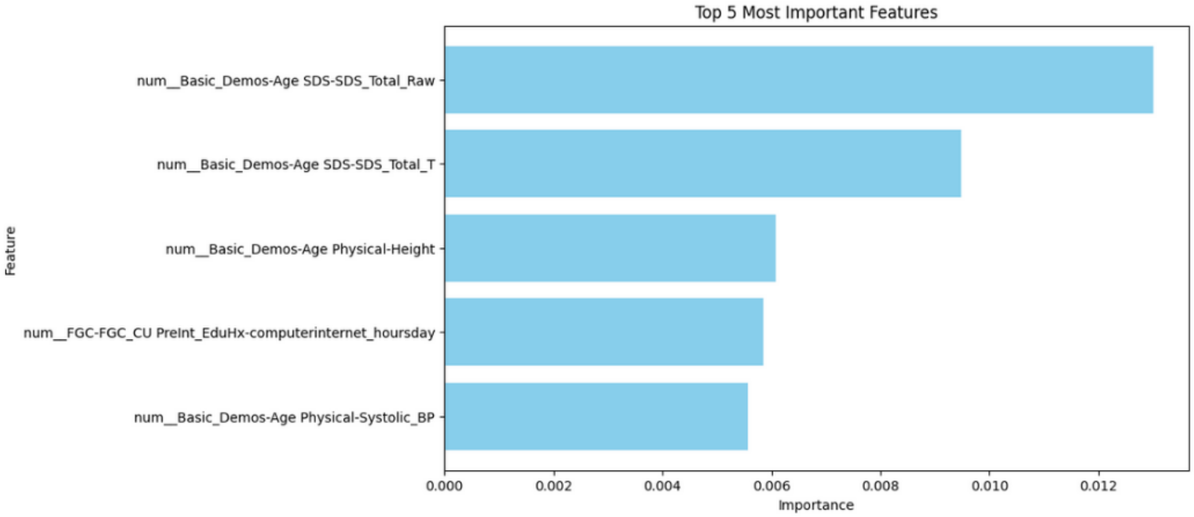
Those are the features selected by the Stepwise QWK algorithm.

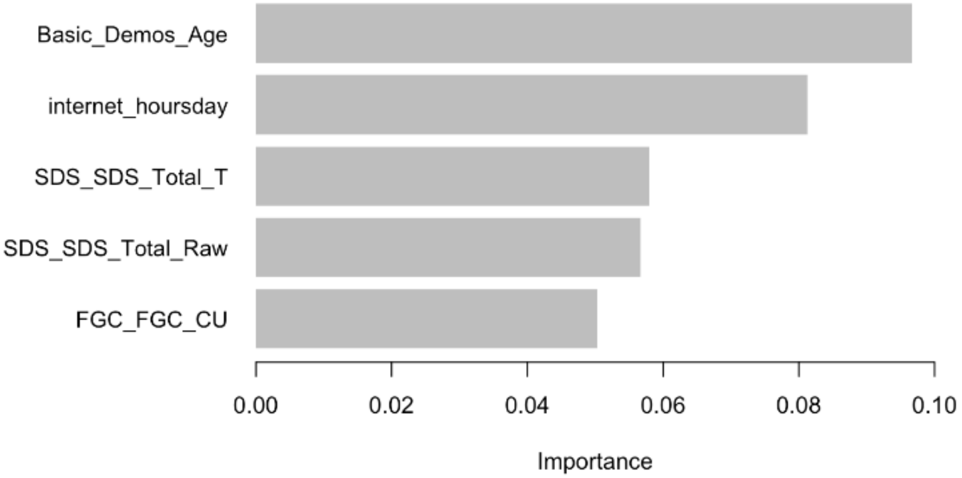


These are the top 5 features in the "cauchit" model by the absolute magnitude of coefficients selected by the Stepwise QWK function. Hours spent per day on the internet (PreInt_EduHx.computerinternet_hoursday; 0, 1, 2, 3) and body frame (BIA.BIA_Frame_num; small, medium, large) play a big role in predicting the severity impairment index as their respective has 2 terms (cubic linear and linear quadratic) in the top 5 features in the model.

## Feature Selection

Feature selection for this project was important for maximal predictive ability, replicability, and explainability of most influential features in predicting sii. To start features with 50% missing data or more were removed. This was done to reduce bias during imputation, and maintain data quality. Several feature selection methods were attempted with the data including Lasso and PCR. These methods had poor performance as regression models and maintained poor performance when implemented in other models used such as random forest and XGboost. Since these traditional feature selection methods impacted performance significantly, we opted to use variable importance plots from high predictive performance models in order to get an idea of the most influential predictors. The results of this can be seen below with the top 5 predictors being shown.





# Results

Our analysis highlights the effectiveness of machine learning techniques in predicting the Severity Impairment Index (SII). Key results are summarized below:

1. Top Predictors:

Physical and behavioral markers such as symptom severity scores, daily internet usage, height, and systolic blood pressure emerged as significant predictors of PIU severity. Sleep disturbances and vigorous physical activity participation were also important indicators.

2. Model Performance:

- Random Forest: Achieved a 5-fold cross-validated Quadratic Weighted Kappa (QWK) of 0.420, demonstrating robust performance and reliable feature importance estimation.
- XGBoost: Delivered improved predictions with fine-tuned hyperparameters, including a maximum depth of 3 and a learning rate of 0.1.

- Bidirectional Stepwise QWK: Offered interpretable outputs and insight into ordinal relationships but was computationally intensive.

3. Imputation and Data Quality:

MICE was found to be the most effective imputation method, enhancing dataset quality and reducing bias from missing values.

4. Insights:

Physical activity and internet usage behaviors were strongly correlated with PIU severity, providing actionable insights for interventions. Despite strong model performance, the complexity of PIU underscores the need for further analysis incorporating time-series data for longitudinal insights.

## Strengths of the Analysis

1. Robust Methodology:

- The use of multiple machine learning techniques (Random Forest, XGBoost, and Stepwise QWK) allowed for both interpretability and predictive accuracy. Each method contributed unique insights, with Random Forest offering feature importance and XGBoost providing fine-tuned predictions.
- The implementation of imputation techniques like MICE ensured a more complete dataset, reducing biases that often arise from missing data.

2. Relevance of Predictors:

- Identifying specific physical measures (e.g., BMI, systolic blood pressure) and internet usage behaviors as significant predictors helps bridge the gap between physical health and mental health research.

3. Innovative Metric Use:

- The adoption of Quadratic Weighted Kappa (QWK) was commendable, as it accounts for the ordinal nature of the SII variable and emphasizes class-level balance in predictions.

4. Handling Class Imbalance:

- While QWK addresses ordinal data, the SII categories may have been imbalanced, potentially skewing predictions. Applying techniques like SMOTE improved class balance and model generalizability.

## Critiques and Limitations

1. Limited Exploration of Time-Series Data:

- The accelerometer data, which could provide valuable longitudinal insights into physical activity patterns, was underutilized. A more detailed analysis using sequence models like recurrent neural networks (RNNs) or time-series clustering might reveal trends that static tabular data cannot capture.

2. Feature Engineering:

- Although effective predictors were identified, additional feature engineering could have enhanced model performance. For instance, creating interaction terms (e.g., combining sleep disturbances with daily internet usage) might reveal deeper relationships.

3. Generalizability:

- The models were trained on a specific subset of the population (youth aged 5–22). Expanding the dataset to include other demographics or testing on external datasets could validate the robustness of the findings.

# Conclusion

This study underscores the potential of machine learning in identifying and predicting the severity of problematic internet use among youth. Models like Random Forest and XGBoost proved effective in isolating key predictors, while innovative metrics like Quadratic Weighted Kappa optimized performance for ordinal classification. Despite strong results, incorporating longitudinal analysis of accelerometer data and broader feature engineering could enhance the scope and applicability of findings. The insights gained provide a foundation for tailored interventions, addressing the physical and behavioral dimensions of PIU, and paving the way for future research in mental health and digital engagement.

# Works Cited

Anderson, Eric L., Ewen Steen, and Vasileios Stavropoulos. "Internet Use and Problematic Internet Use: A Systematic Review of Longitudinal Research Trends in Adolescence and Emergent Adulthood." *International Journal of Adolescence and Youth*, vol. 22, no. 4, 2016, pp. 430–454. https://doi.org/10.1080/02673843.2016.1227716.

Child Mind Institute. “Child Mind Institute - Problematic Internet Use.” *Kaggle*, www.kaggle.com/competitions/child-mind-institute-problematic-internet-use/data. Accessed 21 Nov. 2024.

Ho, Roger C et al. “The association between internet addiction and psychiatric co-morbidity: a meta-analysis.” BMC Psychiatry vol. 14 183. 20 Jun. 2014, doi:10.1186/1471-244X-14-183

MIT School of Distance Education. “Data Imputation Techniques: Handling Missing Data in Machine Learning.” *MIT School of Distance Education Blog*, blog.mitsde.com/data-imputation-techniques-handling-missing-data-in-machine-learning/

“Parent-Child Internet Addiction Test.” HealthyPlace, www.healthyplace.com/psychological-tests/parent-child-internet-addiction-test. Accessed 1 Dec. 2024.

Prabhakaran, Selva. “Mice Imputation - How to Predict Missing Values Using Machine Learning in Python.” *Machine Learning Plus*, 14 Mar. 2023, www.machinelearningplus.com/machine-learning/mice-imputation/.

“Understanding Problematic Internet Use among Youth.” *Problematic Internet Use | Children’s Hospital Colorado*, www.childrenscolorado.org/doctors-and-departments/departments/psych/mental-health-professional-resources/primary-care-articles/internet-addiction/. Accessed 24 Nov. 2024.

“What Is Xgboost? An Introduction to XGBoost Algorithm in Machine Learning: Simplilearn.” *Simplilearn.Com*, Simplilearn, 7 Nov. 2023, www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article.