Problematic Internet Use Classification

Michael Lu
Brown University, Department of Computer Science

December 14, 2024

GitHub Repository

https://github.com/mdlu02/InternetUseClassification

1 Introduction

In today's increasingly digital world, the rise of problematic internet usage among adolescents is becoming a significant concern, closely linked to issues such as depression and anxiety. Understanding and addressing this issue is crucial to promote healthier lifestyles and mental well-being in young people. However, current methods for assessing problematic internet usage often require complex professional evaluations, which can pose access, cultural, and linguistic barriers for many families. As a result, direct measurement of internet use issues is often bypassed, with focus instead placed on associated mental health problems.

Interestingly, physical and fitness measures are much more accessible and can be obtained with minimal intervention or clinical expertise. Changes in physical habits, such as poor posture, an irregular diet, and reduced physical activity, are commonly observed in excessive technology users. This Kaggle competition proposes using these easily obtainable physical fitness indicators as proxies for identifying problematic internet use, especially in settings where clinical expertise or suitable assessment tools are lacking ([Ins]).

The challenge is to develop a predictive model that can analyze children's physical activity data to detect early signs of problematic internet and technology use. By doing so, we can enable timely interventions that promote healthier digital habits and contribute to a future where children are better equipped to navigate the digital landscape responsibly.

The dataset comes from the Healthy Brain Network (HBN), which research screenings of about five thousand individuals aged 5 to 22. The competitions aims to use biological markers to predict the presence of mental health disorders. The dataset incldues physical activity data (from wrist-worn accelerometers, fitness assessments, and questionnaires) and internet usage behavior data. The goal is to predict a participant's Severity Impairment Index (`sii`), a standard measure of problematic internet use, using the data. `sii` categorizations are based on buckets of Parent-Child Internet Addiction Test (PCIAT) scores which range from 0 to 100. As such, this problem can be approached from both a classification and regression problem. Current top submissions achieve a quadratic weighted kappa of around 0.5 with gradient boosting methods such as XGBoost ([Ins]).

2 Exploratory Data Analysis (EDA)

An initial exploratory data analysis was conducted to assess the relationships between features and the target variables `sii` and `PCIAT-PCIAT_Total`.

2.1 Missing Data

There were significant proportions of missing data within the dataset. Notably, 74.82% of the data didn't have associated time series data, and 30.9% of the samples lacked target variable data (figs. 1 and 2).

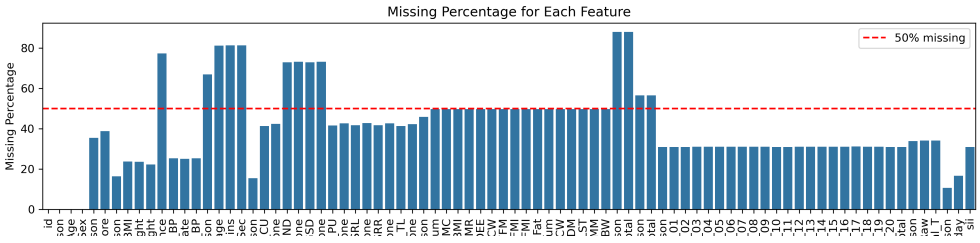

Figure 1: Missing data percentages for all tabular features in the original dataset

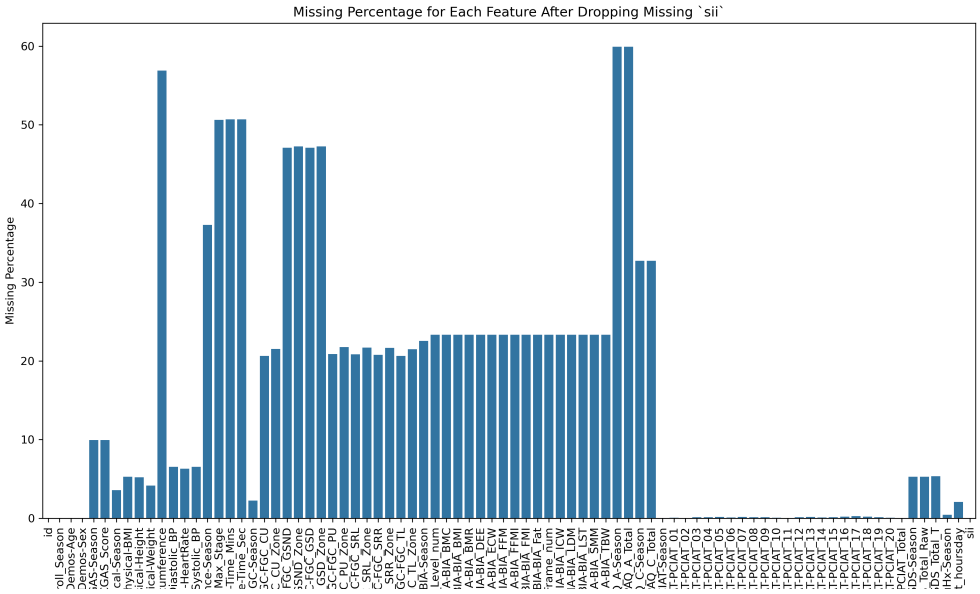

Figure 2: Missing data percentages for all tabular features after dropping samples with missing target variables

2.2 Target Variable Distributions

The target variable is heavily skewed as shown in fig. 3.

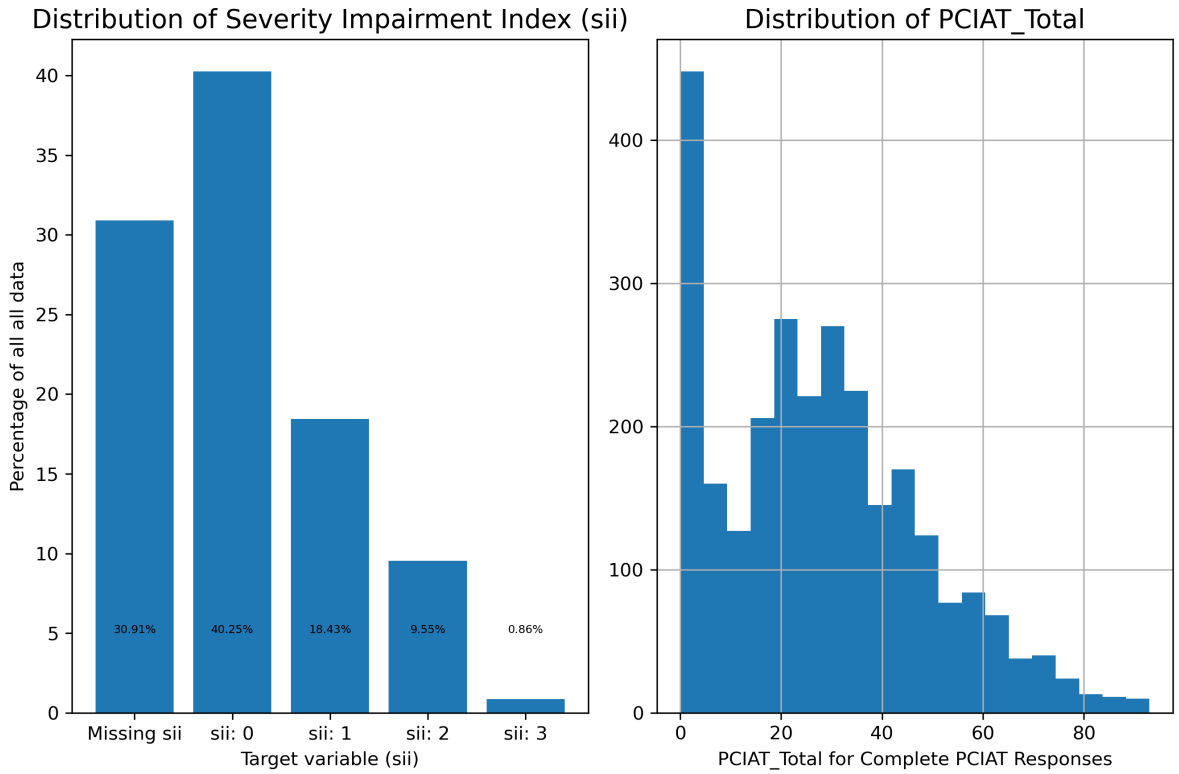

Figure 3: Target variable distributions

2.3 Feature Correlations

Continuous tabular features didn't have strong correlations with the target variable. The feature with the strongest correlation coefficient with respect to `PCIAT-PCIAT_Total` was `Physical-Height` (fig. 4).

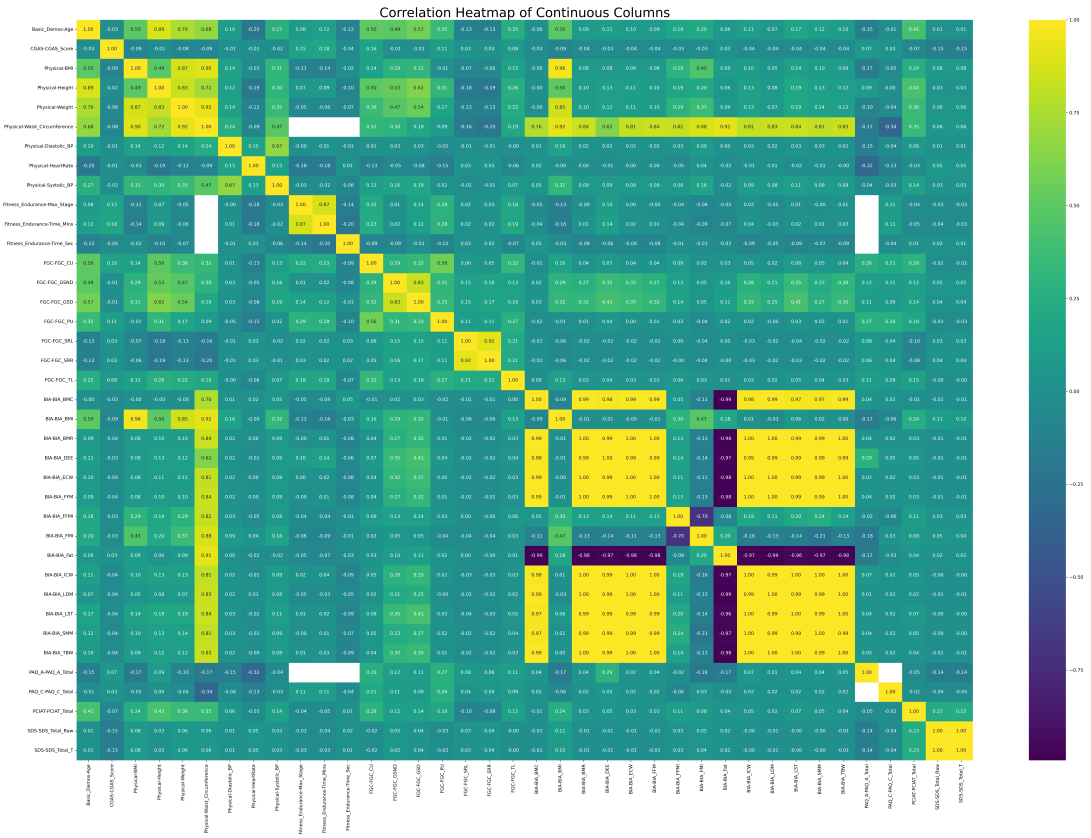

Figure 4: Continuous tabular feature correlation matrix

Welch ANOVA was applied to evaluate associations between categorical features and `PCIAT-PCIAT_Total`. Several features (`PreInt_EduHx-computerinternet_hoursday`, `BIA-BIA_Frame_n` etc.) had significant p-values (fig. 5).

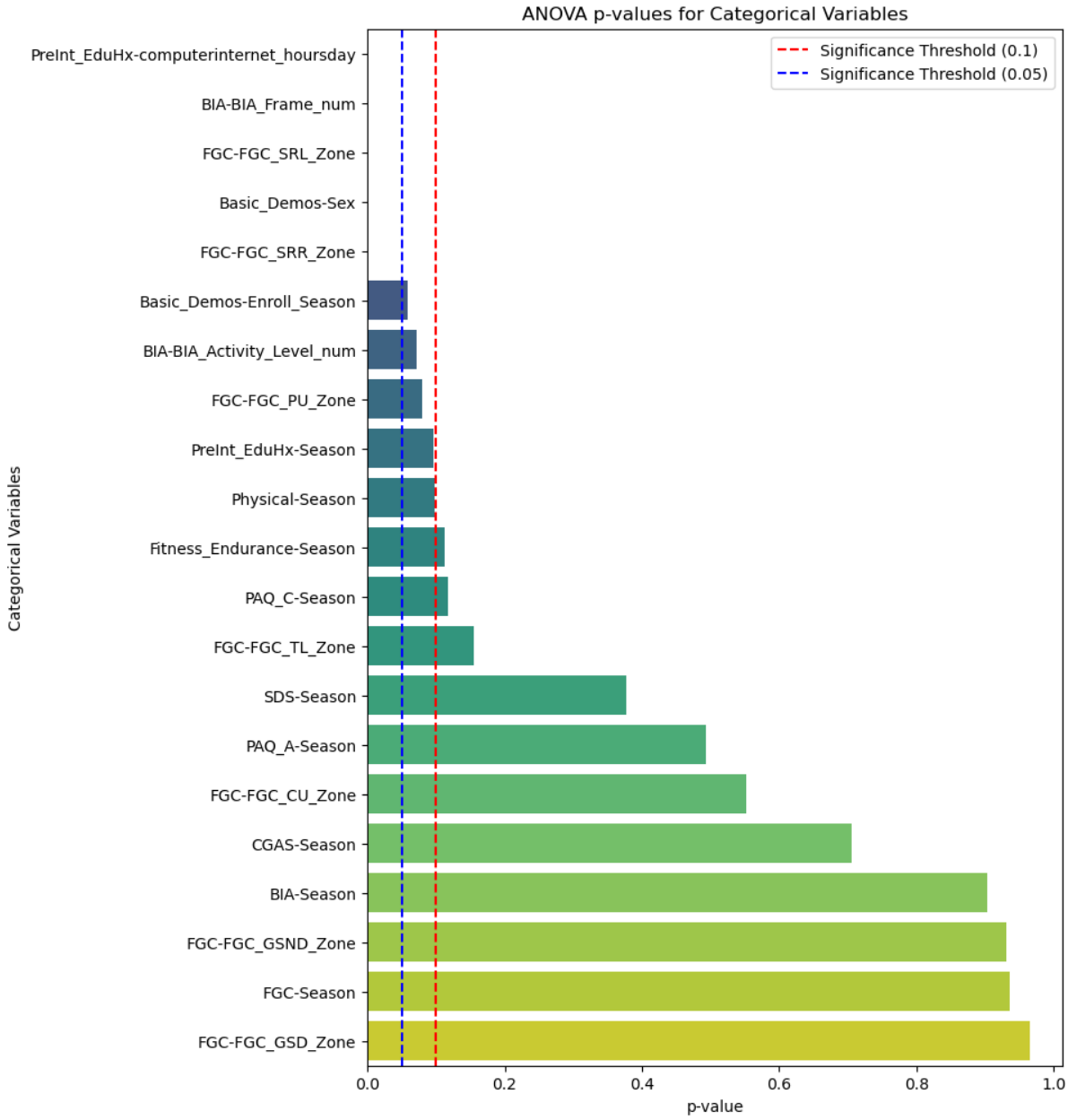

Figure 5: Categorical tabular feature Welch ANOVA p-values with respect to `PCIAT-PCIAT_Total`

2.4 Time Series Analysis

Key time series features were plotted using histograms. `light` was found to contain large outliers (fig. 6).

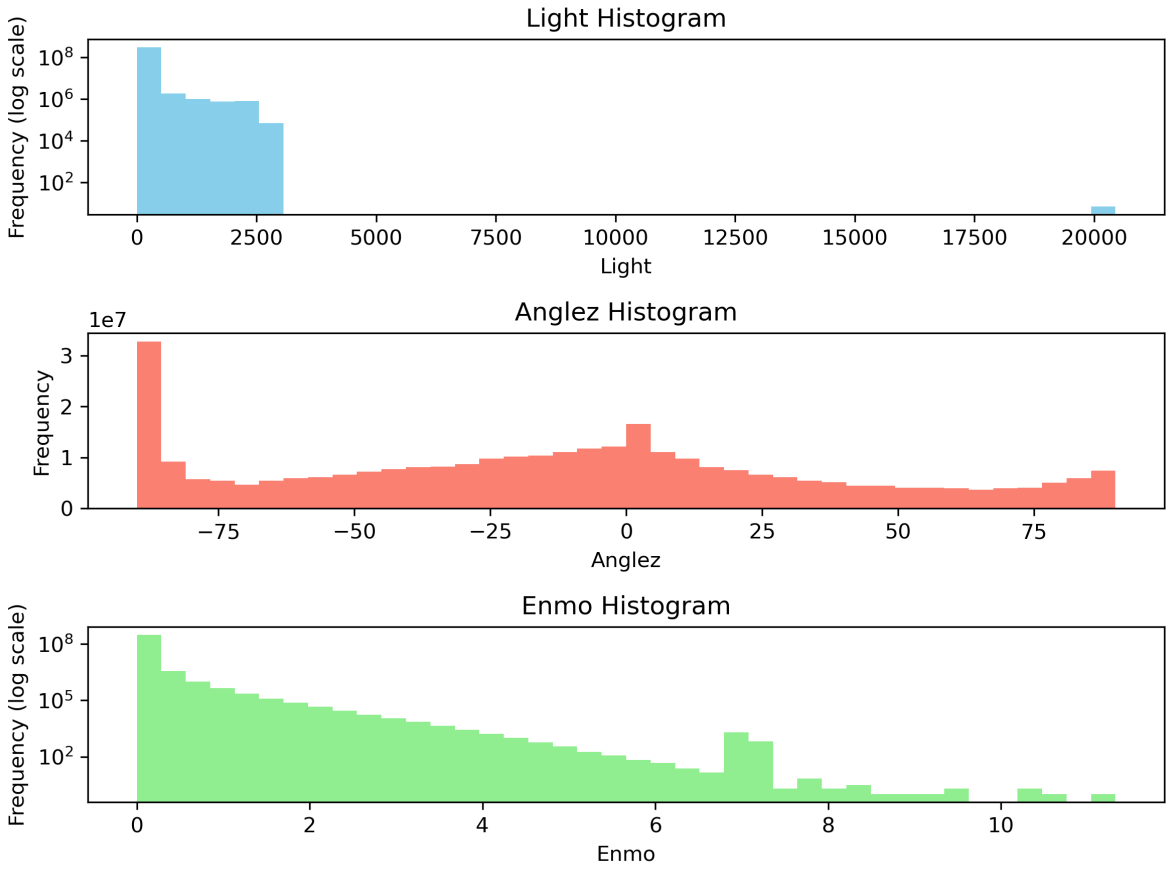

Figure 6: Histograms for time series features

3 Methods

3.1 Splitting

Before splitting, samples without target variables were dropped. Features were split into a test set (20%) and a set used for k-fold cross validation (80%). The split was stratified by `sii`, age group, whether or not a sample had time series data, and `sex`. The k-fold cross validation test set was split further for a 3-fold stratified cross validation (fig. 7). 3 folds was selected since the dataset isn't very large.

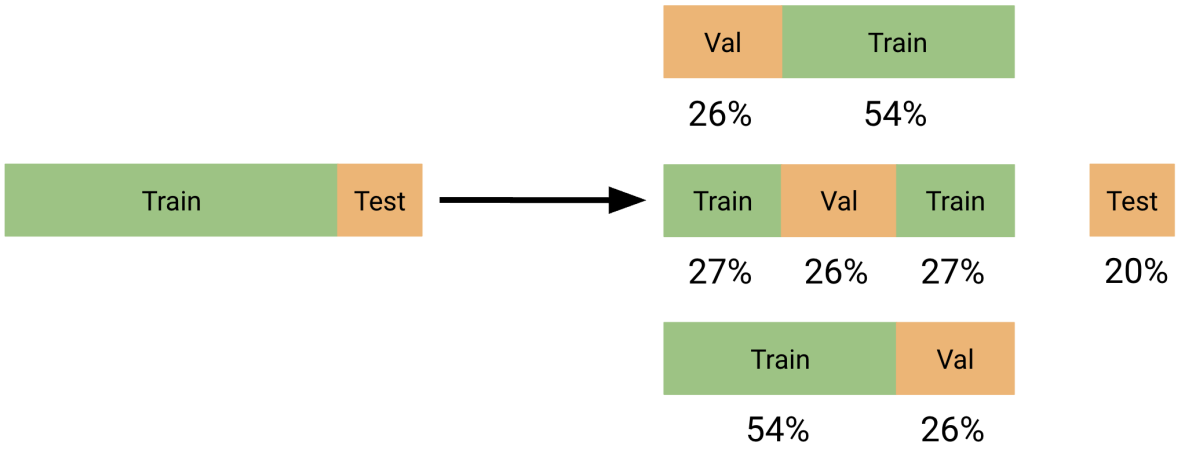

Figure 7: 3-Fold Cross Validation Split Diagram

3.2 Feature Engineering and Transformation

The time series features `light`, `enmo`, and `anglez` were transformed into summary statistics and joined to the feature table on sample ID. Categorical variables were encoded using `OneHotEncoder`, and ordinal variables were transformed using `OrdinalEncoder`. Continuous variables were scaled with `MinMaxScaler` or `StandardScaler` based on the presence of upper and lower bounds. Missing categorical and original features were grouped into a separate class. Continuous features with clear upper and lower bounds were imputed with the data set mean while those without bounds were imputed with the dataset median.

3.3 Feature Selection

Features were removed to ensure no pairwise absolute correlation coefficients greater than 0.95.

3.4 Grid Search and Cross Validation

5 machine learning regression models were selected to tune (fig. 8).

Linear Regression	

Elastic Net	
alpha	l1_ratio
0.1	0.1
0.5	0.5
1	0.9

KNN Regressor	
n_neighbors	weights
3	uniform
5	distance
10	

XGBoost Regressor			
n_estimators	max_depth	learning_rate	subsample
50	3	0.1	0.8
100	5	0.01	1
	7		

Neural Network					
layers	layer_sizes	learning_rate	epochs	batch_size	activation
1	4	0.001	5	8	relu
2	8	0.01	10	16	
4				32	

Figure 8: 3-Fold Cross Validation Split Diagram

Each model/parameter set was trained and evaluated using the 3 folds. The parameter set that achieved the highest average validation error in the cross validation was selected as the model's best parameters. Final test metrics were calculated on the hold out set after training each model with its best parameters on the 80% dataset used for k-fold cross validation. Recoded metrics include QWK and root mean squared error (RMSE). QWK is a measure of agreement between two sets of labels. It is weighted by the distribution of the data such that the baseline score for any dataset is 0. To evaluate QWK, the raw regression prediction for each sample was sorted into the following buckets to get the appropriate `sii` value:

1. `sii` = 0: 0-29
2. `sii` = 1: 30-49
3. `sii` = 2: 50-79
4. `sii` = 3: 80+

Baseline RMSE on the test set is 20.335.

4 Results

4.1 Overall Model Performance

As we can see in the box plots, the KNN Regressor has significantly worse performance than the other 4 models for both QWK and RMSE (figs. 9 and 10). All the models performed better than the baseline QWK and RMSE scores. Linear regression, elastic net, XGBoost, and the neural network all had very similar performance with average QWKs around 0.39 with the neural network having a median QWK just under 0.4 (fig. 9). Linear regression and the neural network have large RMSE outliers which can be explained by the lack of regularization in both models. XGBoost Regressor had the lowest median RMSE (fig. 10).

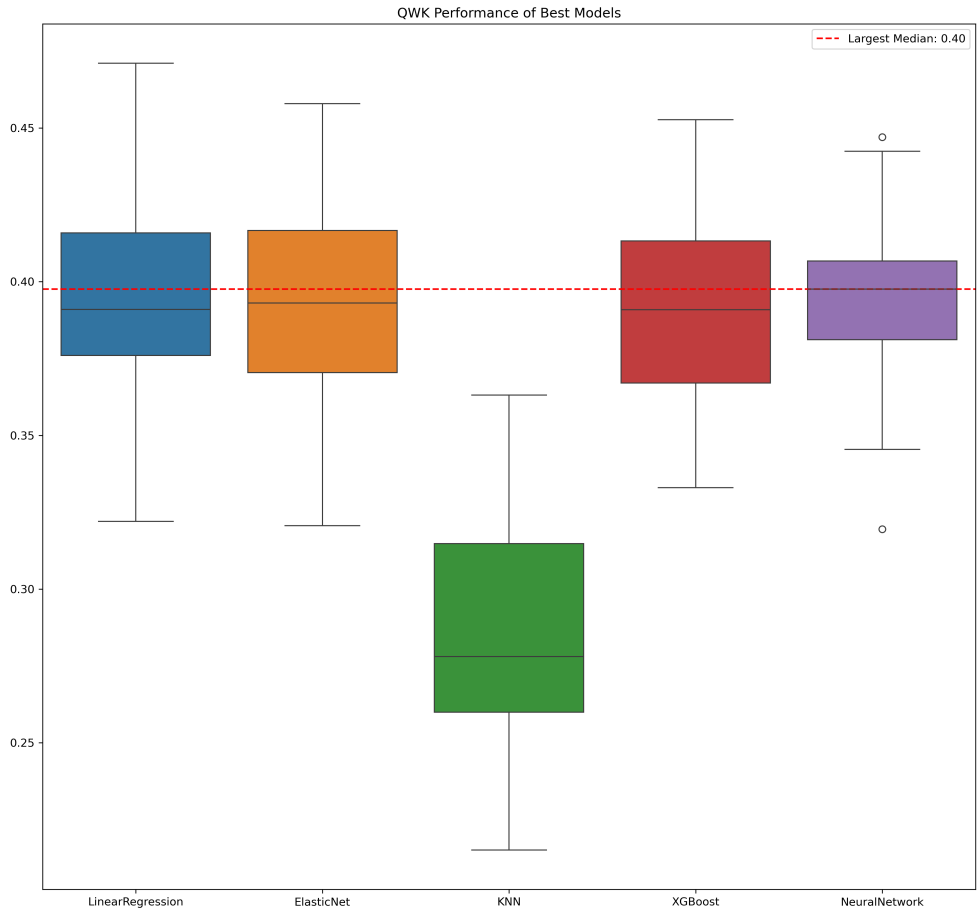

Figure 9: Best Model Results: Quadratic Weighted Kappa

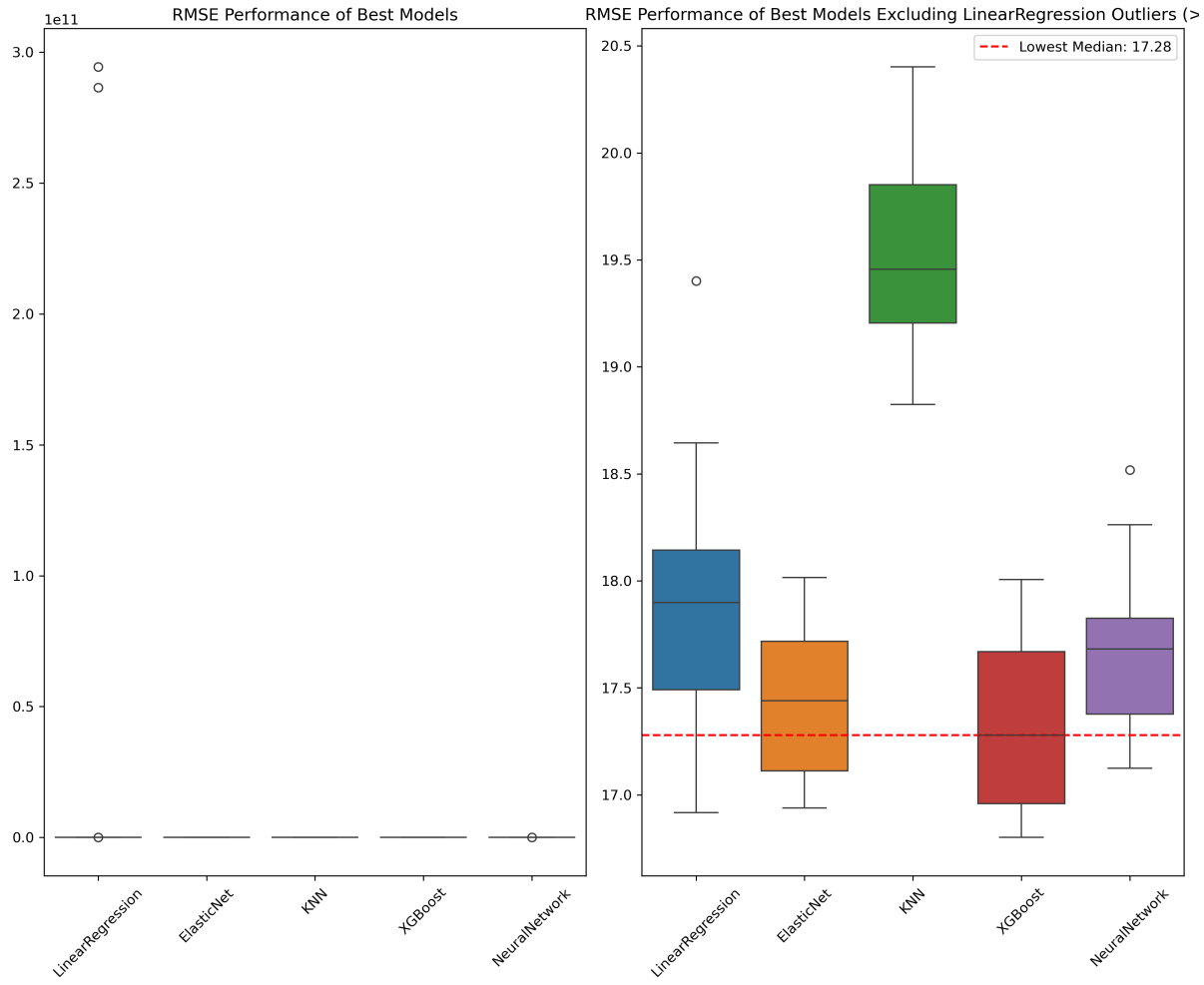

Figure 10: Best Model Results: Root Mean Square Error

4.2 Confusion Matrices

Looking at the confusion matrices reveals that all the models perform poorly on the minority classes and often incorrectly predict lower `PCIAT-PCIAT_Total` scores compared to the true labels (figs. 11 to 15).

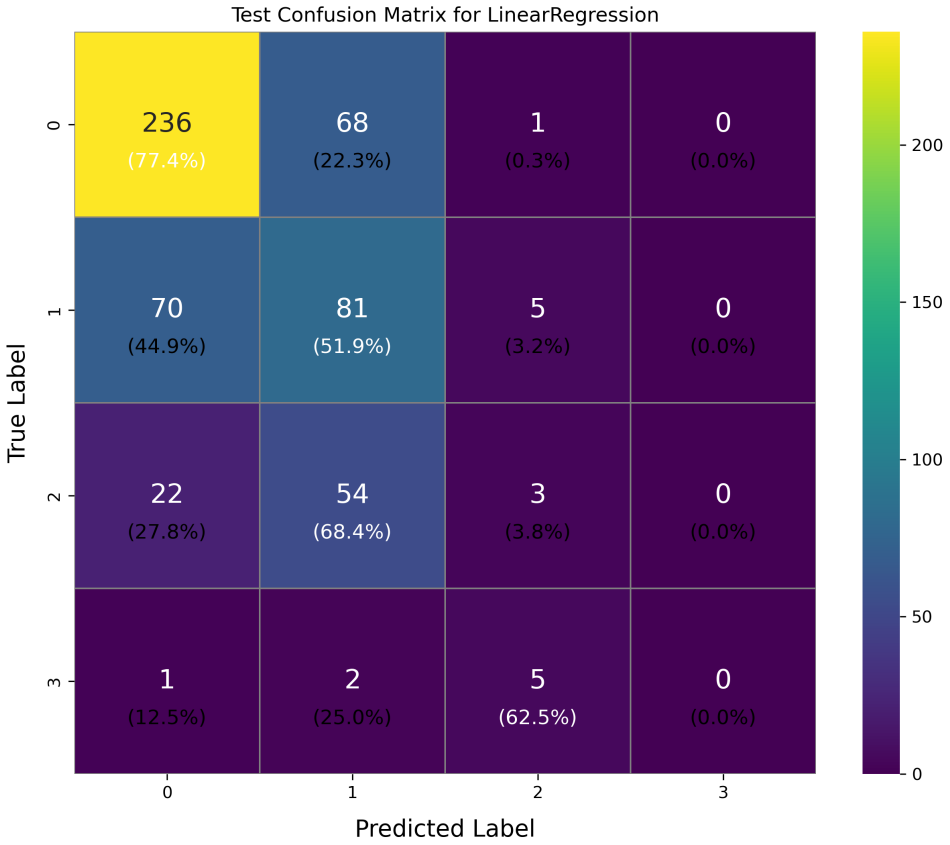

Figure 11: Linear Regression Test Prediction Confusion Matrix

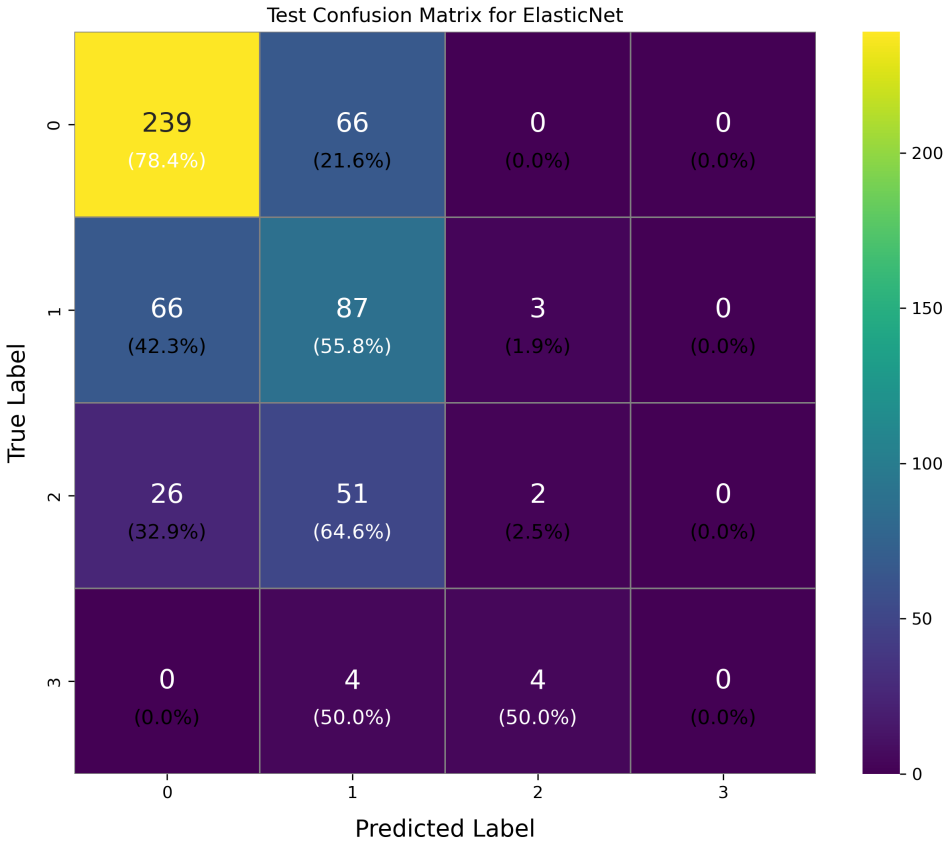

Figure 12: Elastic Net Test Prediction Confusion Matrix

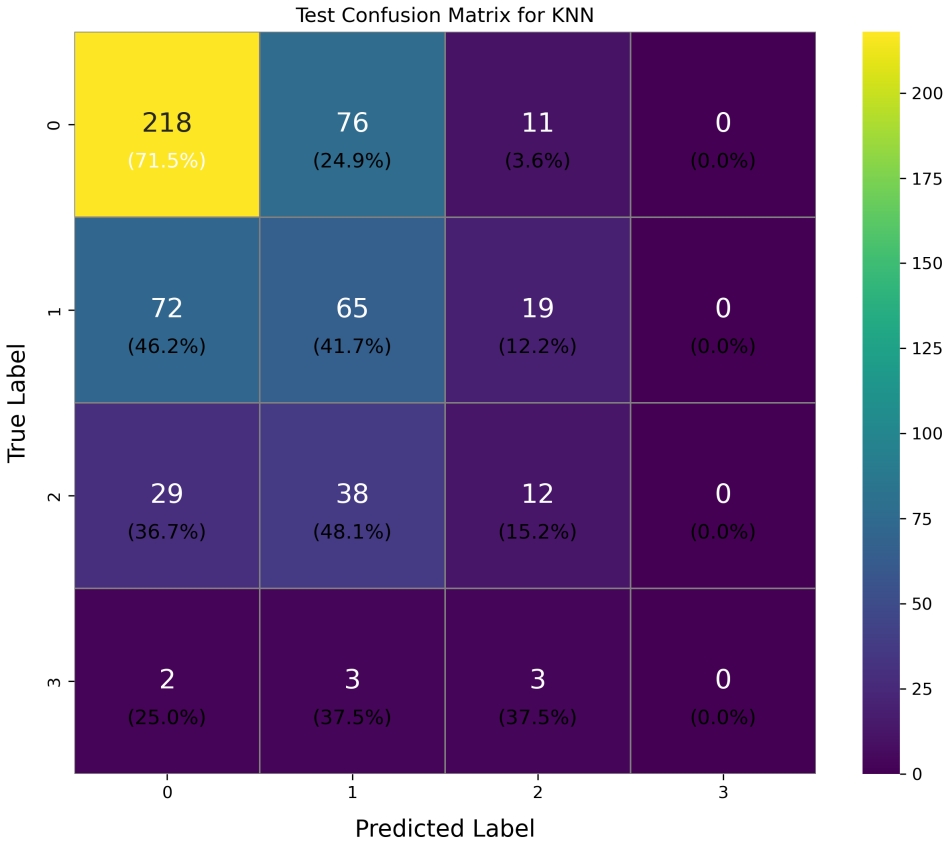

Figure 13: KNN Regressor Test Prediction Confusion Matrix

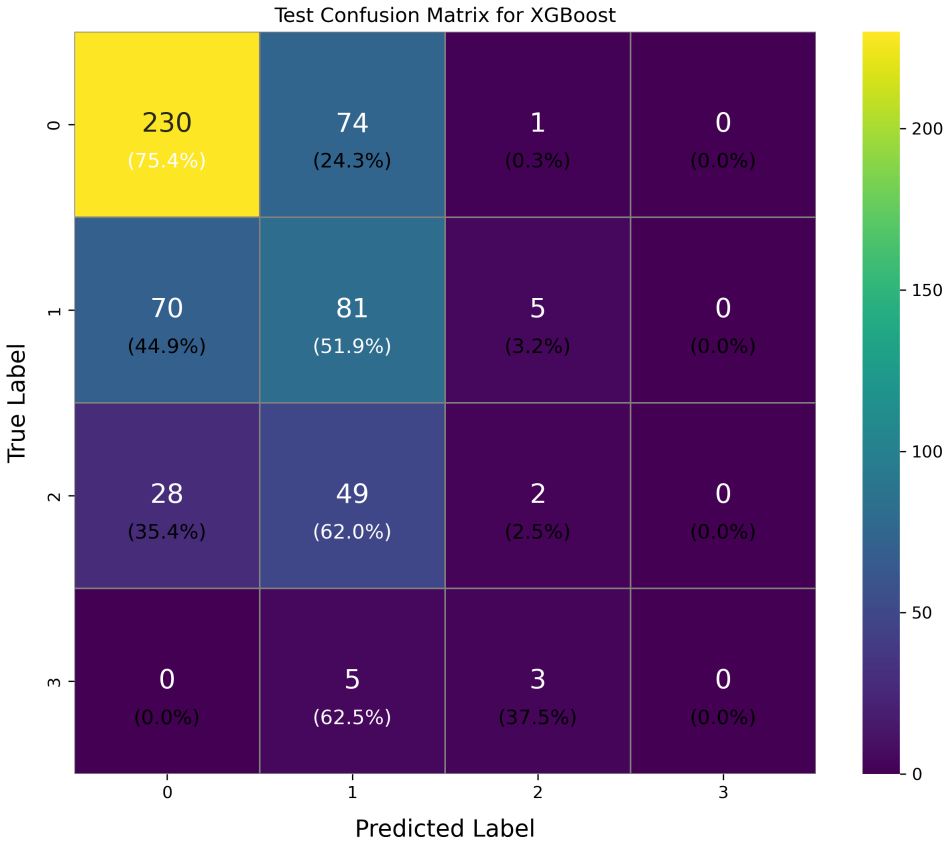

Figure 14: XGBoost Regressor Test Prediction Confusion Matrix

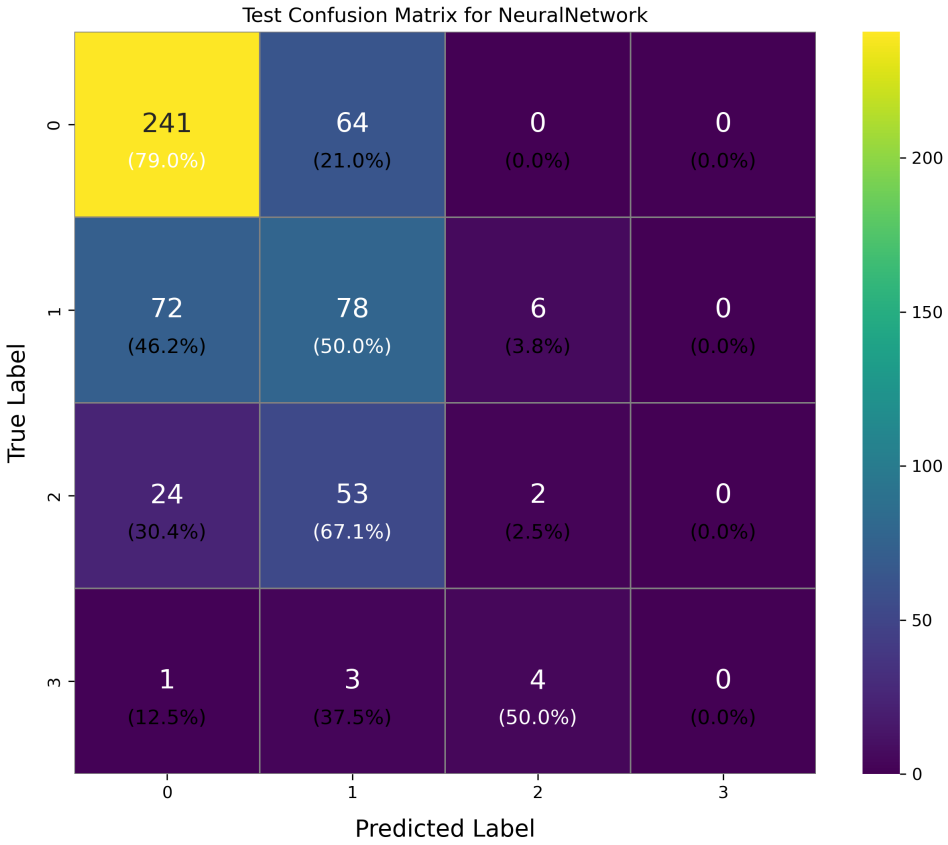

Figure 15: Neural Network Test Prediction Confusion Matrix

This behavior can be attributed to the significant imbalance in the dataset. Due to the regression approach, there is limited ability to weight the training error of the model to prioritize performance on minority classes after bucketing.

The neural network has the best performance for class 0 points but is beat by simpler models such as the linear regression based models and XGBoost for class 1. Elastic net outperforms all other models for class 2 accuracy. None of the models were able to correctly predict any class 3 points. Interestingly, the simple linear regression model had the best accuracy for class 3 points when considering the distance of incorrect predictions with $\frac{5}{8}$ of the class 3 points classified as class 2. Assessing the relative distances from the desired class provides a different way to assess model performance.

One-vs-all ROC-AUC curves for each model were generated. For sake of brevity only the neural network AUC-ROC curve was included (fig. 16).

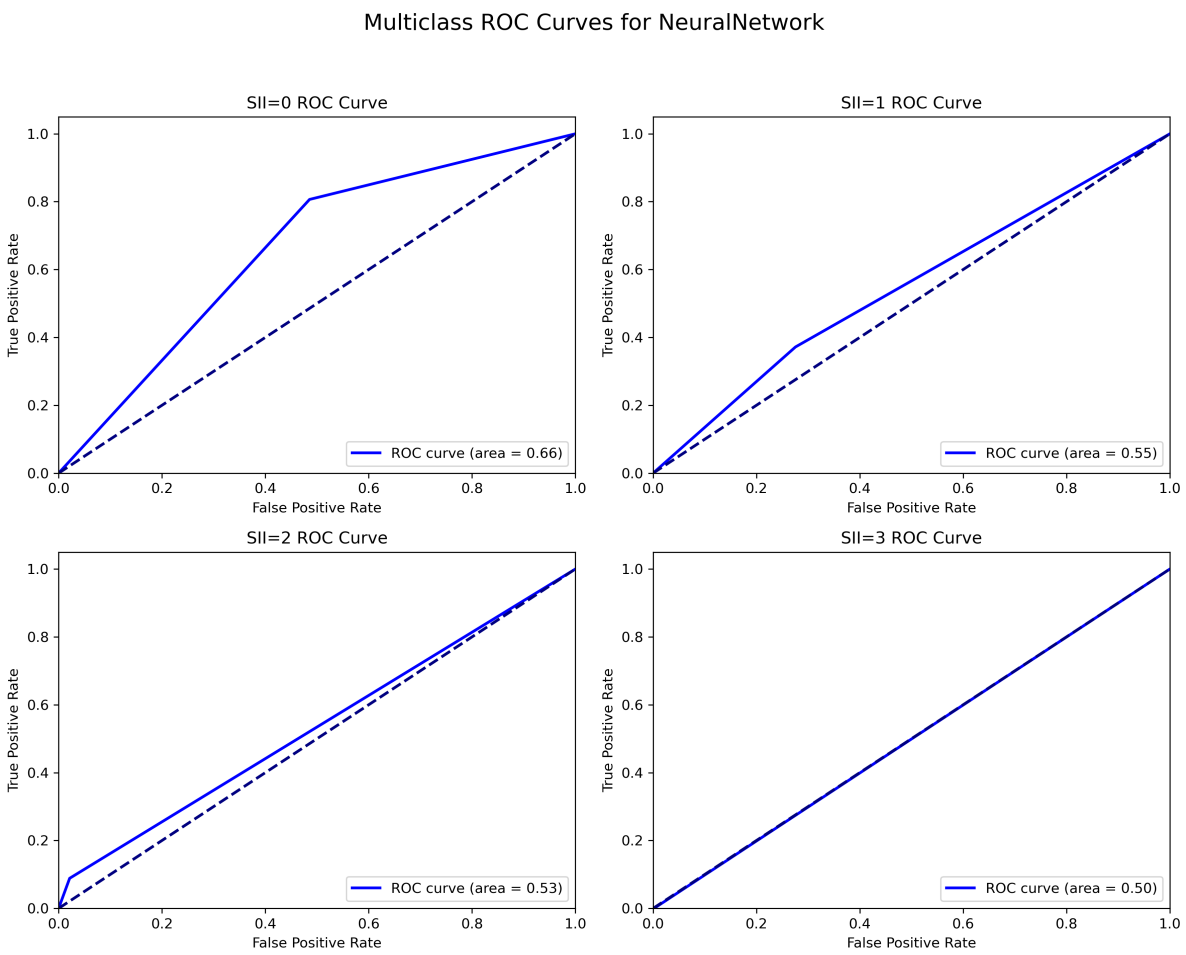

Figure 16: Neural Network Test Prediction Confusion Matrix

Class 0 ROC-AUC values ranged from 0.65 to 0.7 indicating decent performance across all models for the majority class. ROC-AUC drops off to around 0.56 to 0.62 for class 1 predictions before dropping to 0.5 for class 2 and 3. Similar behavior can also be seen in the precision recall curves (again, only including for the neural network for sake of brevity) (fig. 17).

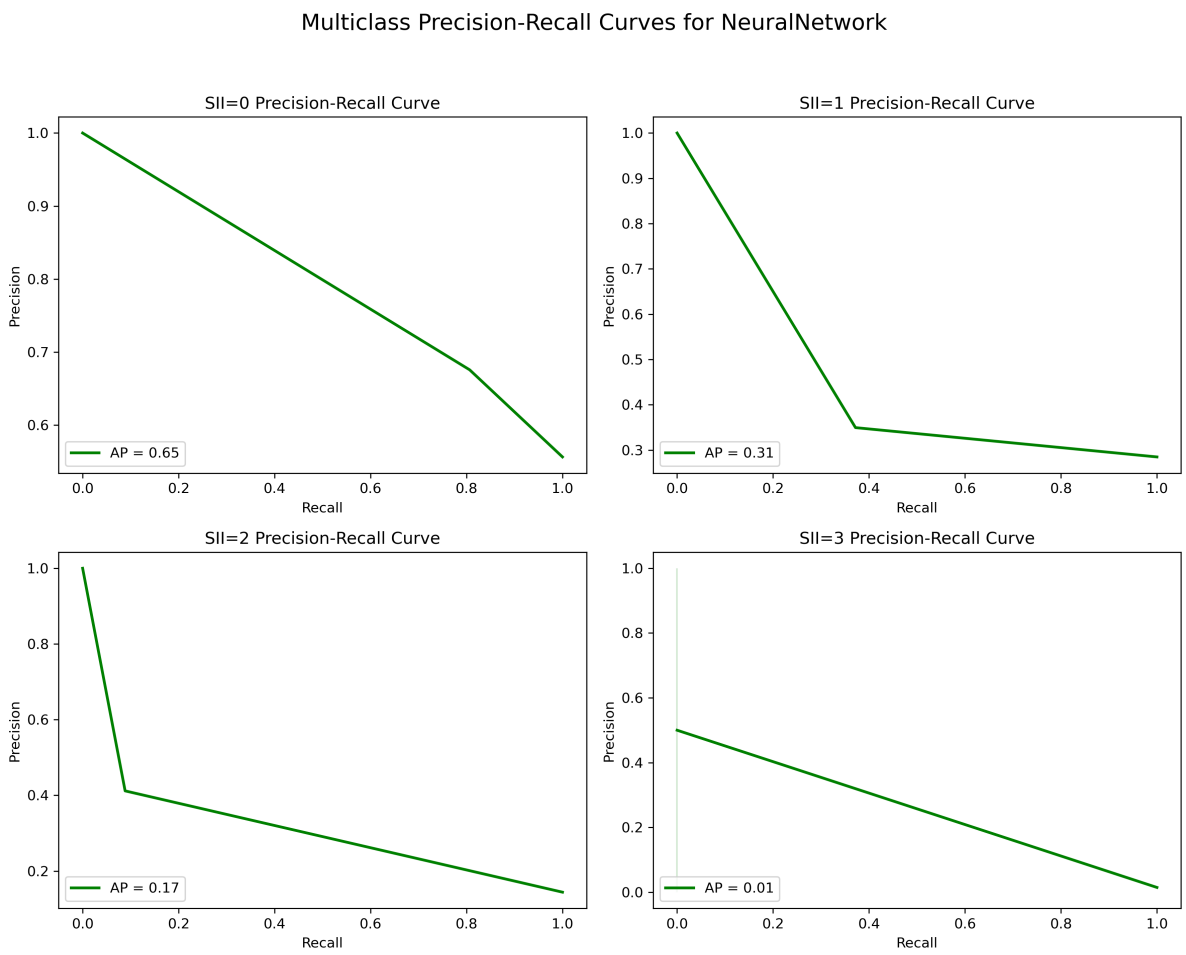

Figure 17: Neural Network Test Prediction Confusion Matrix

4.3 Permutation Importance

Feature permutation importance scores were calculated and top ten features were plotted for each model (figs. 18 to 22). The features `weekday_enmo_90`, `BIA-Season_missing`, and `Physical-Season_Summer` frequently appear near or at the top of the ranked permutation importance across the models. `enmo_90` is the 90th percentile of the Euclidean Norm Minus One time series feature which tracks acceleration and motion. The importance of this feature makes intuitive sense as the more active an individual is, the less time they spend the internet. `Physical-Season_Summer`'s importance also makes sense as adolescent individuals might be more active during break. BIA examinations measure key body composition elements, such as BMI, fat, muscle, and water content.

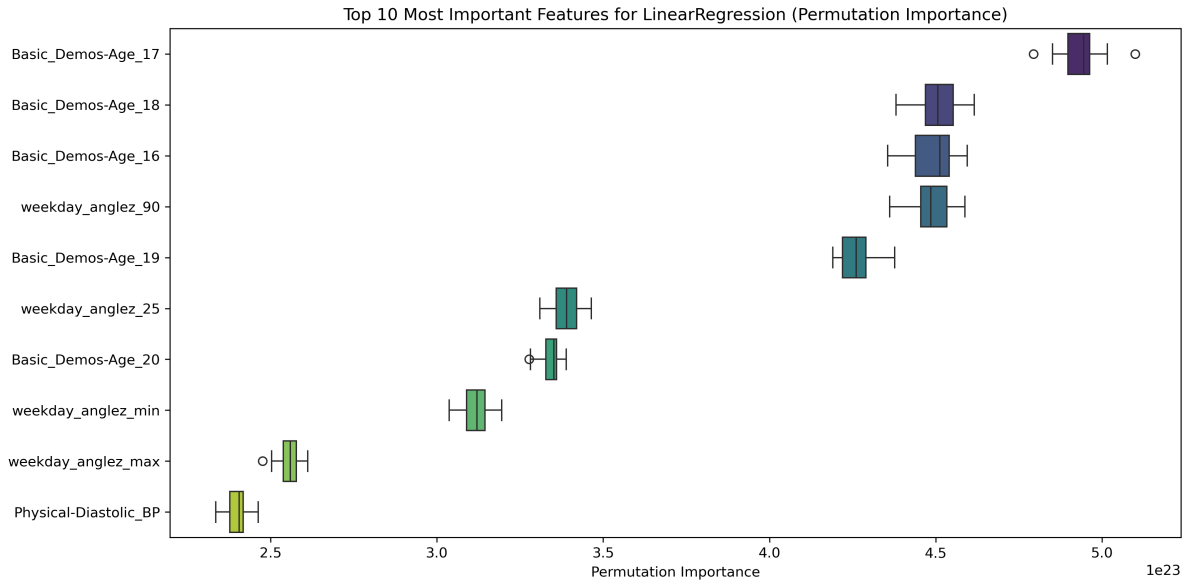

Figure 18: Linear Regression Permutation Importance

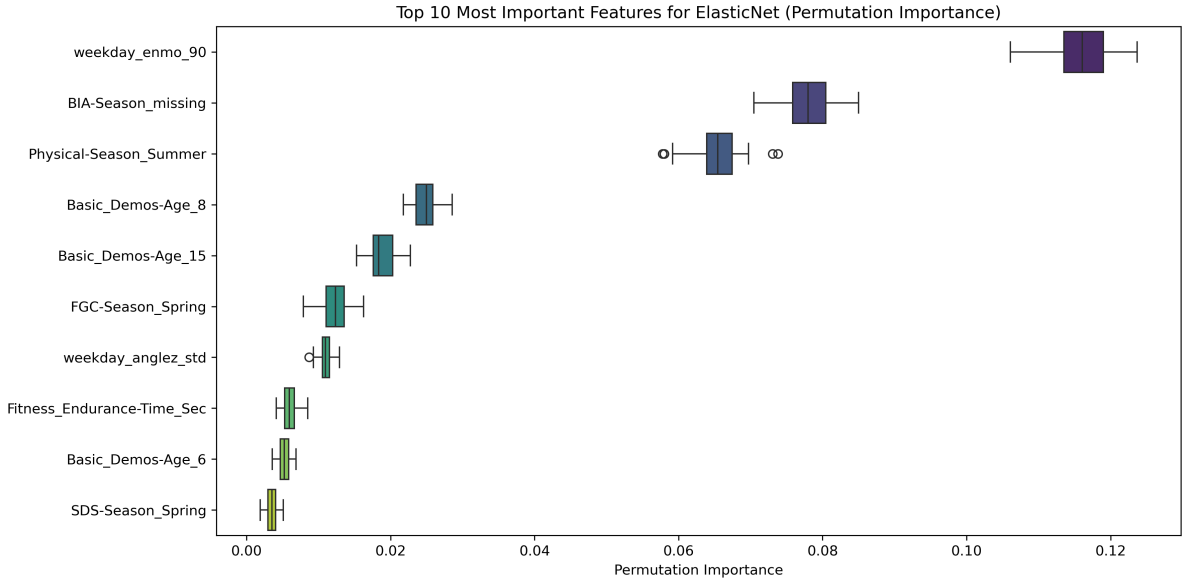

Figure 19: Elastic Net Permutation Importance

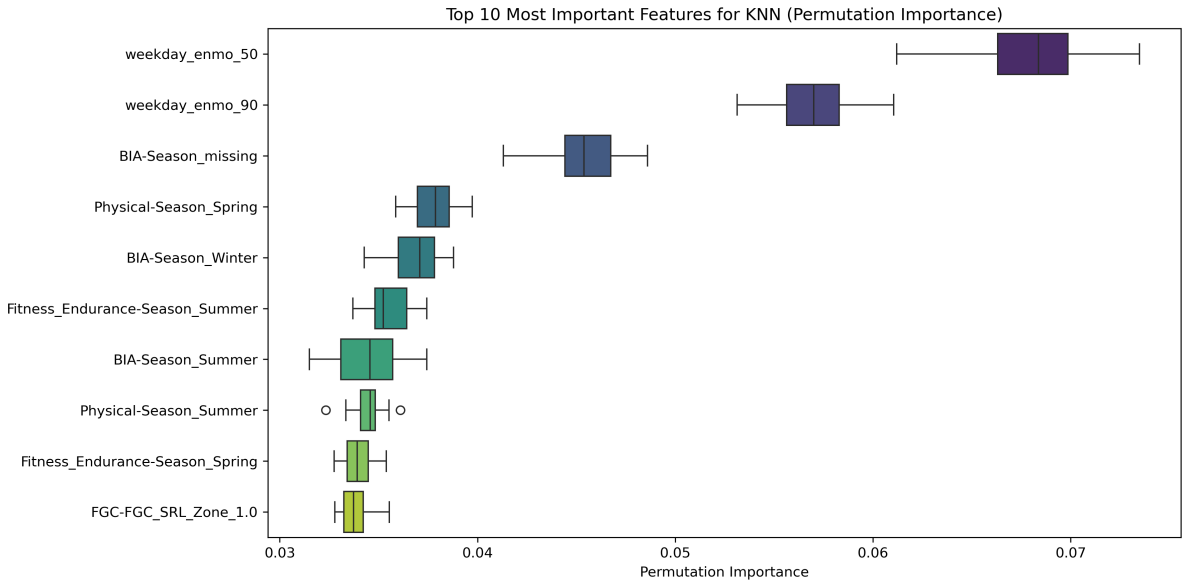

Figure 20: KNN Regressor Permutation Importance

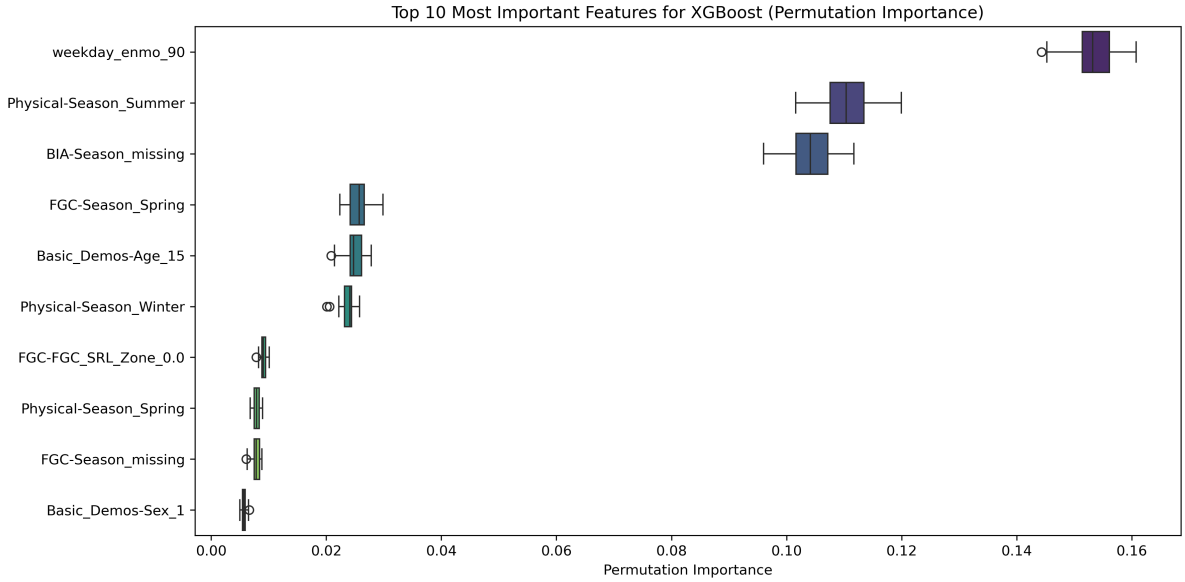

Figure 21: XGBoost Regressor Permutation Importance

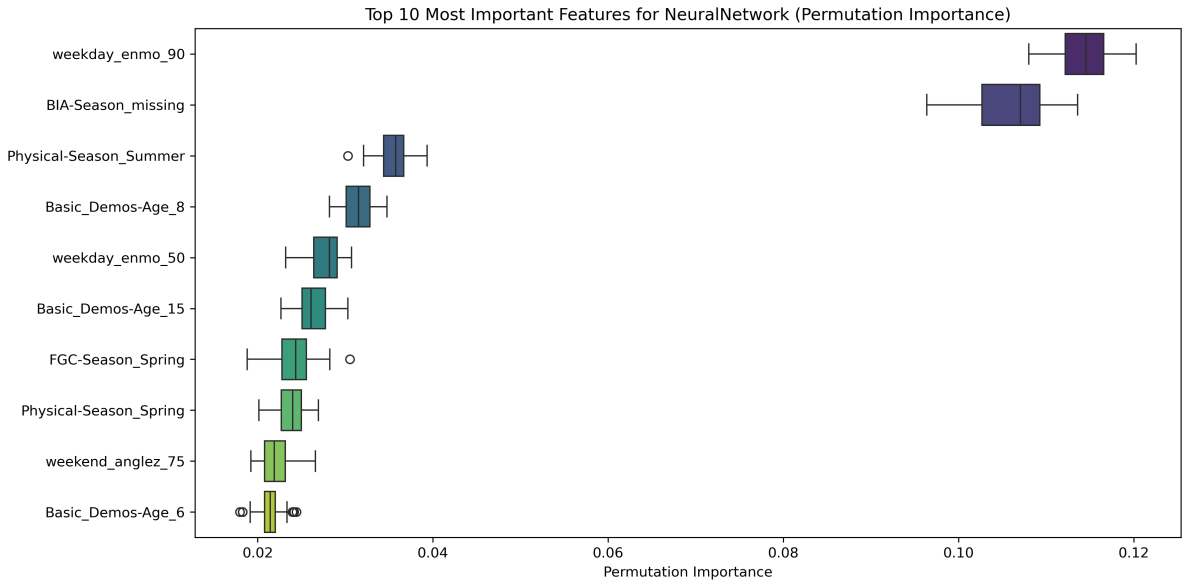

Figure 22: Neural Network Permutation Importance

4.4 Local Importance for Neural Network

SHAP was used to investigate local feature importance for several test samples (figs. 23 to 25).

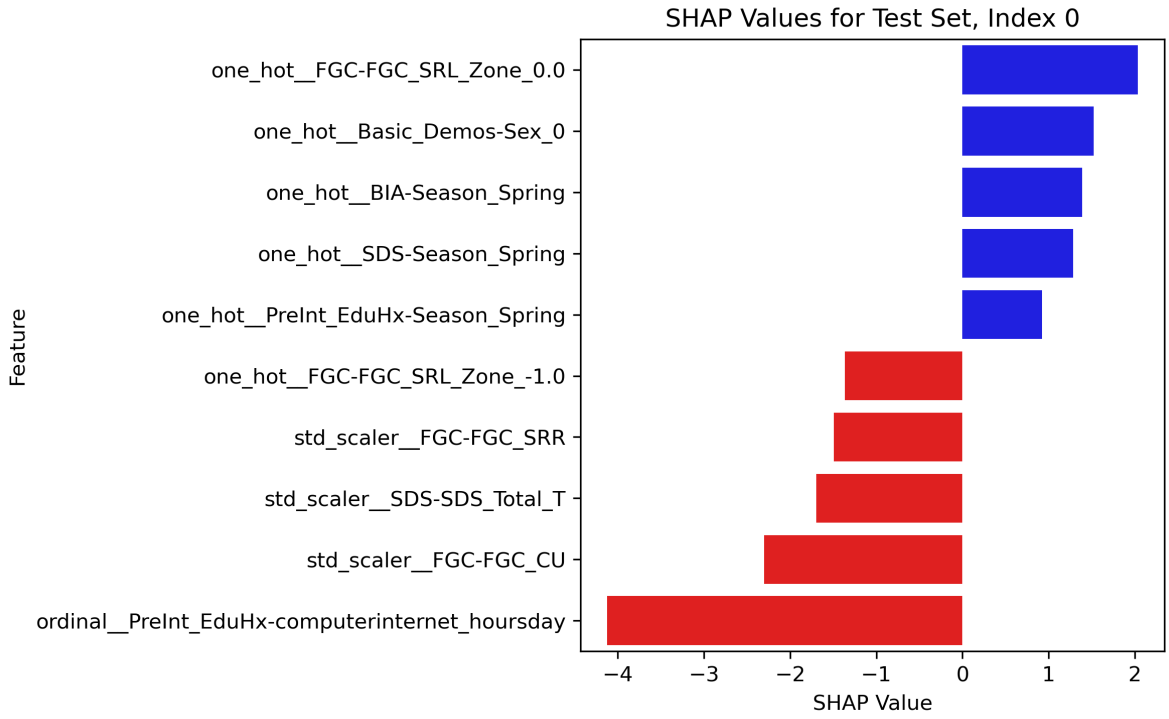

Figure 23: Top 5 Negative and Positive Local SHAP Features for Test Sample 0

Figure 23 shows that the top 5 positively contributing features for test sample 0 are `FGC-FGC_SRL_Zone_0.0`, `Basic_Demos-Sex_0`, `BIA-Season_Spring`, `SDS-Season_Spring`,

and PreInt_EduHx-Season_Spring. FGC_SRL_Zone is from a sit and reach fitness test which indicates an individuals range of motion to their left while seated. The remaining top positive contributors are demographic or assessment season data. On the negative side, PreInt_EduHx-computerinternet_hoursday had the largest contribution. This is an ordinal variable indicating the range of average hours the individual spends on the internet per day. Other notable negatively contributing features are seated range of motion to the right, curl up score, and sleep disturbance score.

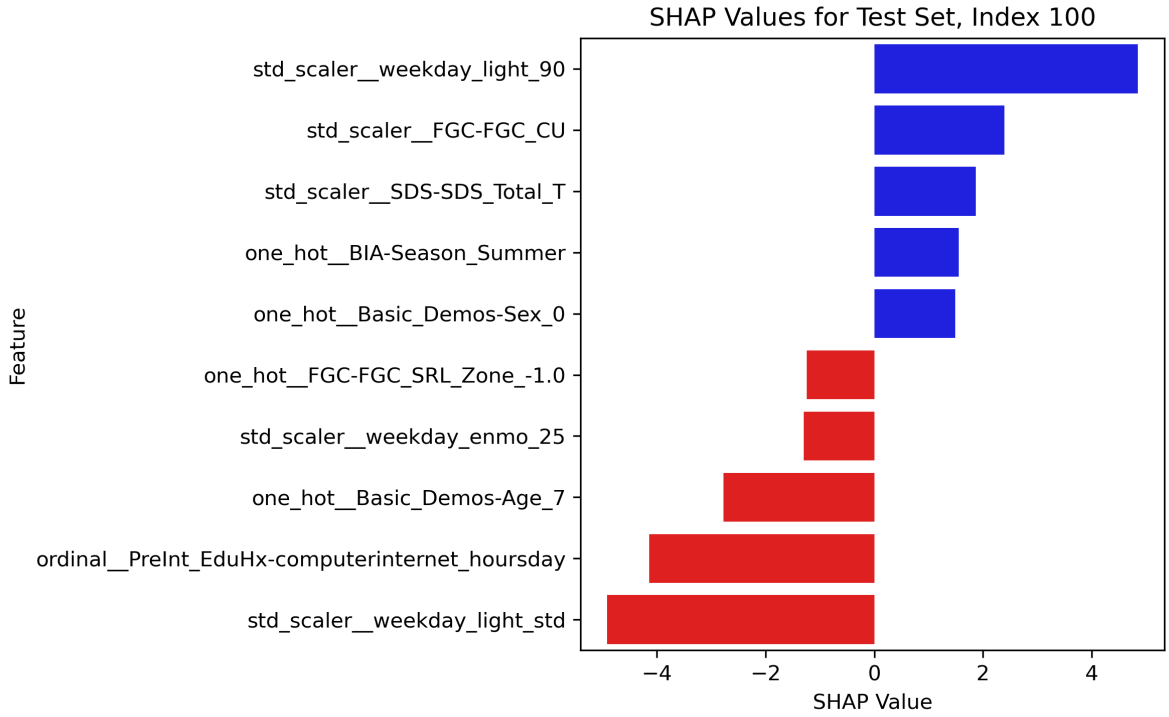

Figure 24: Top 5 Negative and Positive Local SHAP Features for Test Sample 100

Figure 24 shows that the top 5 positively contributing features for test sample 100 are weekday_light_90, FGC-FGC_CU, SDS-SDS_Total_T, BIA-Season_Summer, and Basic_Demos-Sex_0. weekday_light_90 the 90th percentile of light levels (lumens) measured by the wrist wearable on weekdays. Other top features include demographic, sleep quality, and physical conditioning data such as the sex, curl up score, and sleep disturbance score. On the negatives side, weekday_light_std had the largest contribution. Other notable negatively contributing features are PreInt_EduHx-computerinternet_hoursday, variablity of weekday enmo, seated range of motion, and demographics.

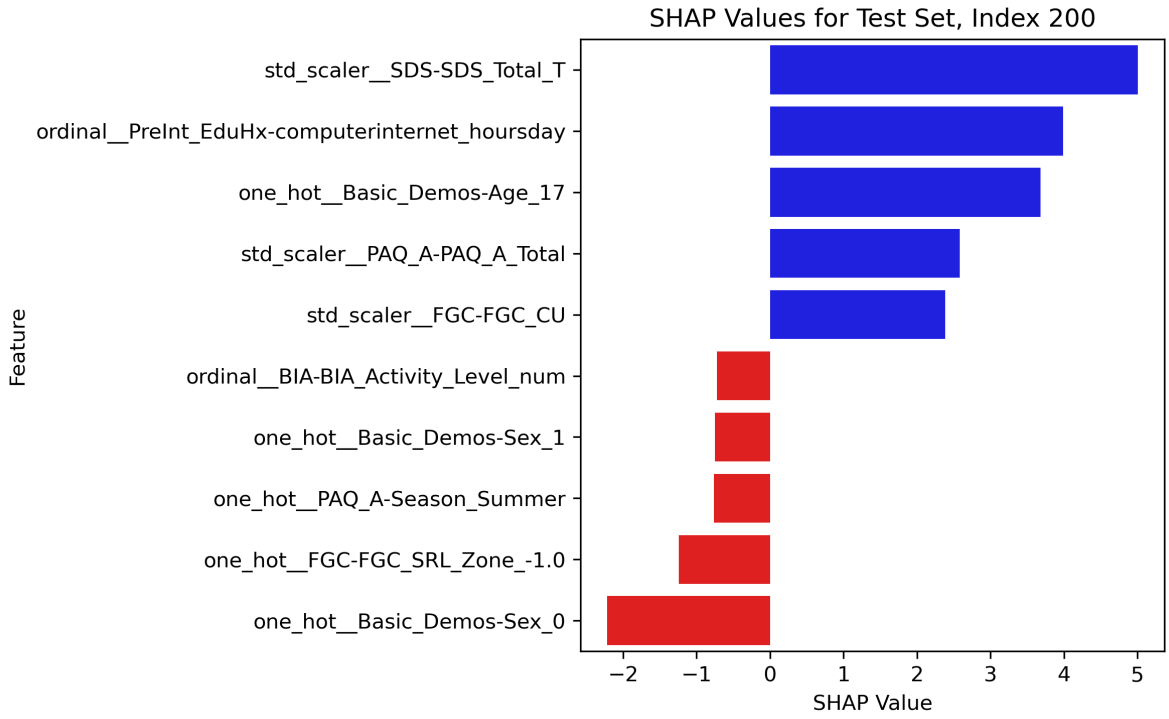

Figure 25: Top 5 Negative and Positive Local SHAP Features for Test Sample 200

Figure 25 had greater positive contributions compared to the top negatively contributing features. The feature with the largest positive contribution is `SDS-SDS_Total_T` indicates total sleep disturbance. Other features top positive features are the individual's daily internet usage time, age group, physical activity questionnaire, and fitness curl score. Activity level, sex, physical questionnaire, and seated range of motion were the top negatively contributing features.

5 Outlook

One area where could makes improvements is in hyperparameter tuning. Models like the neural network and XGBoost have additional parameters such as `colsample_bytree` and `activation_function` which can change the behavior and performance of the model. If I had more time, I would dramatically expand the grid search parameters and add new ones for my models to see if I can achieve better QWK scores.

I would also create a new pipeline for a classification approach. This would allow me to implement class weighting and tune critical p-vals to maximize QWK scores.

I would also take a closer look at how I handle missing values in the dataset. Currently, I drop all samples without `sii` and/or `PCIAT-PCIAT_Total`. I noticed that several features without the target variables do have some values for other `PCIAT-PCIAT` test scores which are used as part of the calculation for the `PCIAT-PCIAT_Total` score. I could investigate if it is reasonable to use these to infer `PCIAT-PCIAT_Total` and `sii`.

Finally, I would try different approaches for handling time series data. For example, I could add a recurrent neural network to my neural network model to improve learning on the raw time series data and I know there are methods for passing time series data directly into XGBoost.

References

[Ins] Child Mind Institute. *Child Mind Institute - Problematic Internet Use*. URL: https://www.kaggle.com/competitions/child-mind-institute-problematic-internet-use/overview (visited on 12/14/2024).