1. **Dataset Description:**

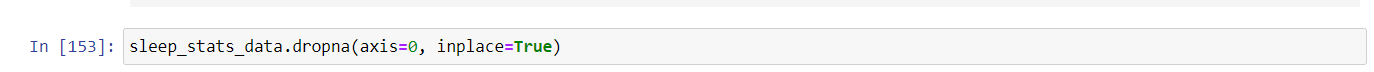
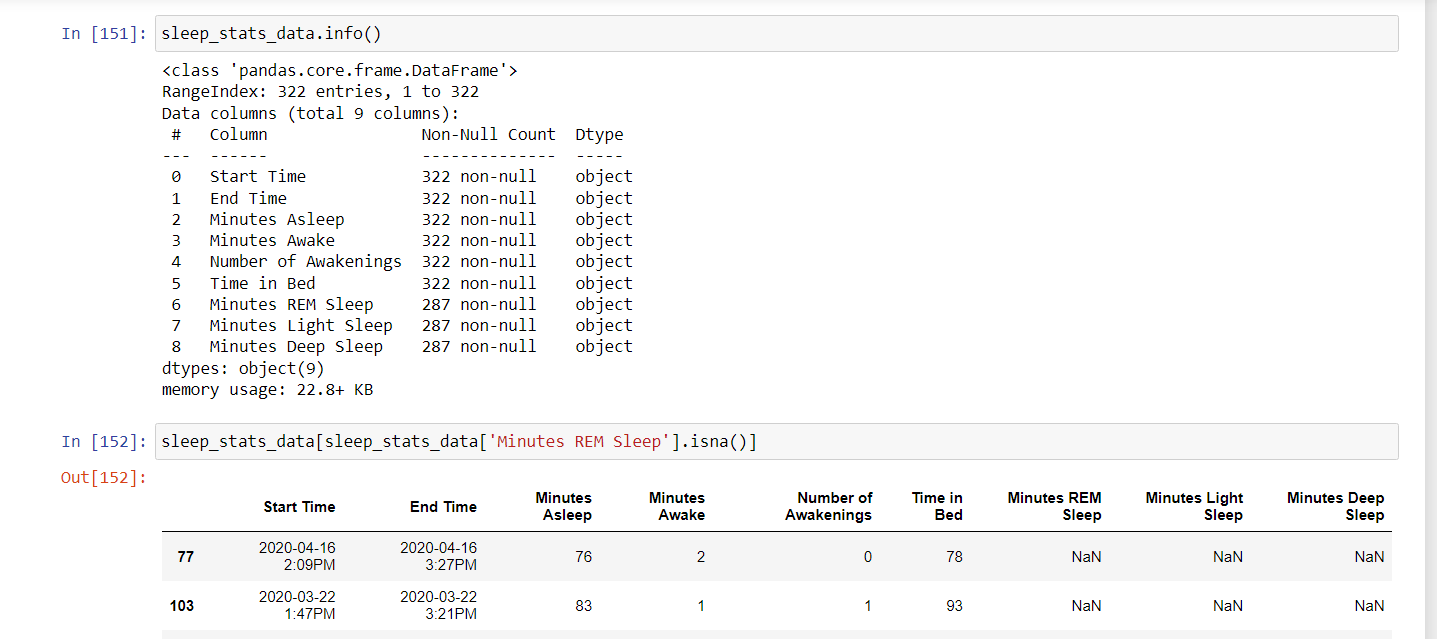
The research paper utilizes datasets gathered from **Fitbit wearable devices**, focusing on sleep-related data. One dataset encompasses a range of sleep metrics recorded at different times, including overall sleep scores, composition scores indicating sleep stage distribution, and various other parameters like deep sleep duration and resting heart rate. The second dataset provides detailed information on sleep start and end times, total sleep duration, frequency of awakenings, and breakdowns of different sleep stages. These datasets collectively offer valuable insights into individuals' sleep patterns, quality, and behaviors, facilitating comprehensive analysis in the study of sleep science.

1. A diagram of a flowchart

   Description automatically generated**Diagram:**
2. **Code:**
3. **Pre-processing the sleep\_stats dataset:**

* **A screenshot of a computer

  Description automatically generated**The first row is set as column names for the dataset, and 1st row is being dropped.
* Upon inspection and by having a close look at start and finishing times for the sleep stats with NaNs for REM, Light and Deep sleep we see that these data points refer to afternoon naps. These should be excluded from our data set as they do not have a sleep score, or the necessary sleep stats attached to them.



* **A screenshot of a computer

  Description automatically generated**Convert the columns from object type to **float datatype** (for the numerical columns).

1. **Pre-processing the sleep\_score dataset:**

* Only the first two columns (“timestamp” and “overall\_score”) of the dataset are imported.
* “timestamp” column is required during merging of the two datasets and “overall\_score” column is the target variable.

1. **Merging the datasets:**

* A column named “Date” is being created in both the (using “timestamp” column in sleep\_score dataset and “End time” in sleep\_stats dataset) and the datasets are merged using this column.

**A screenshot of a computer

Description automatically generated**

1. **Remove columns that are not necessary:**

* **A screenshot of a computer

  Description automatically generated**Columns like “End time”, “timestamp”, “Date”, “Number of awakenings” as these are not required for prediction of target variable.

1. **Visualization:**

* Inspecting the relationship between the dependent variable(“overall\_score”) and all the independent variables using **scatter plot** to get a sense of their impact on it.
* “**Minutes Asleep**”, “**Minutes REM Sleep**” and “**Time in Bed** seem to have the strongest positive relationships with the overall sleep score.

**A screenshot of a computer screen

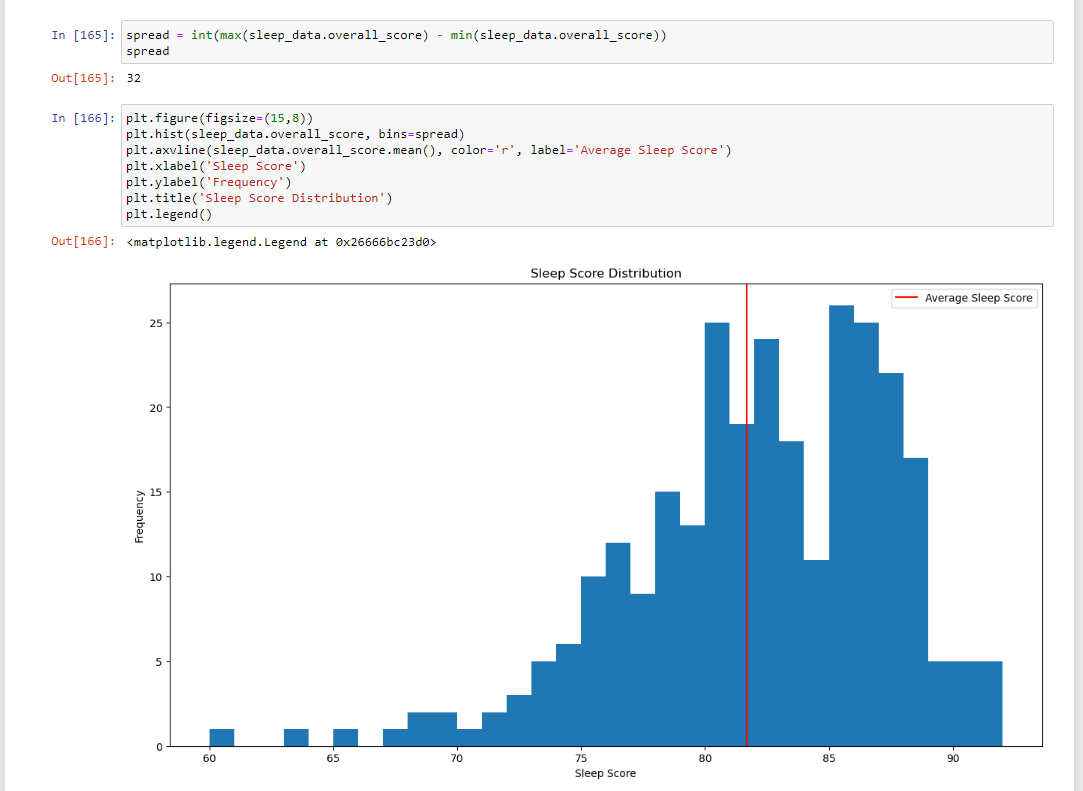
Description automatically generated**

* Inspecting the correlations using **a heat map** – “**Time in Bed**” and “**Minutes Asleep**” had the strongest correlation. This makes sense because more time in bed should lead to more sleep, and therefore more quality sleep.

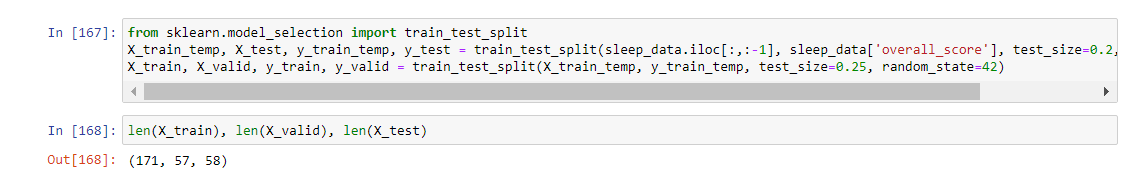
**A screenshot of a computer

Description automatically generated**

* Inspecting the sleep score distribution using a **histogram** and visualizing the mean - The distribution of sleep scores is skewed to the left. This makes sense because bad night sleeps are more likely to occur than exceptionally good ones due to multiple reasons such as staying out late, having to get up extremely early, etc. Given that the average sleep score in the data set is **already relatively high at 82 and the upper limit being 100**, **it is difficult to have many outliers that lie above the average**.

****

1. **Splitting the dataset into training, validation and test set:**

* It's crucial in ML to prevent information from test/validation data influencing model training. Scaling features involves using data statistics like mean and variance, which could introduce bias if applied before splitting. Thus, data should be split into **training, validation, and test sets** beforehand. This ensures no data leakage, maintaining model integrity during training.
* ****The training and test set are being split in the ration 8:2. This training set is further divided in the ratio 1:4 into training and validation set. This is done using **the train\_test\_split function**.

1. **Feature Scaling:**

* Feature scaling is done to scale down all the numerical features to a common interval, here we have used **Min-Max scaler** which fits all the values between 0 and 1.
* Why Min-Max scaler?
* A screenshot of a computer

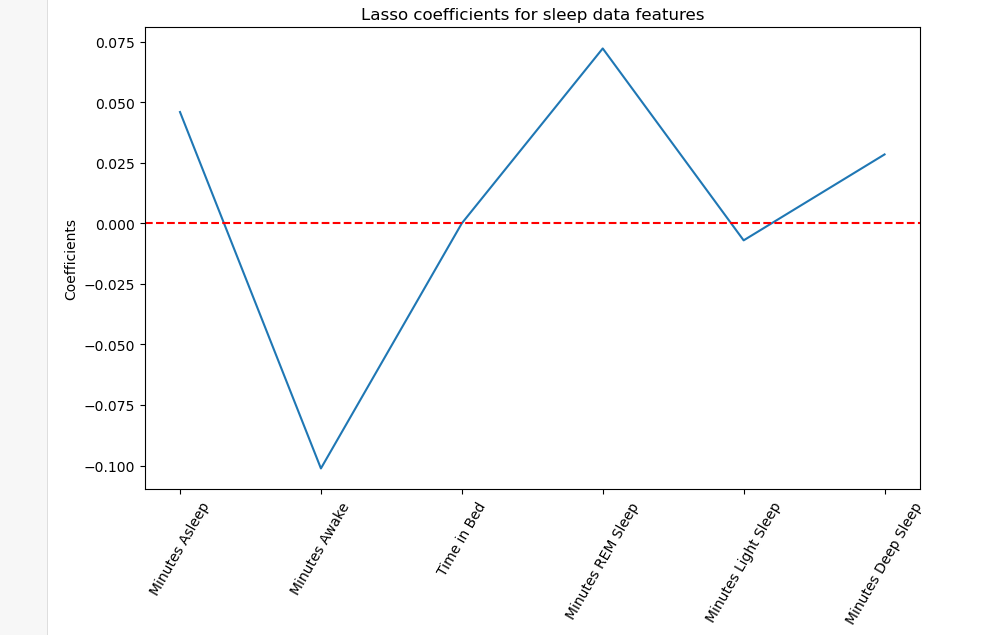
  Description automatically generatedIf we want to preserve the original range of the column and our data **doesn't have outliers** that could heavily skew the scaling, Min-Max scaling might be appropriate. If our data has outliers or doesn't follow a uniform distribution, Standard scaling might be a better choice as it provides robustness against outliers and standardizes the data around its mean and standard deviation. Hence, we use Min-Max scaler.

1. **Feature Selection Using Lasso Regression:**

* We would face major multicollinearity issues if we included all the features in our models i.e., having large coefficients can lead to **overfitting.**
* To solve this problem, we use **Lasso Regression (L1 regularization)** to select important features of a dataset. This is because it tends to shrink the coefficients of less important features to close to zero. we would face major multicollinearity issues if we included all the features in our models, so they are dropped.

A screenshot of a computer

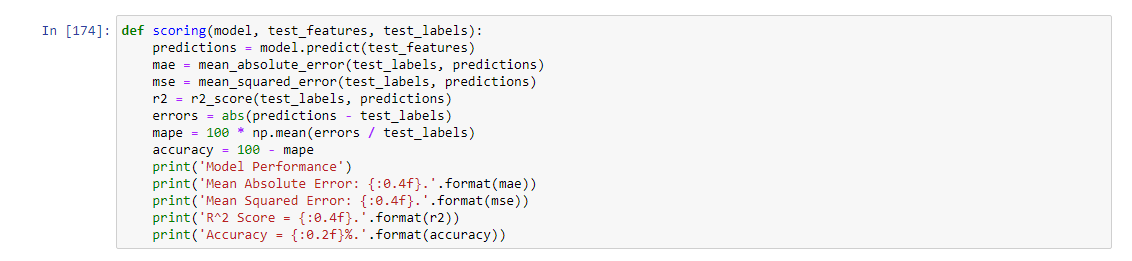
Description automatically generated



* While the scatter plot suggests a strong positive relationship between "Time in Bed" and the overall sleep score, Lasso regression, through its feature selection process, may have identified other features as more influential for predicting sleep scores, leading to the reduction of the coefficient for "Time in Bed" towards zero. This discrepancy highlights the importance of considering both raw relationships and model-driven feature selection techniques in data analysis.

1. **Defining performance measures:**

* We use three ways to measure how well our ML models perform: **Accuracy, Score, and Average Error.**
* Even though Accuracy is usually for classifying things, we're using it differently to see how close our predictions are to the real values, like how accurate our sleep score predictions are.



1. **Establish a Baseline:**

* We'll compare our Machine Learning models by seeing how well they predict Sleep Scores compared to just guessing the average Sleep Score every night. A good model should be much better than just guessing the average.

A white screen with black text

Description automatically generated

1. **Multi Linear Regression:**

* MLR is utilized to analyze the relationship between multiple sleep-related metrics and the overall sleep score.
* It **estimates coefficients for each independent variable** to predict changes in the dependent variable (overall sleep score).
* MLR provides a **simple and interpretable way to analyze relationships** but may **not capture complex nonlinear patterns** in the data.
* A white box with text

  Description automatically generatedIt serves as a **baseline model for comparison** with more advanced techniques like Random Forest Regression or Extreme Gradient Boosting.
* this code segment is likely used to visually assess the performance of the MLR model by comparing its predictions to the actual values on a scatter plot, while also showing a reference line indicating perfect predictions.

**A screen shot of a graph

Description automatically generated**

**Reason for Using Random Forest and XGBoost:**

**Flexibility:** Both XGBoost and Random Forest are versatile algorithms capable of handling both classification and regression problems. This flexibility makes them attractive choices for various types of predictive modeling tasks.

**Non-linear Relationships:** Sleep score prediction might involve complex, non-linear relationships between input features and the target variable (sleep score). Decision tree-based models like Random Forest and XGBoost are well-suited for capturing such non-linear relationships.

**Robustness to Outliers:** Both Random Forest and XGBoost are robust to outliers and noisy data, which can be common in real-world datasets. They can handle outliers gracefully without significantly impacting the overall model performance.

**Ensemble Methods:** Both Random Forest and XGBoost are ensemble learning methods, which means they combine the predictions of multiple individual models (decision trees in this case) to produce a final prediction. Ensemble methods often lead to more accurate and stable predictions compared to individual models.

**Interpretability:** While decision trees themselves are interpretable, the ensemble models built upon them (Random Forest and XGBoost) provide a balance between interpretability and predictive performance. This can be advantageous when you need insights into how the model is making predictions for sleep scores.

**Performance:** Random Forest and XGBoost are known for their high performance and scalability. They can handle large datasets efficiently and often outperform other algorithms in terms of predictive accuracy.

1. **Random Forest Regression:**

* Random Forest Regression gives accurate predictions by using many decision trees together, which **helps handle tricky data** patterns well.
* It's sturdy against overfitting and can **deal with missing information**, making it a dependable option for making predictions in different situations.
* this code segment is likely used to visually assess the performance of the MLR model by comparing its predictions to the actual values on a scatter plot, while also showing a reference line indicating perfect predictions.

A screenshot of a computer

Description automatically generated

A screen shot of a graph

Description automatically generated

**A screen shot of a graph

Description automatically generated**

1. **Extreme Gradient Boosting:**

* XGBoost makes really accurate predictions because it **learns from its mistakes**, improving each time it makes a new guess.
* A screenshot of a computer

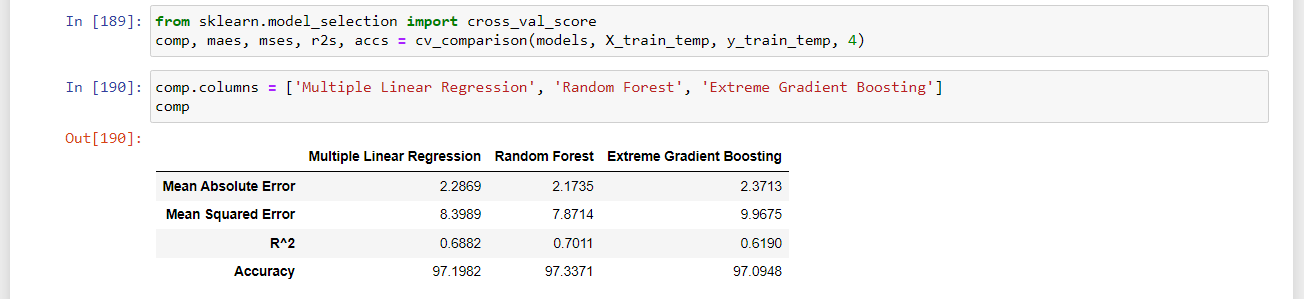
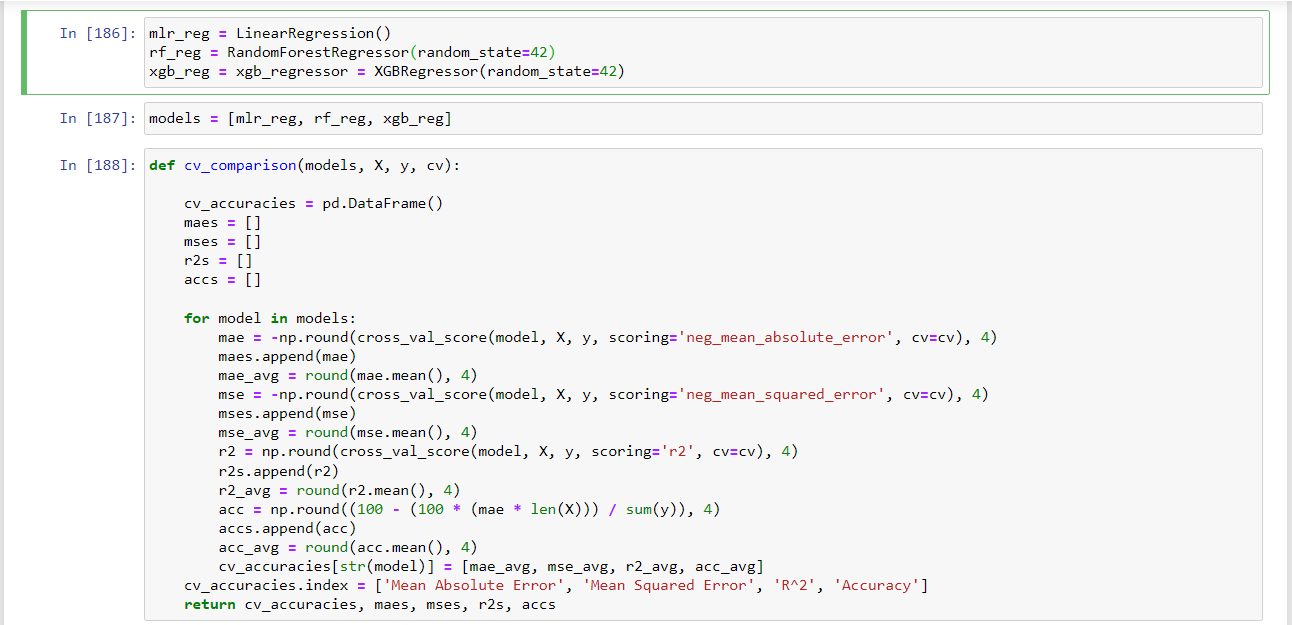
  Description automatically generatedIt also stops itself from getting too complicated, so it doesn't just memorize the data, and it helps us understand which factors are most important for making predictions.

A screen shot of a graph

Description automatically generated

1. **Cross Validation:**

* To prevent overfitting or underfitting with a small dataset of sleep scores, Cross Validation is used. It repeatedly splits the data into training and testing subsets, ensuring thorough evaluation of the model's accuracy.



1. **Using the test data:**

* The models will be tested using the untouched 20% holdout set. Utilizing the entire 80% of the original dataset for training, optimal hyperparameters obtained from randomized grid search for Random Forest and Extreme Gradient Booster will be applied.A screenshot of a computer

  Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Comparison of result of Predictors:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Linear Regression** | **Random Forest** | **Extreme Gradient Boosting** |
| **Mean Absolute Error** | 1.9623 | 1.9289 | 2.2042 |
| **Mean Squared Error** | 6.4451 | 6.0976 | 9.0791 |
| **R2 Value** | 0.7888 | 0.8002 | 0.7025 |
| **Accuracy** | 97.5790 | 97.6145 | 97.2408 |

* **Random Forest** and **Extreme Gradient Boosting** **models** outperform the **Linear Regression model** across all metrics.
* **Mean Absolute Error (MAE)** and **Mean Squared Error (MSE)** are lower for Random Forest compared to Extreme Gradient Boosting, indicating slightly better predictive performance in terms of these metrics.
* **R2 score** is higher for Random Forest compared to both Extreme Gradient Boosting and Linear Regression, indicating better model fit.
* **Accuracy** is highest for Random Forest, followed closely by Extreme Gradient Boosting and then Linear Regression. This indicates that Random Forest and Extreme Gradient Boosting provide more accurate predictions of sleep scores compared to Linear Regression.
* Overall, both Random Forest and Extreme Gradient Boosting models are suitable for predicting sleep scores, with Random Forest exhibiting slightly better performance across most metrics.
* Reason for better performance by Random Forest:
  + **Robustness to Overfitting**: Random Forest inherently tends to be less prone to overfitting compared to XGBoost, especially when dealing with noisy or limited data. This is because Random Forest averages predictions from multiple decision trees, which reduces variance and helps to generalize better to unseen data. In contrast, XGBoost builds trees sequentially, which can lead to overfitting if not properly regularized or if the data is noisy.
  + **Interpretability:** Random Forest models are generally easier to interpret compared to XGBoost. Each decision tree in the Random Forest can be visualized and interpreted individually, providing insights into which features are most important for predicting sleep scores. This interpretability can be valuable for understanding the factors influencing sleep quality and for communicating the model's predictions to stakeholders.
  + **Less Sensitivity to Hyperparameters:** Random Forest has fewer hyperparameters to tune compared to XGBoost, making it less sensitive to hyperparameter choices. While hyperparameter tuning is still important for optimizing performance, Random Forest may require less computational resources and effort for tuning compared to XGBoost, which has more hyperparameters and a more complex optimization process.
  + **Handling of Non-linearity and Interaction Effects:** Random Forest is capable of capturing non-linear relationships and interaction effects between features effectively, which is crucial for predicting complex phenomena like sleep quality. Although XGBoost is also capable of capturing non-linear relationships, Random Forest's ensemble of decision trees may be more robust in capturing complex interactions in the data without explicitly optimizing for them.
  + **Computational Efficiency:** Random Forest training and prediction are typically faster compared to XGBoost, especially for large datasets. Random Forest constructs decision trees independently in parallel, whereas XGBoost builds trees sequentially, which can be computationally intensive, especially for deep trees or large datasets.

1. **Conclusion:**

Based on these results, either Random Forest or Extreme Gradient Boosting can be selected for predicting sleep scores, with Random Forest being slightly preferable due to its marginally better performance in terms of accuracy and predictive power.

**Data set links:**

<https://github.com/JoBe10/Fitbit-Sleep-Score/blob/master/Fitbit_Sleep_Score_JB_041219_010720.csv>

<https://github.com/JoBe10/Fitbit-Sleep-Score/blob/master/Fitbit_Sleep_JB_041219_010720.csv>

**References :**

A Study on ML-Based Sleep Score Model Using Lifelog Data - Jiyong Kim and Minseo Park ,

Appl. Sci. 2023, 13, 1043. <https://doi.org/10.3390/app13021043>

<https://www.mdpi.com/journal/applsci>

A screenshot of a document

Description automatically generated