DS740, Final Project: SleepQuality analysis from Garmin and LoseIt data

John Woodward

2023-08-09

Introduction

Imagine feeling refreshed every morning—awakening energized, mentally resilient, and physically revitalized each morning. This machine learning analysis (e.g., Decision Trees, Gradient Boosting) of my two-year biometric data delves into what distinguishes great, OK, and poor sleep, which may guide more informed lifestyle improvements.

Moreover, this exploration is a gold mine for Garmin, offering valuable insights into sleep. For example, imagine: what if there was a successful `SleepQuality` classification? What if Garmin added it to their software as an automatic `SleepQuality` metric? Wouldn't Garmin boost their sleep tracking device's value and improve users' health and happiness? It would be a chance for Garmin to gain a decisive competitive edge in the sleep-tracking market.

Data Preparation

Data Organization

My project's dataset includes several tables:

1. `food` - a transactional, many-per-day export from LoseIt.com with food details.
2. `exercise` - a transactional, many-per-day export from LoseIt.com containing exercise information.
3. `agg` - a daily CSV from LoseIt.com with biometric data.
4. `sleep` - a daily CSV from Python combining sleep data from Garmin and Apple sources.

The primary table is `sleep`, supplemented by `agg` variables, and later, `food` and `exercise` will aggregate to predict `SleepQuality.`

Data Extraction, Loading, and Transforming (ETL)

I began by preparing the food and exercise datasets, refining them, and importing their CSV files. Then, I enriched the data with aggregated information from LoseIt.com, further refining and importing the CSV. Enhancements included: converting datatypes, rolling averages, statistical smoothing, handling missing data, addressing outliers, and introducing relevant time lags.

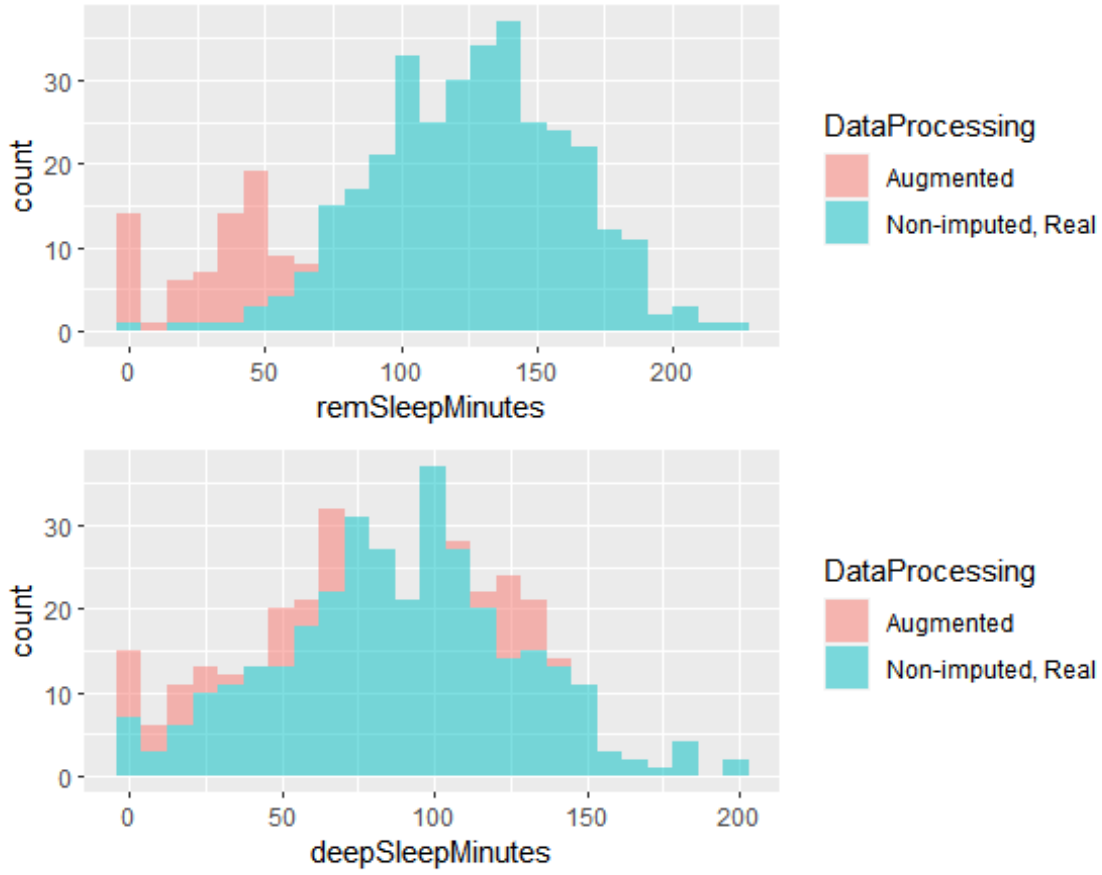
Finally, I focused on the primary 'sleep' dataset. I used Python to collect and merge daily sleep data from Garmin (e.g., JSON) and Apple (e.g., XML) health data exports. I meticulously handled the information to create a well-organized and structured dataset.

Exploratory Data Analysis on `SleepQuality` via Clustering

For strategically assessing `SleepQuality`, I wanted a simplified interpretation of this response. After exploring references, comparing recent sleep experiences to measurements, and data trial and error, my focus narrowed to two key sleep measures: `deepSleepMinutes` and `remSleepMinutes`. However, the data revealed a bias within my sleep: a lack of low remSleepMinutes.

As Figure 1 shows, `remSleepMinutes` has significantly fewer data count compared to `deepSleepMinutes`, which explains why I needed to augment the data:

Figure 1: Clustering Variables



Two possible explanations for the variation exist. First, the COVID-19 pandemic (2021-2022) amplified anxieties and multiplied `remSleepMinutes`. Second, my consistent healthy sleep habits, such as avoiding sleep deprivation and alcohol, could account for higher `remSleepMinutes`. It's important to note that this dataset is personalized, potentially leading to irregularities. Thus, to improve clustering, I augmented low `remSleepMinutes` data to represent "Poor" sleep quality better.

Clustering Data Augmentation

Using an `augment_data` function, I generated variations of the original `sleep` dataset by perturbing single data points with normal distribution-derived alterations. This augmentation encompassed different sleep scenarios, such as "very poor sleep," "poor REM sleep," and a "poor-ish REM/Deep sleep." For example, the "very poor" data point was a 0 deepSleepMinutes and a 0 remSleepMinutes, a miserable all-nighter. With perturbed duplicates created for three poor REM situations, I inserted them back into a separate sleep clustering dataset (alongside an indicator to keep track of them) and then scaled them for non-hierarchical clustering.

Clustering Tuning

Next, I performed a non-hierarchical clustering analysis determining the optimal cluster counts (`K`) ranging from 2 to 6. I iteratively applied the `kmeans` function, employing seven different initial splits within each cluster. I then compared clustering results for each and calculated matching proportions to assess similarity to decide which `K` made sense to consider.

This unsupervised approach helped with learning what clusters consistently repeated among random trials, as you can see in Table 1 below:

Table 1: K-means Cluster Match Proportions (median of 7 runs each K)

K	Median_Match_Proportion
2	0.724
3	0.990
4	0.946
5	0.813
6	0.760

Table 1 shows that 3 or 4 clusters are great selections with the data and the augment. However, I chose *4* clusters as it allows for a potential "poor" `SleepQuality` category (Figure 2).

Clustering and Classifying `SleepQuality`

Therefore, I employed k-means clustering on the scaled sleep quality data using *4* clusters. I ran extra initial iterations (i.e., 10,000 additional runs) to increase the odds of optimizing clusters globally.

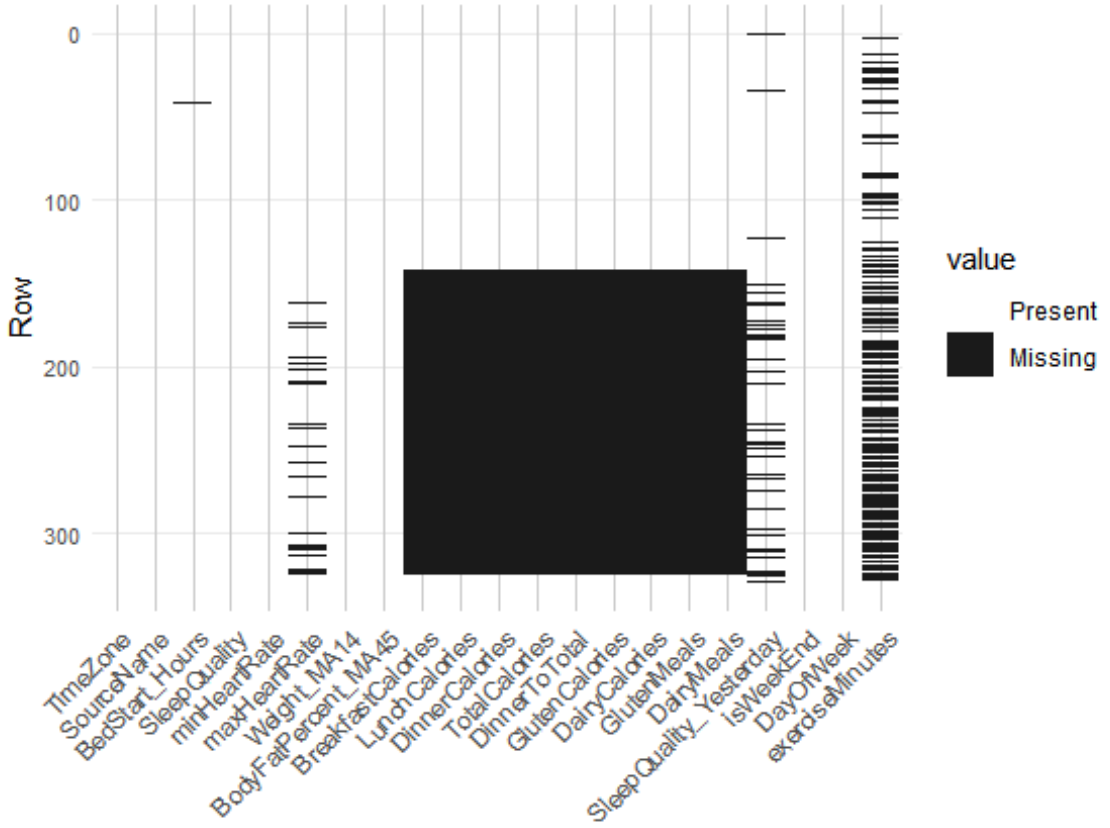
Figure 2 visually shows: (1) four colored clusters that *make sense* - for instance, having low `remSleepMinutes` but high `deepSleepMinutes` (or vice versa) is just "OK" `SleepQuality`; (2) and two shapes differentiating virtual and actual data- notably, clearly showing the low `remSleepMinutes` augmented:

Figure 2 also shows how I classified `SleepQuality`. First, I used the cluster centers and summed the components for each cluster. By strategically positioning the lower-left (i.e., low REM and Deep sleep) as "Poor" and the upper-right (i.e., high REM and Deep sleep) as "Great" on the scatter plot, I could classify the rest of the clusters as "OK" `SleepQuality`.

Further Data ETL and Missingness

Ultimately, I finished the last touches of preparing the core sleep dataset. Please see in Figure 3 the tidied dataset, `zzz`, with its moderate missingness:

Figure 3: Missingness of Sleep Data (where SleepQuality is known)



Machine Learning Model Fitting

Model Fitting

Missing values are the bane of most machine learning and often require discarding or replacement. Yet, the methods I pre-selected, Decision Trees (e.g., "rpart") and Boosting (e.g., xgbTree), resourcefully use NAs: NAs are usable if an observation has the following:

1. a **dependent variable** and
2. **at least one** non-NA independent variable

As such, and because Decision Trees prune variables, I will use just one model:

`Model`: `SleepQuality` ~ .

But, because one `SleepQuality` class, "Poor", has just 12 out of 330 rows, I decided to consider a two-class version: "Great" or "NotGreat". Therefore, I will vary the response label like so:

1. a three-class response: "Great", "OK", or "Poor".

2. a two-class response: “Great” or “NotGreat”.

Cross-Validations, 5-fold, Single and Double

As the dataset contains only 330 rows of non-NA `SleepQuality` and missing information, this affects the hyperparameter tuning for optimal model performance. Likewise, one of the datasets had a low amount of classes (“Poor” with 12 of 330), so I needed to use “Kappa” for optimization, a classification metric that measures agreement while accounting for chance. Similarly, I needed stratified 5-fold groups in my single cross-validation because of the severe class imbalance. By experimenting, I changed the range of values for specific parameters to balance computation time and maximize Kappa.

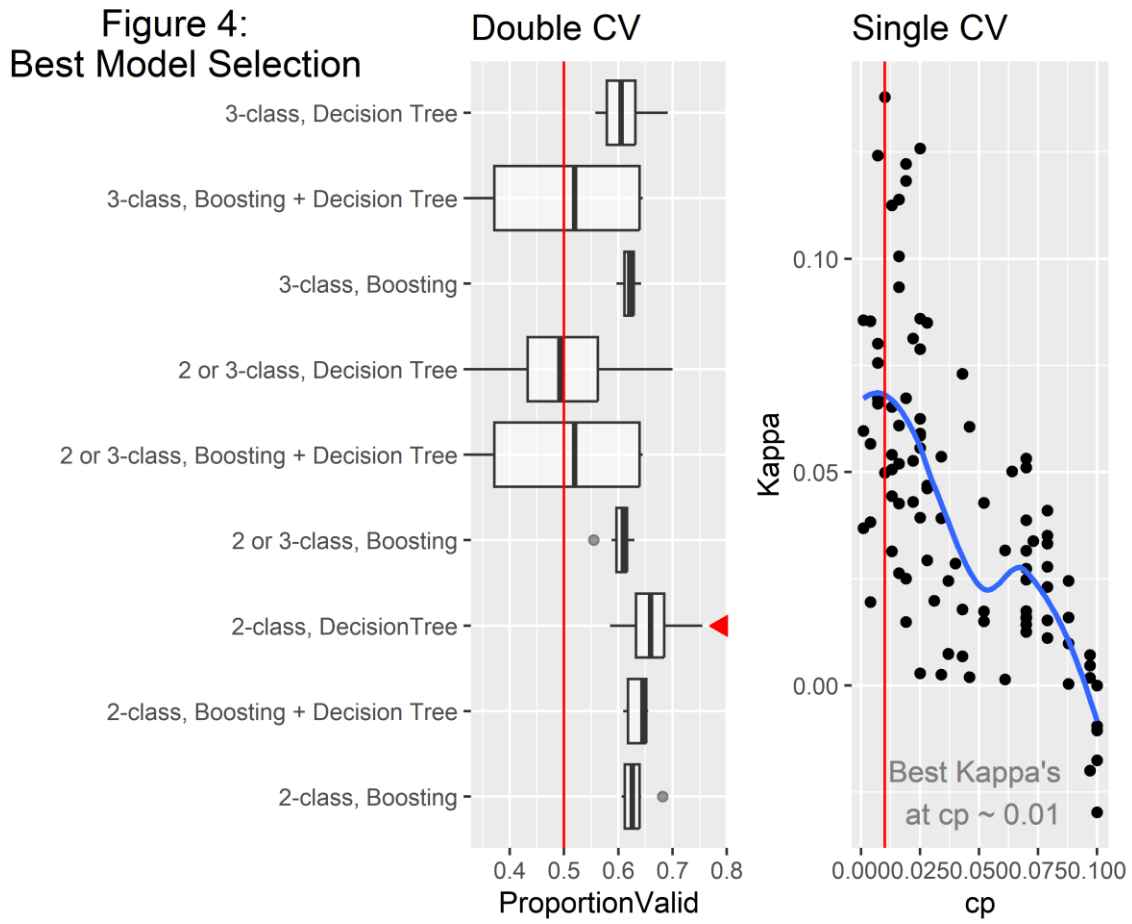
For the method “rpart,” I tuned the following:

- `cp` (complexity parameter) - between 0.001 and 0.1 (increments of 0.003). I found that increments of 0.001 wasted computational time as it repeated Kappa multiple times; thus, I settled with 0.003.

For the method “xgbTree,” I tuned the following:

- `gamma`, `min_child_weight`, and `subsample` are constant 0, 1, 0.45, respectively. These three parameters were mostly the same for my problem, and I did not need to optimize.
- `max_depth`, or the maximum depth of the tree, I found the answer every time was 1. As the dataset was small, and the computation time was about 17 seconds per run at 1, I wanted to keep it simple and rely more on the `nrounds` for complexity tuning.
- `nrounds`, or the number of boosting trees; I varied this one in fine increments of 1 from 5 to 20.
- `eta`, or the learning rate for the gradient boosting algorithm, can be the difference between finding a local optimum or a global optimum, so I also varied it from 0.1 to 0.3 by 0.01.
- `colsample_bytree`, or the proportions of the features used in `nrounds`, I varied by either 0.5 or 0.8 as this affected optimization.

Best Model



I selected the best model and tuning parameters using results from double cross-validation and single cross-validations, visualized in Figure 4:

1. *Left*: The Decision Tree method "rpart" performed better than "xgbTree" with less overfitting, leading to a surprising yet favorable choice due to its interpretability.
2. *Left*: The 2-class label tends to be less variable than the 3-class for the response variable, favoring a 2-class Decision Tree model.
3. *Right*: Thus, after 100 random single cross-validations on the 2-class Decision Tree model, the higher Kappa values are around a cp of 0.1, guiding the best model selection and hyperparameter.

We have our best model after fitting a Decision Tree model to the full dataset, and pruning the tree with a cp at 0.1.

Interpretting Best Model

Visualizing Best Model

Decision Trees have a reputation as one of the most interpretable machine learning; Figure 5 shows why:

Figure 5: Decision Tree for John's `SleepQuality` Prediction

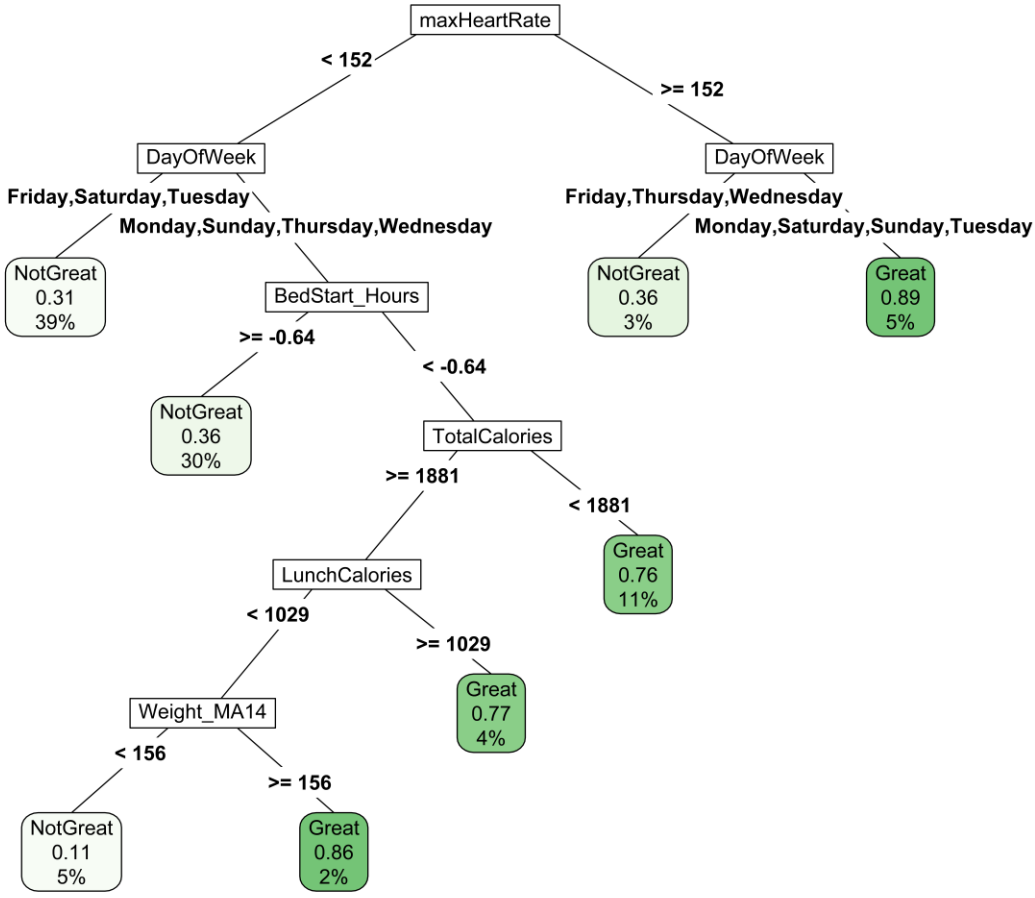


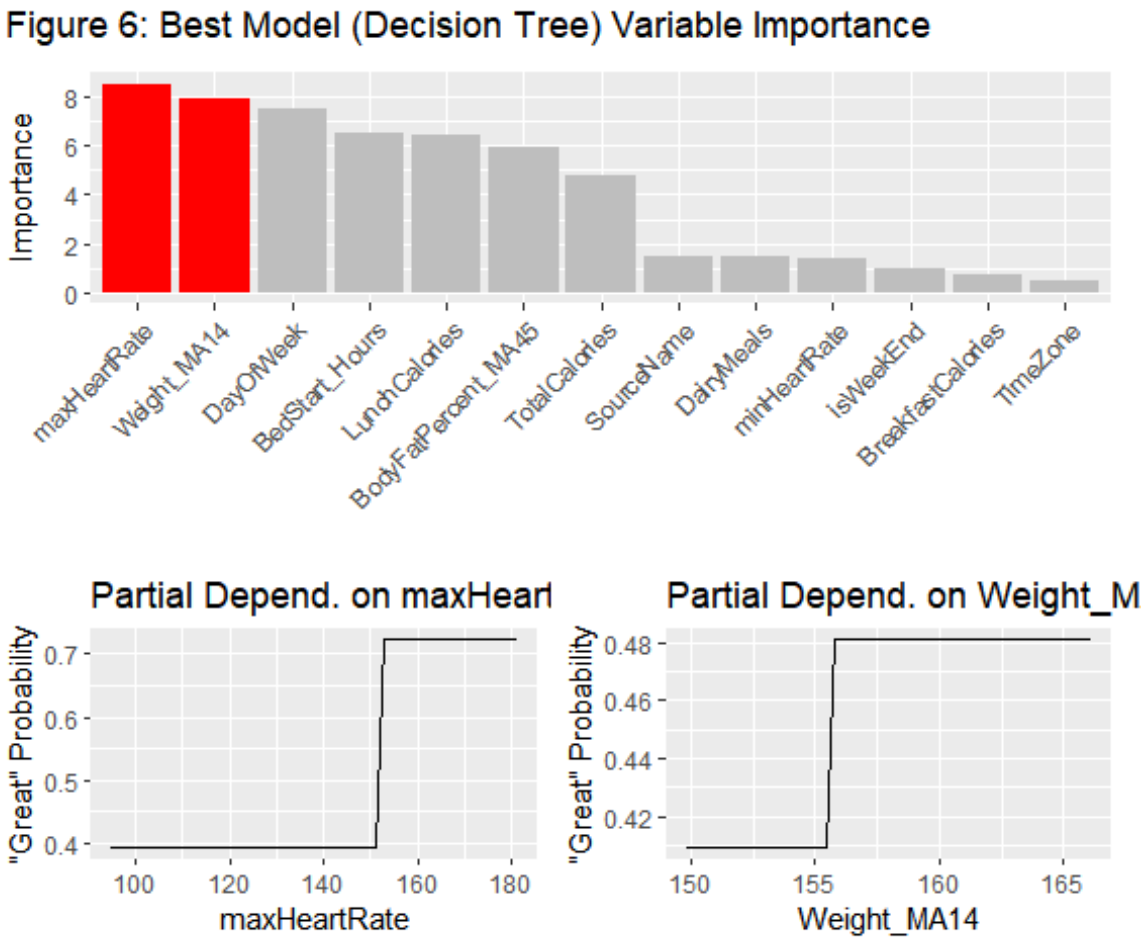
Figure 5 shows a few insights for me:

- My `maxHeartRate` is the most crucial variable for `SleepQuality`, especially at the **152 BPM** threshold; I better exercise more intensely!
- `BedStart_Hours` -0.64 translates to a threshold of 11:21 PM. This *makes sense* because the later I enter the bed with my phone, the less sleep duration I get due to my regular biological or occupational wakeups around 7-10 AM.

- `TotalCalories` and `LunchCalories` indicate that if my `Weight_MA14` exceeds 156, I should have "Great" sleep. So this makes sense because if I am underweight and likely eating a massive dinner (or breakfast), my gastrointestinal system may wreck my SleepQuality.
- An uncontrollable variable is included: `DayOfWeek`. It reflects my weekly patterns (e.g., hot yoga Fridays with my wife, weekday working).

Variable Importance

Figure 6 shows the variable importances from the best model of the Decision Tree:



Relationship of Predictors with Response

In Figure 6, the partial dependence plot shows:

1. *Left*: Shows how higher maximum heart rates correspond to an increased likelihood of "Great" sleep quality. This suggests intense exercise may promote better sleep due to enhanced melatonin production.

2. *Right*: Highlights a connection between rolling average body weight (Weight_MA14) and the "Great" sleep quality probability. Around 155 lbs mark a significant threshold, linked to a month of weight loss and lower sleep quality.

Model Accuracy and Prediction Use

Given a binary class scenario, random chance predictions would yield around 57% accuracy by guessing the majority class ("NotGreat"). Our model, however, achieves approximately 65% accuracy, demonstrating its improved predictive power.

Although our best model struggles to classify "Great" `SleepQuality` according to the ROC curve in the Appendix, there's room for enhancement. Incorporating more diverse data and considering less disruptive periods than the COVID-19 pandemic could reduce misclassification and enhance prediction accuracy.

Appendix

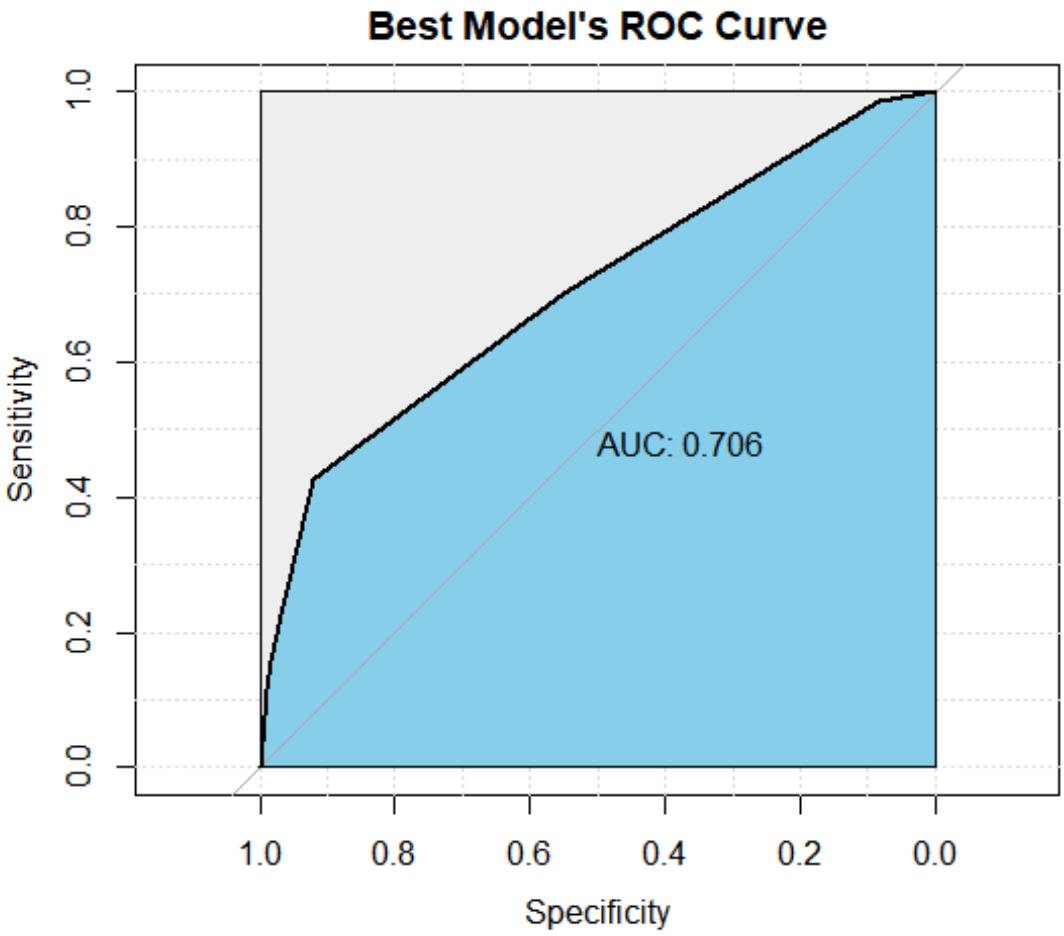
Association rules

Table 2: Top 5 Association Rules (supp = 0.005,conf = 0.7)

rules	support	confidence	coverage	lift	count
{Icon=MilkShake,Meal=Breakfast} => {SleepQuality=Great}	0.0052	0.7778	0.0067	2.0126	7
{Icon=Milk,Calories=[103,160),HasDairy=FALSE} => {SleepQuality=Great}	0.0052	0.7778	0.0067	2.0126	7
{Icon=Milk,Meal=Breakfast,Calories=[103,160)} => {SleepQuality=Great}	0.0060	0.7273	0.0082	1.8819	8
{Icon=Milk,Calories=[103,160)} => {SleepQuality=Great}	0.0127	0.7083	0.0179	1.8329	17
{Icon=Curry,Calories=[264,1.48e+03],HasDairy=TRUE} => {SleepQuality=Great}	0.0052	0.7000	0.0074	1.8114	7

While the above table makes sense, a morning glass of milk gives me enough energy as a breakfast that is quick, easy, and helps the day; I have been avoiding dairy for more than a year due to a tested dairy allergen, which is surprising. I also was surprised to find out that a large curry meal with dairy (e.g., Palaak Paneer) may help my sleep.

ROC Curve


Best Model's ROC Curve
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
AUC: 0.706
1.0
0.8
0.6
0.4
0.2
0.0
Specificity

References

Page 18, Section 5: Missing Data An Introduction to Recursive Partitioning Using the RPART Routines Terry M. Therneau Elizabeth J. Atkinson Mayo Foundation October 21, 2022

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4486966/

https://www.kaggle.com/code/pelkoja/visual-xgboost-tuning-with-caret