# Using non-linear, machine learning methodology to assess the potential metabolomic-based biomarkers of total fat and percentage fat intake using a controlled feeding study.

Caroline Lea Nondin

A thesis
Submitted in partial fulfillment of the
Requirements of the degree of

Master of Science

University of Washington
2023

Committee:
Marian Neuhouser
Eardi Lila

Program Authorized to Offer Degree:
Nutritional Sciences

University of Washington

## Abstract

Using non-linear, machine learning methodology to assess the potential metabolomic-based biomarkers of total fat and percentage fat intake using a controlled feeding study.

Caroline Lea Nondin

Chair of the Supervisory Committee:

Marian Neuhouser

Nutritional Sciences Department

**Background:** Understanding and identifying objective dietary biomarkers is a crucial component of nutrition research today. By investigating the relationship between biomarker profiles and dietary intake using machine learning methodologies, there could be a way to more objectively assess study participant nutrient profiles and better understand the relationship between nutrient intake and disease. Our aim in this thesis is to assess the utility of non-linear tree-based models in predicting daily intake of total fat and the percent of energy from fat from serum and 24-h urine high dimensional metabolites.

**Methods:** Our analysis used the dataset from a 2-week controlled feeding study mimicking the participants' habitual diets among 153 post-menopausal women from the Nutrition and Physical Activity Assessment Study Feeding study, conducted in the Women's Health Initiative (WHI). Fasting serum metabolite profiles, urine metabolites, as well as demographic and Food Frequency Questionnaire (FFQ) data, were used to predict total fat and percent energy from fat using four cross-validated tree-based machine learning models. A LASSO model for regression was used as a way to compare the models to a linear model.

**Results:** The highest cross-validated multiple correlation coefficients (CV-$R^2$) for total fat intake and percent energy intake were 10.2% and 10.4%, respectively. None of the models had a CV-$R^2$ of over 36%. There were no significant differences found between the performance of linear and non-linear models in predicting fat intake.

**Conclusion:** Both linear and non-linear models were shown to be unable to predict total fat and dietary fat intake using serum and urinary metabolites accurately and reliably. Variable importance suggests that tree-based, machine-learning models have the potential to help understand non-linear interactions between biomarkers and dietary intake.

# 1. Introduction

## 1.1. Bias and the search for objectivity in Nutritional Science research

Understanding the relationship between diet and the risk of disease is at the forefront of nutrition research today. However, research in Nutritional Science regarding these associations is hindered by our current methodological approaches. One crucial issue is the reliance on participant self-report to assess their diet, such as the commonly used Food Frequency Questionnaire (FFQ). The reliance on self-report in these methodologies can lead to systemic bias and measurement error that can influence our understanding of the diet-disease relationship[1]. Self-report bias can be due to participants inability to fully recall their dietary intake, intentional misreporting, and/ or due to the natural variations in food preparations and quantities that are not reflected by the food composition database used to calculate energy and nutrient intake from the FFQs[1]. There is, therefore, a critical need to incorporate objective measures, such as metabolomic biomarkers, into these epidemiological methodologies.

Previous work from the Women's Health Initiative (WHI) has provided a strong foundation for correcting self-report biases using regression calibration methods. These methods using nutritional biomarkers have been used to successfully identify biomarker signatures of dietary patterns [2], to assess macronutrient intake based on serum and urine metabolites[3] and to calibrate self-reported diet when examining the association between macronutrients and chronic disease risk[4] [5]. So far, there is strong evidence suggesting that carbohydrate and protein intake can be predicted with relatively high accuracy[3] using participant biomarker panels, which has significant implications for the future of nutritional epidemiology. However, current methodologies using linear models have not accurately predicted fat intake, perhaps due to the interconnected and non-linear nature of lipid metabolomic pathways[3]. Therefore, a nonlinear or non-additive model could be tested on the metabolomic data to make better predictions.

## 1.2. Nutritional Metabolomics

Nutritional metabolomics is an emerging field that uses metabolites, or small molecules, in biological samples to help bring objective measures to nutrition[6]. Metabolomics is most often used in the field of nutrition to objectively assess dietary exposure and thus provide an alternative or complement to self-reporting[7]. Research in nutritional metabolomics has so far

mainly focused on very specific dietary patterns or food groups[6] or has focused on the link between various metabolites and disease. However, more general macronutrient analyses using metabolomics are fewer primarily due to the highly interconnected and complex nature of macronutrient metabolism[3]. Urine Nitrogen has been found to be a consistently reliable biomarker for protein intake[3], and serum phospholipids combined with participant characteristics for total carbohydrate intake[3]. Objective biomarkers for fat intake remain to be found, due to the highly complex and interconnected nature of lipid metabolism[3].

### 1.3.Machine Learning: a promising modeling tool for Nutritional Epidemiology

Over the past few years, there has been an increased interest in machine learning practices in nutrition research. With an upsurge in large datasets and new technologies providing large amounts of information on participant characteristics, there is a need for new methodologies able to recognize patterns and assess the importance of factors that would not be picked up by more traditional analytical methods [8]. Machine learning is an area of artificial intelligence that can improve modeling and find non-linear associations that could not be assessed using conventional methods of data analysis[9]. So far, there have been promising results using machine learning algorithms to predict disease states based on biomarker profiles, such as predicting cancer based on serum metabolic patterns[10] or using biomarkers to assess critical features of blood pressure regulation[11]. By investigating the relationship between biomarker profiles and fat intake using machine learning methodologies, there could be a way to more objectively assess study participant nutrient profiles and better understand the relationship between nutrient intake and disease.

## 1.4.Objectives

This thesis aims to investigate the interconnect and nonlinear links between metabolomic-based biomarkers and dietary fat intake using machine learning methodology. The aims of this study are as follows:

1. Using non-linear tree-based models, accurately predict fat intake (in g/day) and energy percentage from fat using serum and 24-h urine samples from the Nutrition and Physical Activity Assessment Study (NPAAS). The models are expected to achieve a CV-$R^2$ of over 36%.

2. Compare the results to a linear LASSO model to assess whether non-linear machine learning methods yield consistently better predictions than linear regression prediction models.

3. Assess which variables, including metabolites and participant characteristics, had the most significant impact (variable importance) on the tree-based machine learning algorithm with the best performances based on their relative influence on the final prediction.

## 2. Methods

### 2.1. Dataset Collection and Information

All data for this study are from the Women's Health Initiative (WHI) database with additional data from WHI ancillary studies. Participant information and samples were taken from the 'Nutrition and Physical Activity Assessment Study-Feeding Study" (NPAAS-FS), which is an ancillary study to the WHI[12]. This study included 153 women. At the time that NPAAS-FS was conducted, participants had been enrolled in the Observational Study cohort, the Dietary Modification Trial Comparison group or Hormone Therapy Trial and were will enrolled in WHI follow-up. All participants were located in the Seattle area at the time of the study. Participants were also required to have full follow-up status within the WHI, were over the age of 79 at the time of study, and did not have any medical conditions that would hinder their ability to successfully complete the protocol (eligible medical conditions include diabetes, kidney disease, bladder incontinence requiring the use of special equipment or routine use of oxygen)[12]. Inclusion criteria and recruitment details can be found in Appendix B.

#### 2.1.1. The NPAAS-FS study design

The NPAAS-FS study was a feeding study that took place over the span of two weeks. Each participant was fed an individualized diet that mimicked each participant's 4-day food record (4FDR) completed as part of pre-study activities. Participants were also asked to complete a daily menu checklist during the study to assess compliance. All foods for the two-week study period were prepared by the Fred Hutchinson Prevention Center Human Nutrition Laboratory (HNL). Participants were fed one study meal on-site approximately 2-3 times per week during the 2-week feeding study. The participants picked up the remainder of the food provided to take home and consume in the following days. Unconsumed foods were returned to be weighed and recorded. The goal of this feeding study was to approximate participants' diets without perturbing blood and urine measures and to preserve variations in nutrient and food consumption over the two-week study period. The nutrient content of participant's diet was assessed using the Nutrition Data System for Research software (version 2010; Nutrition Coordinating Center, University of Minnesota). These data were used to determine the dietary variables used in this study.

### 2.1.2. Data collection and metabolite profiling

24-h urine collections used to assess participant's metabolomic profile were made on day 13 of the two-week feeding period. Serum samples were collected after a 12-h overnight fast on day 14. Participant characteristics, including age, dietary supplement use, season of participation, age, BMI, and self-reported physical activity, were assessed at the time of study enrollment in the NPAAS-FS. Other characteristics such as ethnicity, education level, smoking status, and baseline FFQ responses used in this analysis were measured at the time of enrolment in the WHI. Serum metabolites were analyzed by targeted LC-MS/MS using a mass spectrometer at the Northwest Metabolomics Research Center at the University of Washington. A total of 303 serum metabolites were identified, of which 155 had less than 20% missing values. Serum lipids were extracted using the Lipidizer platform. 1070 different lipids were identified. All 13 classes of lipids were identified by the 2021 paper by Zheng et al.[3]. Overall, 664 serum lipids with less than 20% missing values were measured.

24-H urine samples were analyzed using $^{1}$H NMR spectroscopy. Overall, 57 metabolites were identified, none of which had missing values. Separately, the urine samples were also analyzed using global GC-MS. From this, 285 metabolites were identified, 275 of which had less than 20% missing values.

The full detailed description of the metabolite measurements and selection process has been published by Zheng et al. (2021)[3].

## 2.2. Statistical Analysis

### 2.2.1. Data Preprocessing

All data preprocessing and data analysis were done using RStudio version 2021.09.0 and Microsoft Excel version 16.74.2. All metabolomic variable preprocessing was made available by the WHI NPAAS-FS database. All metabolites with more than 20% missing values were removed from the dataset to ensure robust results. Diet and metabolite variables were log-transformed and truncated to Q1-3*IQR and Q3+3*IQR to be consistent with other analyses in the NPAAS-FS. IQR stands for the interquartile range; Q1 and Q3 represent the first and third quartiles, respectively. LC-MS and GC-MS data were normalized using local polynomial regression fitting.

Because of a slight weight change among participants during the study (ranging from -3.6 to 2.4kg), total energy intake (Ein) was used as a biomarker for energy intake. Ein was calculated using total energy expenditure (TEE) and weight difference between the last and first weight measurements during the NPAAS-FS study.

Age, BMI, ethnicity, season of participation, education level, and weekly physical activity were made into categorical variables in order to stay consistent with previous NPAAS-FS studies[3].

A random forest model was used to impute missing values. Approximations were calculated using participant characteristics and existing values from the variables containing missing values[13].

Once the data were processed, 4 datasets were created for the analysis. Each dataset builds on the previous one, meaning that more variables were added as each new dataset was created. For instance, dataset 2 is composed of all variables from dataset 1, plus participant's demographic characteristics. More details on dataset composition can be found in the footnote of tables 3 and 4.

### 2.2.2. Model Building

For this analysis, we wanted to run and compare various tree-based, nonlinear models to a linear model to see which model would achieve the best predictions of total and percentage fat intake. In total, 10 analyses were performed: five to determine total fat intake (g/d) and five to determine the percentage of kcal from fat (%E). A different analysis was run for each of the 4 datasets, for a total of 4 analyses per outcome. Four tree-based models were selected for comparison: pruned

tree, random forest, boosted tree, and Bayesian Additive Regression Tree (BART), using rpart, ranger, xgbTree and bartMachine packages in R, respectively. A LASSO regression model (the glmnet package in R) similar to the model used in the Zheng et al. paper[3] was also evaluated to compare the performance of non-linear models vs. more traditional linear models.

All 10 separate data analyses used very similar structures in order to ensure adequate comparisons and replicability. The dataset was first split into a training and test set with a respective 80/20 split. A 5-fold cross-validation loop was applied to ensure that the data were trained and tested using different splits in order to avoid natural variations in the data to influence our predictions. The caret package in R[14] was used to create a second nested loop where the models would be refined using the training set. Each model would go through 5 iterations using cross-validation in order to select the best hyperparameters in each model. The training set was once again split into a training and validation set with an 80/20 split. This ensured that the models were able to be trained and validated for maximum prediction performance while keeping the original testing set separate to assess model performance. All models were set to maximize $R^2$ performance.

Once the models were trained, the predict() function[15] was used to predict total fat or percent fat using the models and the test set. The predictions were then compared to the actual outcome values to calculate the $R^2$ value. $R^2$ was calculated as 1 – (mean square error (MSE)/ the variation of the outcome in the test set). All five $R^2$ values from the cross-validation were recorded. CV- $R^2$ was calculated as the mean of all five $R^2$ values for each model.

### 2.2.3. Variable Importance

Variable importance was calculated for the best-performing models for predicting total fat and percentage fat. The VarImp()[16] function was used to assess variable importance for BART model, our best-performing tree-based model. The 20 most significant variables were selected since variable importance decreased significantly after that. The importance of each variable was assessed by calculating the variable's "inclusion proportion", or the proportion of times each predictor is chosen as a splitting rule divided by the total number of splitting rules in the model[17]. More information on how the splitting rules are calculated can be found in appendix C.

3. Results

Participant characteristics of the 153 participants can be found in table 1. Most participants were between the ages of 70-79 (83.0%), are white (94.8%), have attended at least some higher education (92.8%) and used dietary supplements (76.5%). The range of BMI values is relatively well distributed between participants, with 39.9% of participants being in the <25.0 $kg/m^2$ range, 39.2% in the 25-30 $kg/m^2$ range and 20.9% above 30 $kg/m^2$. Table 2 lists the metabolite variables used in the analysis, which include serum samples (including serum lipid samples) and 24hr urine samples.

| | Overall (N=153) |
|---|---|
| **Age (years)** | |
| 60-69 | 10 (6.5%) |
| 70-79 | 127 (83.0%) |
| 80-85 | 16 (10.5%) |
| **Race/Ethnicity** | |
| Caucasian | 145 (94.8%) |
| Non-Caucasian | 7 (4.6%) |
| Missing | 1 (0.7%) |
| **Years of Education** | |
| High school/General Educational Development diploma | 10 (6.5%) |
| Schooling after high school | 60 (39.2%) |
| College degree or higher | 82 (53.6%) |
| Missing | 1 (0.7%) |
| **BMI (kg/m2)** | |
| Normal (<25.0) | 61 (39.9%) |
| Overweight (25-30) | 60 (39.2%) |
| Obese (>30) | 32 (20.9%) |
| **Use of any Dietary Supplements** | |
| No | 32 (20.9%) |
| Yes | 117 (76.5%) |
| Missing | 4 (2.6%) |
| **Currently Smoking** | |
| No | 57 (37.3%) |
| Yes | 9 (5.9%) |
| Missing | 87 (56.9%) |
| **Season of study participation** | |
| Spring | 31 (20.3%) |
| Summer | 55 (35.9%) |
| Fall | 42 (27.5%) |
| Winter | 23 (15.0%) |
| Missing | 2 (1.3%) |
| **Recreational physical activity (MET/week)** | |
| 0–5.5 | 39 (25.5%) |
| 5.6–12.25 | 38 (24.8%) |
| 12.3–24.0 | 39 (25.5%) |
| > 24 | 37 (24.2%) |
| **Total Energy Intake (kcal/d)** | |
| Mean (SD) | 1910 (281) |
| Median [Min, Max] | 1900 [1310, 2840] |
| Missing | 7 (4.6%) |
| **Fat (g/d)** | |
| Mean (SD) | 81.2 (18.2) |
| Median [Min, Max] | 78.2 [42.2, 141] |
| **Fat (%E)** | |
| Mean (SD) | 38.0 (5.84) |
| Median [Min, Max] | 37.5 [21.2, 61.4] |

**Table 1:** Participant characteristics and fat intake of the participants in the Women's Health Initiative Nutrition and Physical Activity Assessment Feeding Study (n=153). Dietary intake, including Total Energy Intake, Fat (g/d) and Fat (%) was based on diet as consumed during the feeding study. Age, BMI, use of dietary supplements, season of study participation, recreational physical activity, Total Energy Intake, and fat intake (g/d and %E) were all collected at the time of enrollment in the NPAAS-FS. All other participant characteristics were collected at WHI baseline.

*Table 2: Number of metabolites identified in each platform and their median coefficients of variation (CV%) across the specimens from the participants of the NPAAS-FS study.*

| Platform and biologic samples | Total[a] Features (n) | <20% Missing | CV[c](%) |
|---|---|---|---|
| LC-Q-TOFMS serum (composition) | 1070 | 664 | 5.5 |
| Targeted LC-MS serum (Composition) | 303 | 155 | 7.2 |
| GC-MS 24-h urine | 285 | 275[b] | 31.3 |
| NMR 24-h urine | 57 | 57 | 4.0[d] |

*[a]Total number of identified metabolites/ features*
*[b]137 features are un-identified*
*[c]CV among those features with <20% missing*
*[d]NMR measurements were made in two batches spaced in time by approx. 1 year*

Table 3 and 4 show the prediction accuracy (CV-$R^2$) for total fat intake (g/d) and percentage of calories from fat (%E) using the 5 models and 4 datasets in the analysis. Overall, the BART model using dataset 2 has had the best performance in predicting total fat intake, with a CV-$R^2$ of 10.2%. This is 0.5% higher than the highest performing LASSO model, using dataset 4. Introducing the dietary biomarkers (dataset 3) and FFQ (dataset 4) has decreased the CV-$R^2$ for the BART model by 3.1% and 0.4%, respectively. Using only metabolite data (dataset 1) decreased model performance by 8.2%.

For percent fat intake, the best performing model is the LASSO model using dataset 4, with a CV-$R^2$ of 10.4%. This is 0.4% higher than the highest-performing tree-based model, which is the BART model using dataset 4. Introducing the FFQ data (dataset 4) increased the CV-$R^2$ by 2%. The addition of participant characteristics and diet-related biomarkers did not increase model performance. Overall, none of the tree-based models or the linear, LASSO model achieved a CV-$R^2$ score of over 36%.

**Table 3:** Cross-Validated R2 using various models and datasets to predict total dietary fat intake.

| | *Dataset 1* | *Dataset 2* | *Dataset 3* | *Dataset 4* |
|---|---|---|---|---|
| *Pruned Tree* | 1.4% | -1.9% | -0.2% | -5.7% |
| *Random Forest* | 6.9% | 7.4% | 8.3% | 5.0% |
| *Boosted Tree* | -6.5% | -36.6% | -16.1% | -45.3% |
| *BART* | 2.0% | **10.2%** | 7.1% | 9.8% |
| *LASSO* | -7.0% | -7.2% | 7.7% | 9.7% |

Dataset 1: Metabolites only.
Dataset 2: Dataset 1 + Participant Characteristics.
Dataset 3: Dataset 2+ Diet-related biomarkers (Total Energy Intake and Urine Nitrogen).
Dataset 4: Dataset 3+ FFQ from WHI enrollment.

**Table 4:** Cross-Validated R2 using various models and datasets to predict percent dietary fat intake.

| | *Dataset 1* | *Dataset 2* | *Dataset 3* | *Dataset 4* |
|---|---|---|---|---|
| *Pruned Tree* | -1.0% | -1.8% | -1.8% | -1.8% |
| *Random Forest* | 8.8% | 9.7% | 8.8% | 9.5% |
| *Boosted Tree* | -12.0% | -16.9% | -13.4% | -38.3% |
| *BART* | -0.8% | -21.0% | -2.6% | 5.1% |
| *LASSO* | 9.5% | 9.5% | 9.5% | **10.4%** |

Dataset 1: Metabolites only.
Dataset 2: Dataset 1 + Participant Characteristics.
Dataset 3: Dataset 2+ Diet-related biomarkers (Total Energy Intake and Urine Nitrogen).
Dataset 4: Dataset 3+ FFQ from WHI enrollment.

Figure 1 and Figure 2 show dotplots of the $R^2$s calculated during the cross-validation process, predicting total fat and percentage fat, respectively. Figure 1 depicts the $R^2$ calculated with dataset 2, while figure 2 depicts the $R^2$ calculated with dataset 4 since these datasets produced the models with the highest prediction accuracy for their respective targets. Additional plots for other datasets can be found in appendix A. Figures 3 and 4 show a plot of the $R^2$ of the best models for each dataset, predicting total fat and percent fat intake, respectively. As depicted by the figures, there seems to be little difference in the prediction accuracy between models when using the same dataset, as there is a large overlap between the prediction value range of each model. The exception to this is the boosted tree model, which generally has a lower median performance score and higher variability than the other models, and the BART model trained with dataset 2, since it's median CV-$R^2$ is outside of the other model's interquartile range (figure 1).

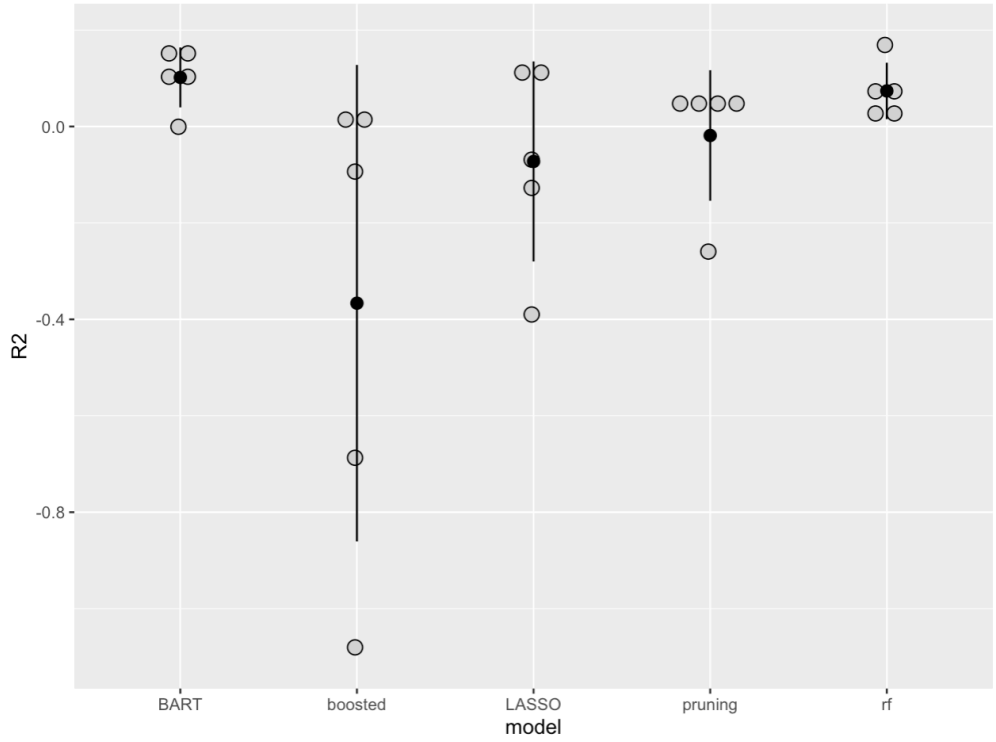



**Figure 1**: R2 results for predicting total fat intake from the 5-fold cross-validation using dataset 2. The grey points represent the individual predictions. The black points represent the mean prediction for each model. The black lines represent the standard deviation for each model (± 1 SD).

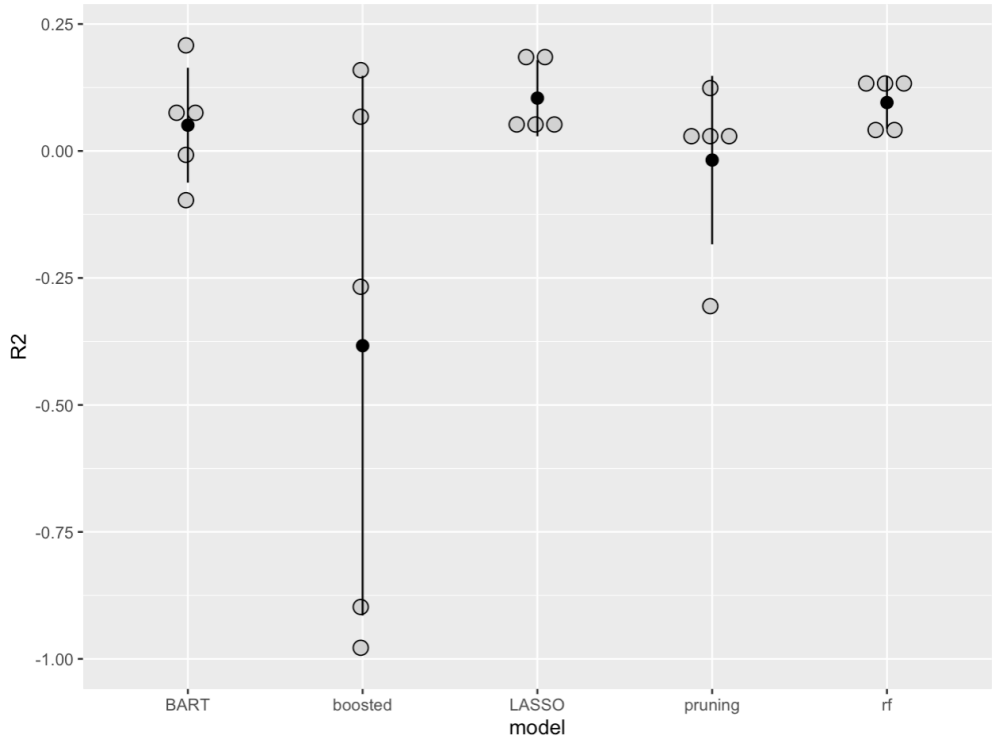



**Figure 2**: R2 results for predicting percent fat intake from the 5-fold cross-validation using dataset 4.

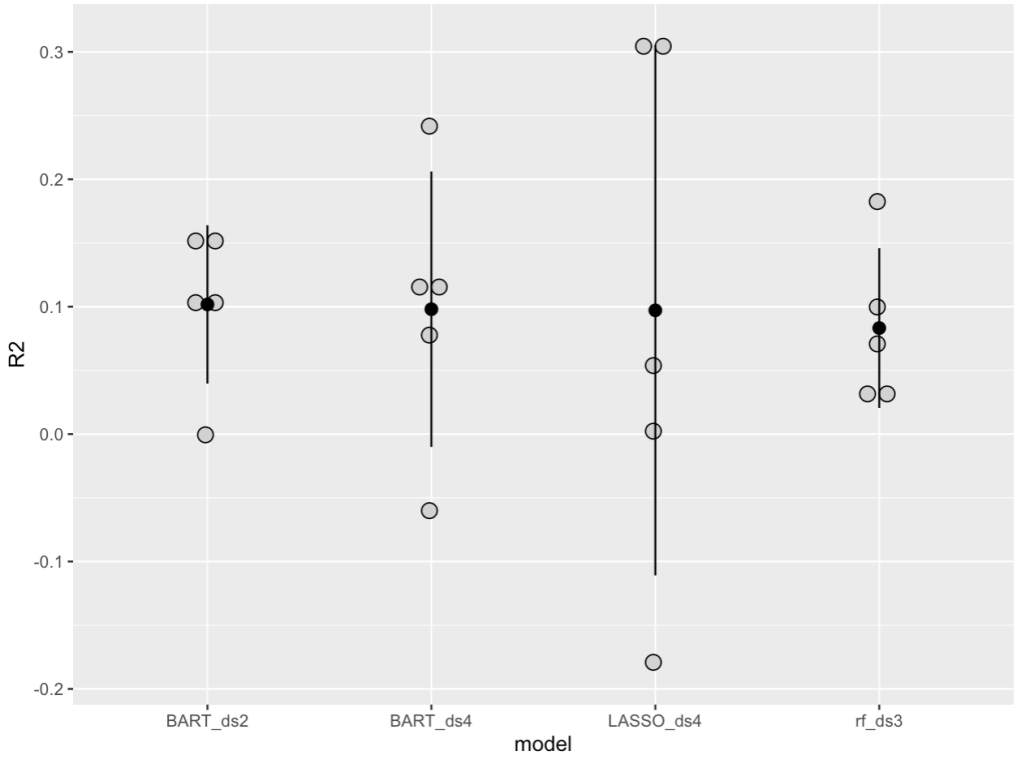



**Figure 3**: R2 results from the 5-fold cross-validation using the best models to predict total fat intake.

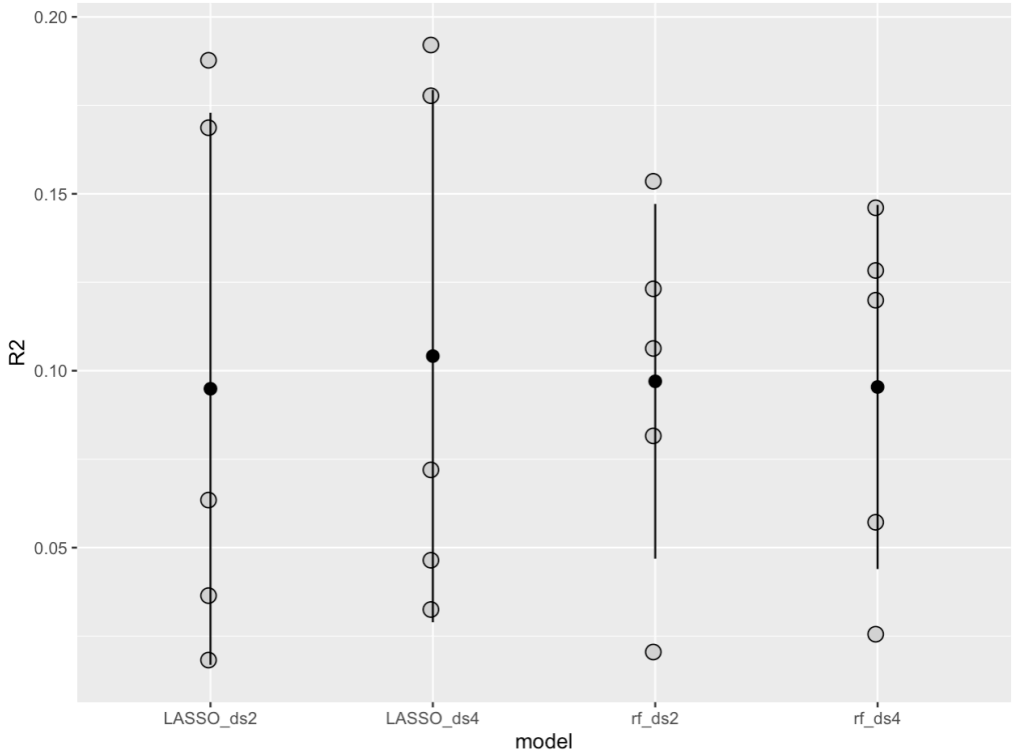



**Figure 4**: R2 results from the 5-fold cross-validation using the best models to predict percent fat intake.

Figures 5 represents the important variables used by the BART model as the best-performing tree-based model for predicting total fat intake. Overall, lipodomic samples had the greatest impact on prediction performance, accounting for 75% of the top 20 most important variables used by the model, followed by 24hr urine samples (20%) and LC-MS/MS serum samples (5%). Out of the top lipodomic samples, 60% came from lipid species composition, 27% came from lipid species concentration and 13% came from fatty acid composition. 67% of the important lipid samples were triacylglycerol samples. Other important lipid samples included diacylglycerol, lactosylceramide, phosphatidylcholine, cholesterol ester and lysophosphatidylcholine samples, which accounted for 7% each of the 20 most important variables used in the BART model. No demographic variables were used extensively in the BART model.

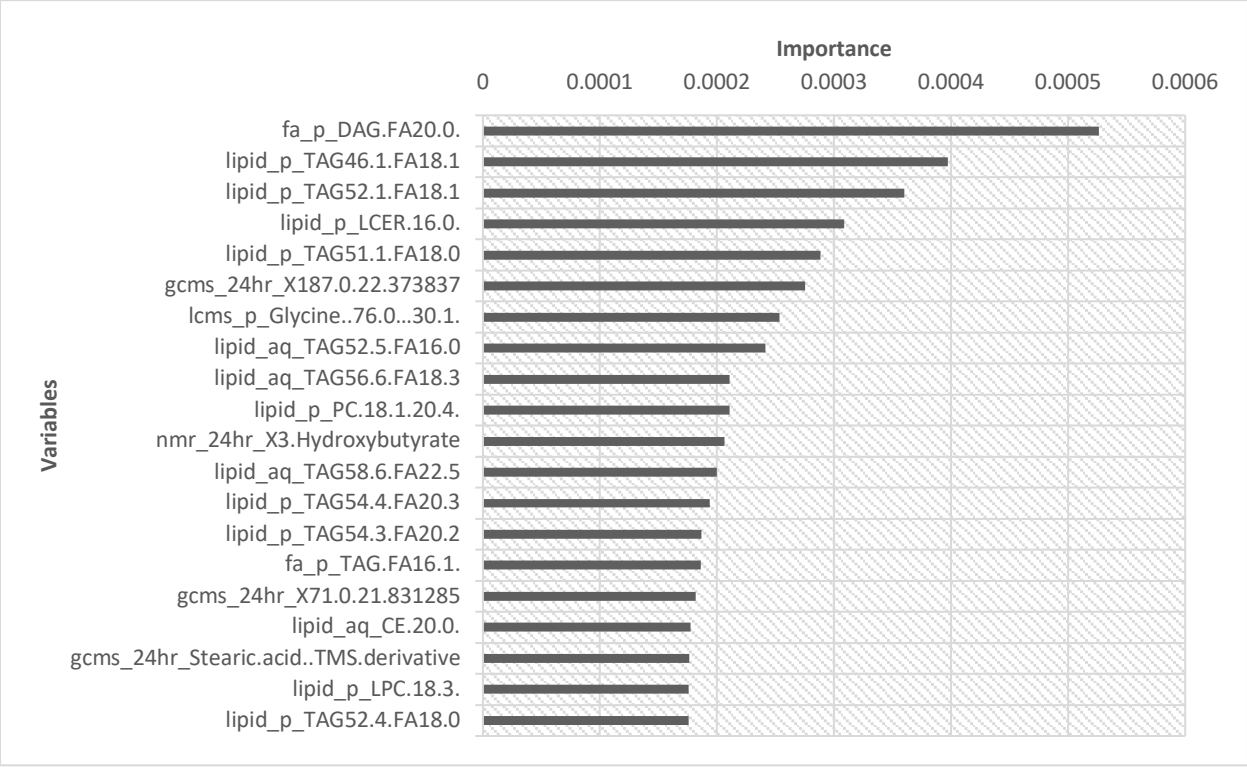



*Figure 5: Variables for selecting total dietary fat content using the BART tree model and dataset 2.*

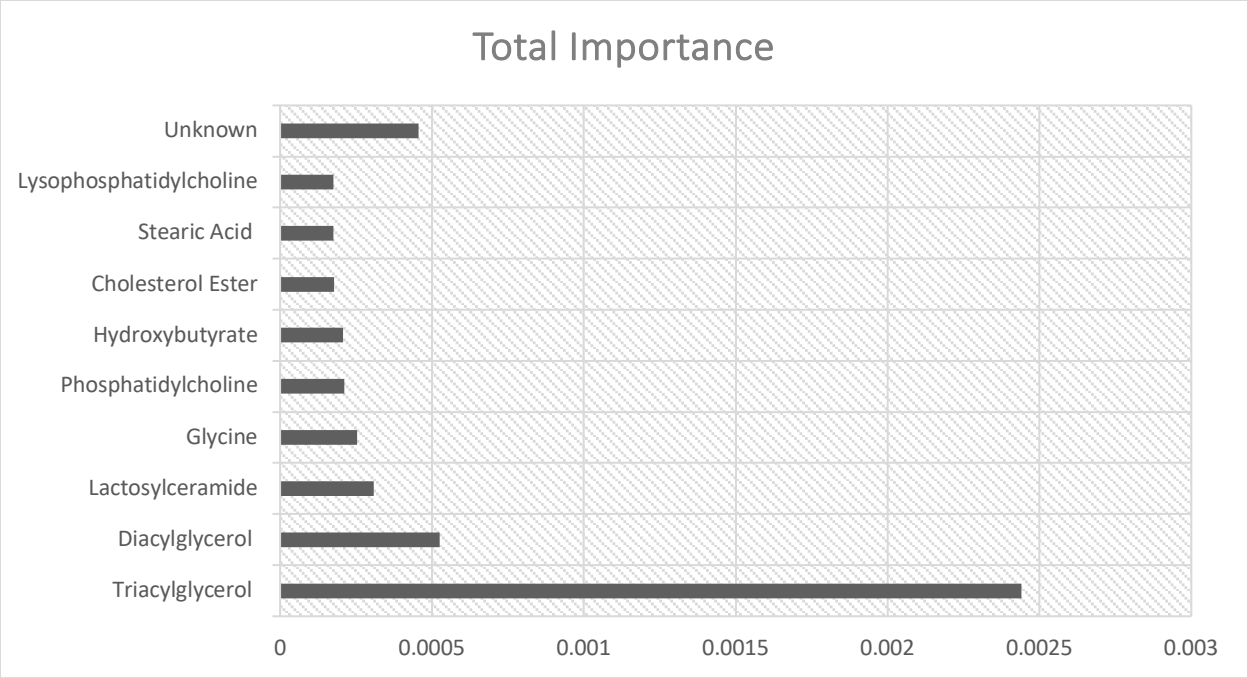



*Figure 6: Graph showing the most important metabolites in predicting total fat intake using the BART model and dataset 2. Metabolite importance was calculated by adding the variable importance of the 20 most important variables in the prediction model.*

## 4. Discussion

### 4.1. Overview of significant findings and interpretations

The goal of this study was to evaluate the predictive accuracy of non-linear, tree-based models in comparison to more traditional, linear regression models in regard to the prediction of dietary fat intake, using a cohort of 153 postmenopausal women. These results show that there is no significant difference between models, both for total fat and percentage fat intake.

Overall, the best-performing tree-based models relied more heavily on lipid serum variables, suggesting that there could be non-linear associations between serum lipid levels and fat intake that could be studied further.

The linear, LASSO model performed best in terms of predicting percent dietary fat intake, and the BART model performed best in terms of total fat predictions. The BART model relied most heavily on lipid species composition, triacylglycerol, diacylglycerol, lactosylceramide, glycine, phosphatidylcholine, hydroxybutyrate, cholesterol esters, stearic acid, and lysophosphatidylcholine levels.

Triacylglycerols, or TAGs, have been shown in some studies to be a reliable biomarker for dietary fat intake[18], as it reflects adipose tissue activity and reflects long term dietary patterns[19]. Diacylglycerols are a known intermediate in fat digestion as a result of TAG hydrolysis[20]. Lactosylcermides may also play an important role in fat metabolism as one study in mice has found that mice fed a high fat, high cholesterol diet had increased levels of lactosylceramide activity[21]. Additionally, the study also found that mice fed a western diet increased levels of oxidized LDLs as well as lactosylceramide synthase activity, which suggests that lactosylceramides may play an important role in LDL oxidation. The relationship between dietary fat intake and lactosylceramide concentration in humans is still yet to be explored. Glycine, which is found in bile acids secreted by the small intestine during lipid absorption, is an important component of fat digestion and absorption[22]. Studies have also found that people with a higher BMI and/or diabetes have lower concentrations of glycine plasma levels, suggesting that glycine levels could be related to fat accumulation in the body[22]. Phosphatidylcholine is also part of the lipid absorption process, as it is a major component of bile salts produced by the liver to break down dietary fats in the intestine[23]. It is also an essential component in the chylomicron membrane that aids in the transport of triacylglycerol throughout the body[23]. 3-hydroxybutyrate is a well-known metabolite for fatty acid oxidation in the absence of sufficient blood glucose[24].

Although 3-hydroxybutyrate is a more useful biomarker for dietary glucose, it can be an indicator for higher protein and fat intake, especially when combined with other biomarkers. The connection between thyroxine and fat consumption is less direct. However, there is some evidence that thyroxine, a thyroid hormone, has effects on hepatic fatty acid and cholesterol synthesis, which is indirectly connected to exogenous lipid metabolism[25]. Cholesterol ester composition has also been shown to be a strong biomarker for dietary fat composition[26], as well as a qualitative biomarker of fatty acid intake[27]. Stearic acid is often used as a component of plasma fatty acid used as a biomarker for dietary intake. However, there is currently little evidence to suggest that plasma fatty acid levels, including stearic acid levels, are adequate biomarkers for fat intake[28]. Finally, phosphatidylcholines (PCs) have been shown to be a strong indicator for detary carbohydrate to fat ratio in human and mouse models[29].
These findings overall support the fact that these models are using proven variables related to fat metabolism and, therefore, still hold potential for future research using non-linear models.

### 4.2. Limitations

It is important to take into account the limitations of both the dataset and the methodology used. Firstly, the metabolites used to measure fat intake can differ in their ability to indicate long-term or short-term dietary habits. In this paper, we will not distinguish between the metabolites that represent short-term consumption (those that would be the most sensitive to the 14-day feeding intervention), and those that represent more long-term dietary patterns.
The demographic characteristics of the NPAAS-FS participants will also impact the generalizability of the model. These participants are post-menopausal, predominantly white women with some higher education, who generally do not smoke or drink. A more diverse group of participants would be needed to provide more generalizable results. Additionally, the sample size is relatively small for a high-dimension, non-linear analysis, which could impact both predictive performance and accurate variable selection in the models.
The use of total fat intake as opposed to a more detailed fat profile (ex., amount of saturated fat vs. polyunsaturated fatty acids) as the outcome variable could lead to confounding since the metabolite profile in the participants is likely more closely related to the type of fats consumed, rather than the total amount of fat. Therefore, a more detailed analysis using the different types of fat as the outcome might be useful for future studies.

Finally, the nature of machine learning methods, such as the tree-based methods used in this study, are not meant to provide clear causality between the variables and the outcomes. This research is meant to provide a basis for consistent prediction of fat intake and to provide potential observations on the relationships between different metabolites and their relation to fat consumption. These observational findings would need to be further studied to prove causality.

### 4.3.Conclusions and Future Research

In conclusion, this study has shown that there is no significant difference between linear and non-linear, tree-based models when predicting total and percent fat intake. Neither the linear models, nor the non-linear models had a CV-$R^2$ score of over 36%. However, the variables used by the non-linear models are, for the most part, backed up by our current understanding of fat metabolism. Non-linear models also tend to use different variables than linear models, which could indicate that some biomarkers are better used for non-linear models than others. It is therefore recommended to further use non-linear, machine learning models in future studies to uncover connections and pathways that would not be identified using linear models alone. Using a larger dataset and more robust machine learning methods, such as deep learning and other unsupervised machine learning methods, could also yield more consistent and better results, and could help further our understanding of the link between biomarkers and dietary intake.

## Appendixes

Appendix A: Dotplots showing the R2 results

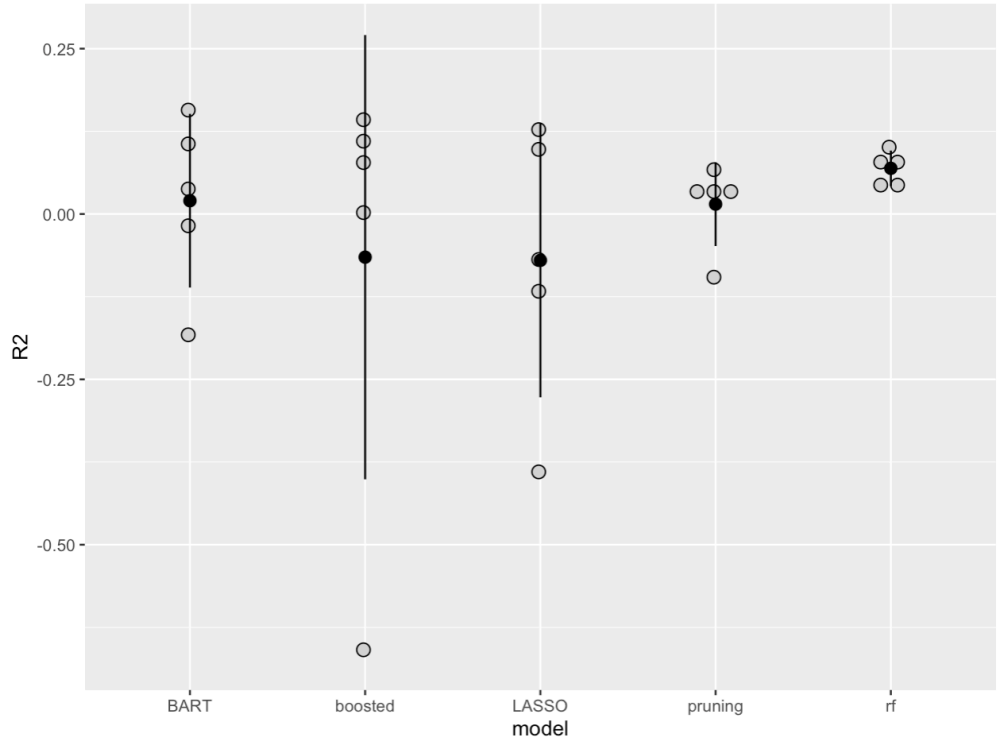



*Figure 6: R2 results for predicting total fat intake from the 5-fold cross-validation using dataset 1.*

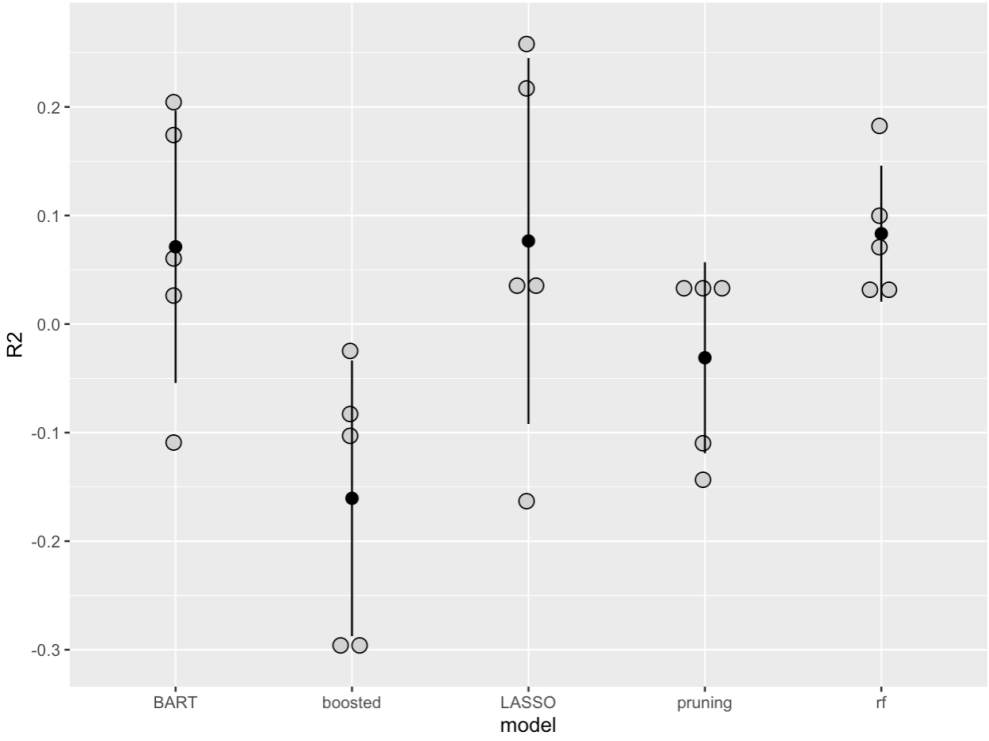



*Figure 7: R2 results for predicting total fat intake from the 5-fold cross-validation using dataset 3.*

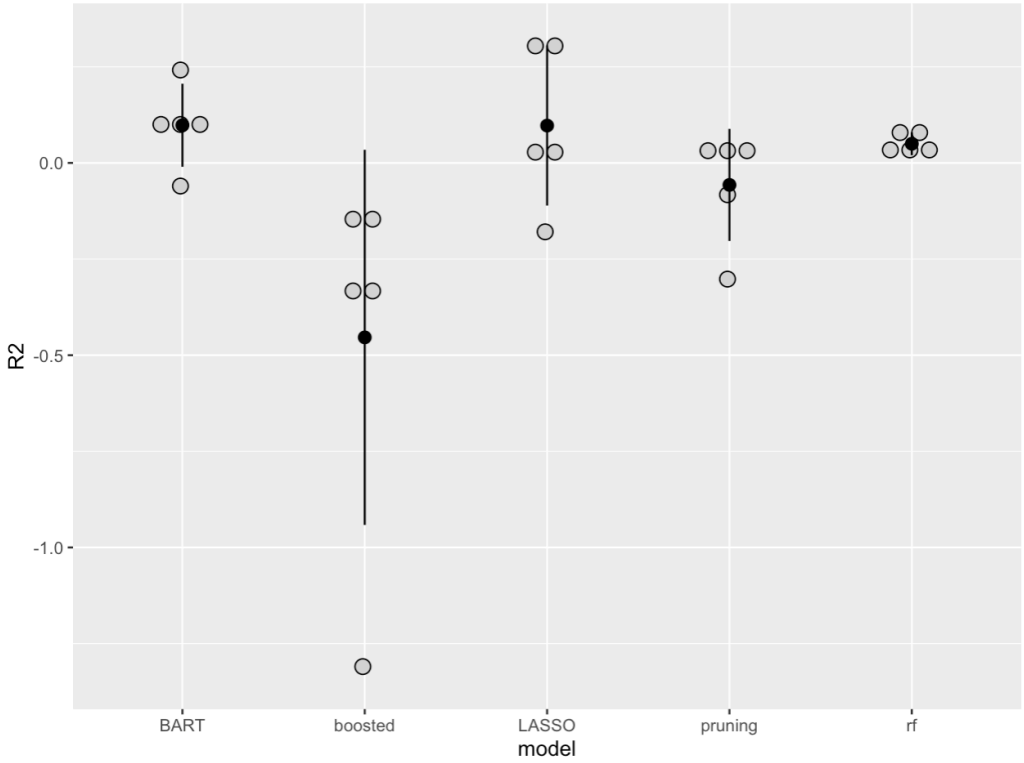



*Figure 8: R2 results for predicting total fat intake from the 5-fold cross-validation using dataset 4.*

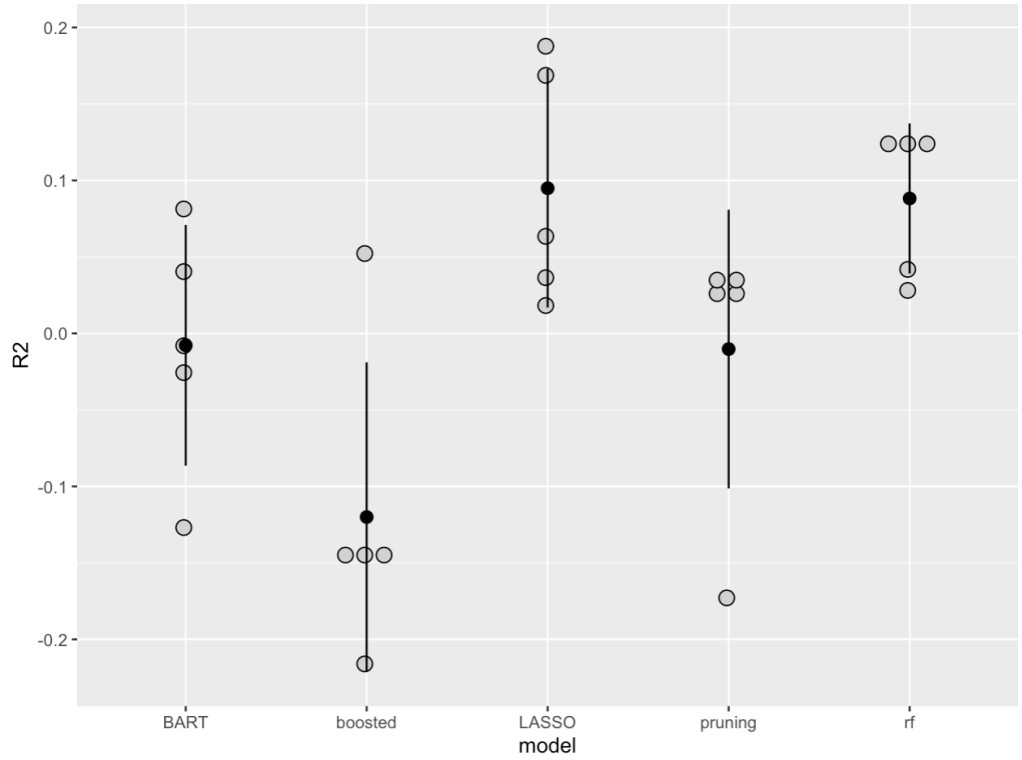



*Figure 9: R2 results for predicting percent fat intake from the 5-fold cross-validation using dataset 1.*

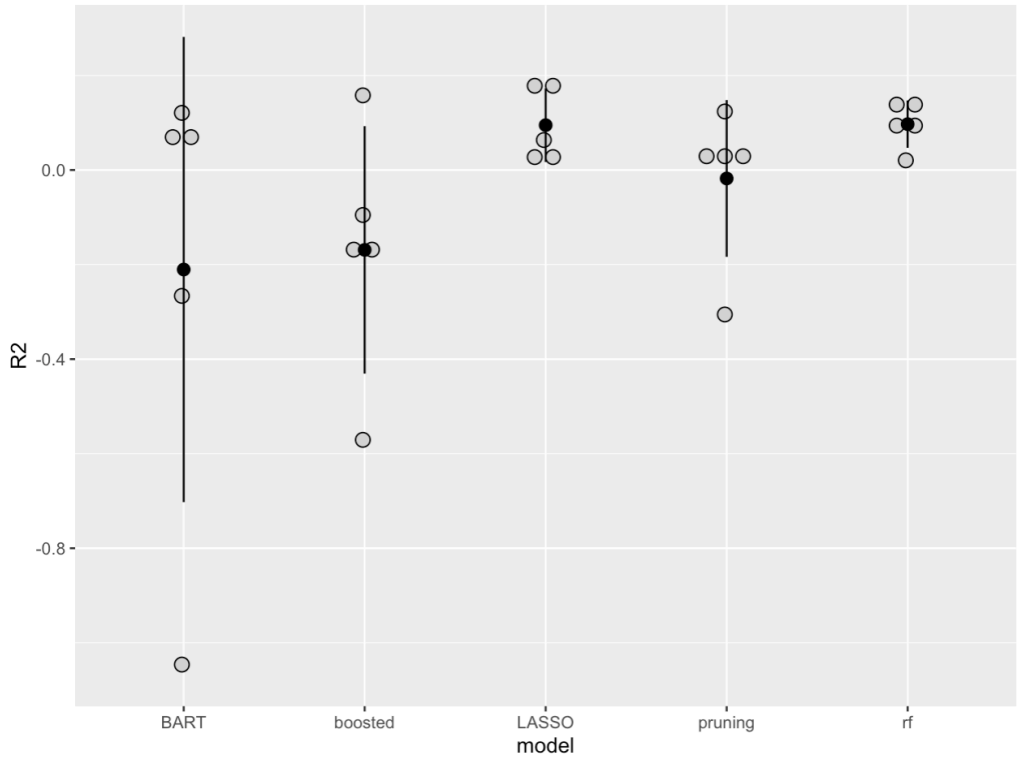



*Figure 10: R2 results for predicting percent fat intake from the 5-fold cross-validation using dataset 2.*

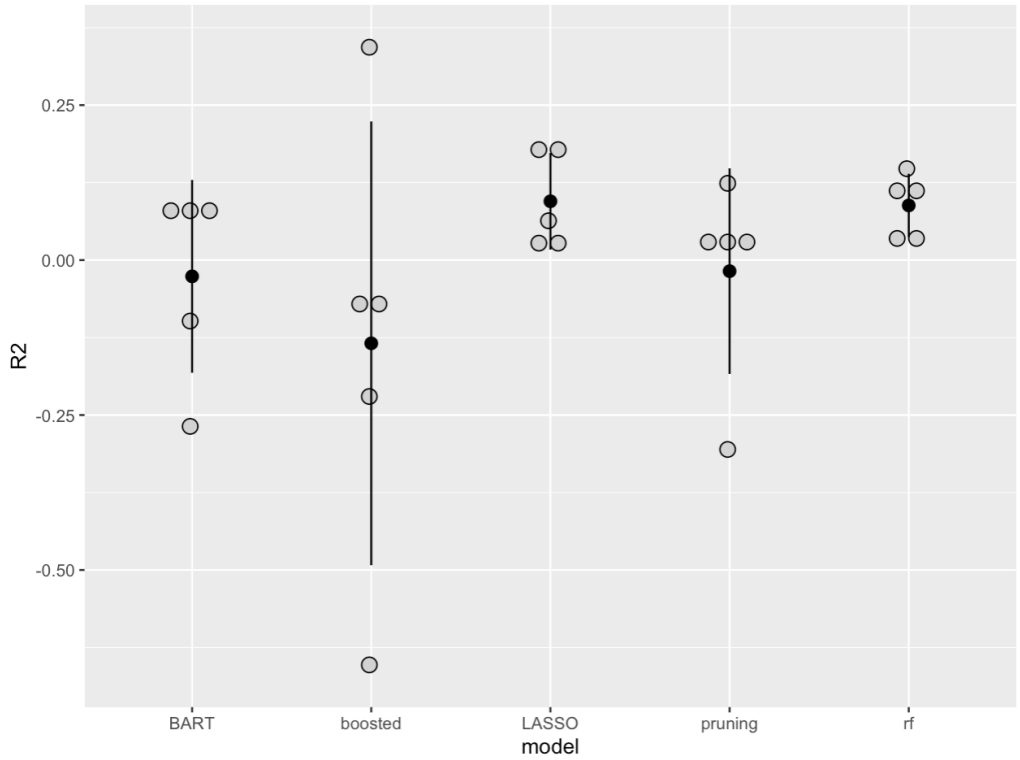
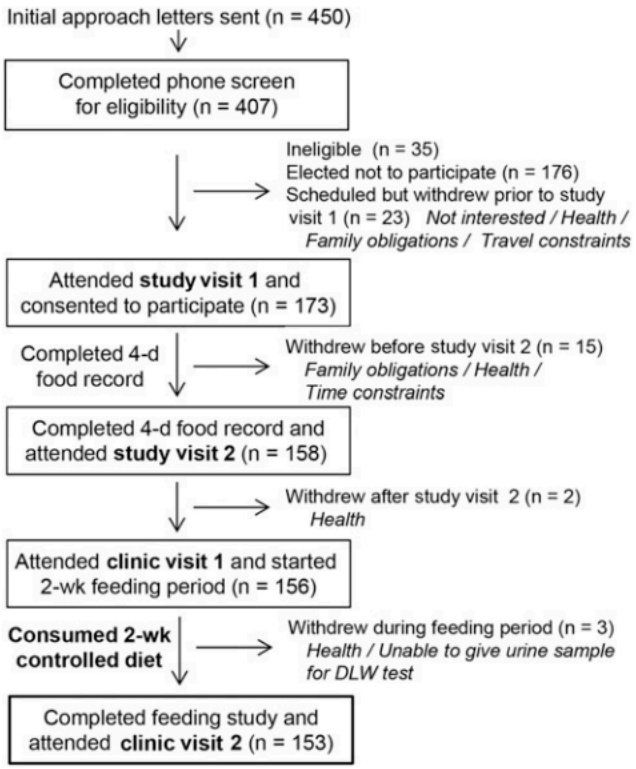



*Figure 11: R2 results for predicting percent fat intake from the 5-fold cross-validation using dataset 3.*

Appendix B: Inclusion Criteria for the NPAAS-FS study




*Source: Lampe et al. (2017)*[12]

Appendix C: Tree-based model descriptions

1. Pruned-Tree

The pruned tree model in machine learning is a model that reduces the size of the decision trees by using the Classification And Regression Tree (CART) algorithm. In this data analysis, the rpart (Recursive Partitioning And Regression Trees) package was used to implement the CART algorithm in R. The decision tree is split by splitting each subset recursively until there is the smallest possible heterogeneity of the predicted variable between the subsets. In this analysis, we used the Gini Index as a way for the algorithm to split the nodes and determine impurity, or the overall lack of homogeneity within a subset. The formula for the Gini Index is given below:

$$Gini\ Index = 1 - \sum_j p_j^2$$

Where $j$ is the class, or subset, and $p_j$ is the probability of a class j. The lower the Gini Index, the lower the likelihood of misclassification. The tree is further split based on each variable that is used in the model. It is important to note that the splits in the tree will not be changed based on splits made later on. This is referred to as the top-down, or greedy approach to partitioning the tree. Once all variables are taken under account, the final nodes of the tree, or "regions", determine the prediction for a given set of variables.

The process of pruning seen in the rpart algorithm limits the number of final nodes or regions in the decision tree, which can decrease the risk of overfitting the predictions to the training dataset[30]. The pruning approach trims down the depth of the tree in order to minimize overfitting[31]. The rpart algorithm "prunes" the tree by minimizing the cost $C_\alpha(T)$:

$$C_\alpha(T) = R(T) + \alpha|T|$$

Where R(T) is the error or variance, |T| is the number of final nodes in the tree and α is a penalty value to minimize tree nodes, which is determined during cross validation.

2. Random Forest

The second tree-based algorithm used in this analysis is a random forest method using the ranger package in R. The ranger algorithm works in three steps[32]:

1. Grow a random forest tree from the training data.
2. For each of the test observations, the weights of all training observations are computed by counting the number of trees in which both the training observation and the test observations were placed in the same terminal node.
3. For each of the test observations, grow a weighted random forest tree using the training data, and the weights established in step 2.

Similarly to the bagged model, the random forest model builds a number of decision trees using the training sample. However, instead of using all predictors, only a random sample of predictors is considered to make the split. In the case of this analysis, five subsets of predictors were used to determine the importance of each predictor. This method is favored when dealing with highly correlated predictors, as the algorithm is less likely to use only the strongest predictor for every tree[30].

3. Boosted Tree

The boosted tree method was utilized in this analysis using the XGBoost (or eXtreme Gradient Boosting) package in R. Unlike the pruned tree or random forest models, the boosted tree model does not create a decision tree by using random subsets from the training data, a method also called bootstrapping[30]. Instead, the boosted tree model uses the information from the previous tree to make a new tree, making the tree-growing process sequential as opposed to separate. In boosting, each tree is fit from a modification of the original subsample from the training set instead of using a random sample each time. The boosted model uses the residuals from the previous tree to make a new model. Here are the algorithm steps for boosting in regression trees[30]:

1. Set the number of trees $f(x) = 0$ and $r_i = y_i$ for all ith observations in the training set, where r is the residuals, or the difference between the ith observed response value and the ith response value that is predicted by the model. $y_i$ is the ith observation of the outcome.
2. For b = 1, 2, …, B, where B is the number of predetermined trees, repeat:
    a. Fit a tree $f^b$ with $d$ splits to the training data (X, $r$), where X is a ($n$ x $p$) matrix where n is the number of observations, and p is the number of variables.
    b. Update $f$ by adding in a shrunken version of the new tree:

$$f(x) \leftarrow f(x) + \lambda f^b(x_i)$$

Where $\lambda$ is the predetermined shrinkage parameter that controls the rate at which the boosting algorithm learns.

c. Update the residuals:

$$r_i \leftarrow r_i - \lambda f^b(x_i)$$

3. Output the boosted model:

$$f(x) = \sum_{b=1}^{B} \lambda f_b(x)$$

4. Bayesian Additive Regression Tree

The Bayesian Additive Regression Tree (or BART) is the fourth and final tree-based algorithm used in this analysis. Similarly to the boosted model, the BART model uses the original training data to grow its trees successively. To implement the BART model in this analysis, the bartMachine package was used in R. The BART algorithm is implemented as follows:

1. Let *K* equal the number of regression trees during the training, *B* is the number of iterations the algorithm will run, and $f_k^b(x)$ is the prediction at *x* for the *k*th regression tree in the *b*th iteration. The BART algorithm starts with all K trees having a single root node, with $f_k^1(x)$ equal to the mean of the response values divided by the total number of trees.
2. For $b = 2, \ldots, B$:
    a. For $k = 1,2, \ldots, K$:
        i. For $i = 1,\ldots, n$, the current partial residual is computed as follows:

$$r_i = y_i - \sum_{k'<k}^{K} f_{k'}^b(x_i) - \sum_{k'>k}^{K} f_{k'}^{b-1}(x_i)$$

        ii. A new tree is then fitted to $r_i$ by randomly perturbing the *k*th tree from the previous iteration, $f_{k'}^{b-1}(x_i)$. The algorithm favors perturbations that improve the fit. Perturbations can include changing the structure of the tree by adding or pruning branches, or by changing the prediction in each terminal node of the tree.
    b. $f^b(x) = \sum_{k=1}^{K} f_k^b(x)$ is computed.

3. The final tree is computed by calculating the mean after L burn-in samples:

$$f(x) = \frac{1}{B-L} \sum_{b=L+1}^{B} f^b(x)$$

Appendix D: Sample code used for the Data Analysis

```
#####data preparation#####
#Libraries
library(tidyverse)
library(dplyr)
library(haven)
library(readr)
library(faraway)
library(randomForest)
library(tibble)

##### USING THE NPAAS-FS dataset###########
#setting the directory
NPAAS_FS_dir  <- '/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/'
setwd(NPAAS_FS_dir)

Age_hght_wght_df <- read.csv('NPAAS-FS(AS272)_form
275_visit_1_common_ID_Age_BodyWeight_Height.csv')

data_var_FS <- Age_hght_wght_df

#creating a dataframe with just the IDs and the age
data_var_select_FS <- data_var_FS %>% select(WHI_COMMON_ID, AGE_AT_VISIT_1,
BMI_V1)
#renaming the age variable
data_var_select_FS <- data_var_select_FS %>%
  rename(agev1_FS = AGE_AT_VISIT_1)

#selecting the age variable
df <- data_var_select_FS
df$agev1_FS[df$agev1_FS >= 60 & df$agev1_FS <= 69] <- 0
df$agev1_FS[df$agev1_FS >= 70 & df$agev1_FS <= 79] <- 1
df$agev1_FS[df$agev1_FS >= 80 & df$agev1_FS <= 85] <- 2
unique(df$agev1_FS)
data_var_select_FS <- df

#getting the BMI variables

df <- data_var_select_FS
#BMI_v1
df$BMI_V1[df$BMI_V1 > 16 & df$BMI_V1 < 25] <- 0
df$BMI_V1[df$BMI_V1 >= 25 & df$BMI_V1 < 30] <- 1
df$BMI_V1[df$BMI_V1 >=30 & df$BMI_V1 <= 100] <- 2
```

```
unique(df$BMI_V1)

data_var_select_FS <- df

#supplement use variable
suppl_df <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/NPAAS_FS(AS272)_npfs_form_45_common_id.csv')
df <- suppl_df %>% select(commonid, anysupp)

#merging the main dataset with the supplement dataset
df <- df %>%
  rename(WHI_COMMON_ID = commonid)

#finding the IDs that are in data_var_select but not in df
data_var_select_FS %>%
  filter(!data_var_select_FS$WHI_COMMON_ID %in% df$WHI_COMMON_ID) #For df1
values not in df2

#adding the missing IDs to the df
new_row1 = c(WHI_COMMON_ID = 147896, anysupp= NA)
new_row2 = c(WHI_COMMON_ID = 194493, anysupp = NA)
new_row3 = c(WHI_COMMON_ID = 241500, anysupp = NA)
new_row4 = c(WHI_COMMON_ID = 282974, anysupp = NA)
df_bind = rbind(df,new_row1, new_row2, new_row3, new_row4)

#sorting the df by ID
df_bind[order(df_bind$WHI_COMMON_ID, decreasing=FALSE),, drop=FALSE]

#binding the suppl column to the main dataframe
df_suppl <- cbind(data_var_select_FS, df_bind$anysupp)
df_suppl <- df_suppl %>%
  rename(anysupp = `df_bind$anysupp`)
data_var_select_FS <- df_suppl

#Ethnicity variable
data_ethn <- read.csv('/Users/cnondin/Documents/THESIS/dataset/ethnicity
/ethnicity_dataset.csv')
df <- data_ethn %>% select(ID, RACENIH)

#selecting rows to only get IDs from the FS dataset
list_ID_FS = as.list(data_var_select_FS$WHI_COMMON_ID)
df <- df[df$ID %in% list_ID_FS,]
#sorting by ID
df <- df[order(df$ID, decreasing=FALSE),, drop=FALSE]
#changing 5 to 0 (white), others to 1 (other ethnicity) and 9 to NAs
unique(df$RACENIH)
```

```
df$RACENIH[df$RACENIH== 5]<-0
df$RACENIH[df$RACENIH== 6]<-1
df$RACENIH[df$RACENIH== 2]<-1
df$RACENIH[df$RACENIH== 4]<-1
df$RACENIH[df$RACENIH== 9]<-NA
unique(df$RACENIH)

#binding the main df and the new df
df_bind = cbind(data_var_select_FS,df$RACENIH)
#renaming the ethnicity column
df_bind <- df_bind %>%
  rename(ethnicity = `df$RACENIH`)

data_var_select_FS <- df_bind

#currently smoking variable
data_cigs <- read.csv('/Users/cnondin/Documents/THESIS/dataset/cigsperday/cigsperday.csv')
#selecting the SMOKNOW variable
df <- data_cigs %>% select(ID, SMOKNOW)
#selecting the IDs for the FS
df <- df[df$ID %in% list_ID_FS,]
df <- df[order(df$ID, decreasing=FALSE),, drop=FALSE]
#binding the main df and the new df
df_bind = cbind(data_var_select_FS,df$SMOKNOW)
#renaming the ethnicity column
df_bind <- df_bind %>%
  rename(cignow = `df$SMOKNOW`)

data_var_select_FS <- df_bind

#season variable
season_df <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/MS3613_NPAAFS_seasonV2.csv')

#finding missing IDs and giving them NA
data_var_select_FS %>%
  filter(!data_var_select_FS$WHI_COMMON_ID %in% season_df$commonid)
#adding the missing IDs to the df
new_row1 = c(commonid = 241500, seasonv2= NA)
new_row2 = c(commonid = 282974, seasonv2= NA)
df_bind = rbind(season_df,new_row1, new_row2)

#sorting the df by ID
df_bind[order(df_bind$commonid, decreasing=FALSE),, drop=FALSE]

df_combine = cbind(data_var_select_FS,df_bind$seasonv2)
```

```
#renaming the season column
df_combine <- df_combine %>%
  rename(season = `df_bind$seasonv2`)

#renaming the season values
df_combine$season[df_combine$season== 'Spring']<-0
df_combine$season[df_combine$season== 'Summer']<-1
df_combine$season[df_combine$season== 'Fall']<-2
df_combine$season[df_combine$season== 'Winter']<-3

data_var_select_FS <- df_combine

#years of education variable
educ_df <- read.csv('/Users/cnondin/Documents/THESIS/dataset/education/education.csv')
df <- educ_df %>% select(ID, EDUC)

#selecting rows to only get IDs from the FS dataset
list_ID_FS = as.list(data_var_select_FS$WHI_COMMON_ID)
df <- df[df$ID %in% list_ID_FS,]
#sorting by ID
df <- df[order(df$ID, decreasing=FALSE),, drop=FALSE]
#changing values
unique(df$EDUC)
df$EDUC[df$EDUC == 5] <- 0 #High school
df$EDUC[df$EDUC >= 6 & df$EDUC <= 7] <- 1 # education after high school
df$EDUC[df$EDUC >= 8 & df$EDUC <= 11] <- 2 #College degree or higher
unique(df$EDUC)

#binding the main df and the new df
df_bind = cbind(data_var_select_FS,df$EDUC)
#renaming the educ column
df_bind <- df_bind %>%
  rename(education = `df$EDUC`)

data_var_select_FS <- df_bind

#Physical activity variable
PA_df <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/NPAAS-FS(AS272)_form_35_common_ID.csv')

#removing ID 171605
PA_df <- PA_df[-60,]

#selecting the physical activity column
df <- PA_df %>% select(COMMID, TEXPWK)
```

```
#changing the variable values
unique(df$TEXPWK)
df$TEXPWK[df$TEXPWK >= 0 & df$TEXPWK<= 5.5] <- 0
df$TEXPWK[df$TEXPWK >= 5.6 & df$TEXPWK <= 12.25] <- 1
df$TEXPWK[df$TEXPWK >= 12.3 & df$TEXPWK <= 24.0] <- 2
df$TEXPWK[df$TEXPWK > 24] <- 3
unique(df$TEXPWK)

df_combine = cbind(data_var_select_FS,df$TEXPWK)
#renaming the physical activity column
df_combine <- df_combine %>%
  rename(PA_wk = `df$TEXPWK`)

data_var_select_FS <- df_combine

#Ein variable
Ein_df <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/NPAAS-FS TEE Energy.csv')
data_var_select_FS %>%
  filter(!data_var_select_FS$WHI_COMMON_ID %in% Ein_df$ID)

#adding the missing IDs to the df
new_row1 = c(ID = 241500, Ein = NA)
new_row2 = c(ID = 282974, Ein = NA)
df_bind = rbind(Ein_df,new_row1, new_row2)

#sorting the df by ID
df_bind = df_bind[order(df_bind$ID, decreasing=FALSE),, drop=FALSE]

#combining df_bind with the main df
df_combine = cbind(data_var_select_FS,df_bind$Ein)
#renaming the Ein column
df_combine <- df_combine %>%
  rename(Ein = `df_bind$Ein`)

data_var_select_FS <- df_combine

#urine nitrogen variable
Nitrogen_df <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/Total N- Herdt Lab-CommonID.csv')

#give the missing IDs a NA value
data_var_select_FS %>%
  filter(!data_var_select_FS$WHI_COMMON_ID %in% Nitrogen_df$Common.ID)

new_row1 = c(Common.ID = 187709, Total.Nitrogen..mg.mL. = NA)
```

```
new_row2 = c(Common.ID = 227839, Total.Nitrogen..mg.mL. = NA)
new_row3 = c(Common.ID = 241500, Total.Nitrogen..mg.mL. = NA)
new_row4 = c(Common.ID = 279163, Total.Nitrogen..mg.mL. = NA)
new_row5 = c(Common.ID = 282974, Total.Nitrogen..mg.mL. = NA)
df_bind = rbind(Nitrogen_df,new_row1, new_row2, new_row3, new_row4, new_row5)

#sorting by ID
df_bind = df_bind[order(df_bind$Common.ID, decreasing=FALSE),, drop=FALSE]

#binding it to the main df
df_combine = cbind(data_var_select_FS,df_bind$Total.Nitrogen..mg.mL.)

#renaming the Nitrogen column
df_combine <- df_combine %>%
  rename(Nitrogen = `df_bind$Total.Nitrogen..mg.mL.`)

data_var_select_FS <- df_combine
#making a new name for the demographic variables
demographic_df <- data_var_select_FS

#making a new dataset for the serum samples
serum_df_S1 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/serum
samples/S1_lcms_aq_study1_log-transformed.csv')
serum_df_S2 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/serum
samples/S2_lcms_p_study1_log-transformed.csv')
serum_df_S2 = subset(serum_df_S2, select = -c(ID_common)) #removing the ID from S2
serum_df = cbind(serum_df_S1, serum_df_S2)
serum_df <- serum_df[order(serum_df$ID_common, decreasing=FALSE),, drop=FALSE]

#making dataset for the lipid samples
lipid_df_L1 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/serum lipid
samples/L1_fa_aq_study1_log-transformed.csv')
lipid_df_L2 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/serum lipid
samples/L2_fa_p_study1_log-transformed.csv')
lipid_df_L2 = subset(lipid_df_L2, select = -c(ID_common)) #removing the ID from L2
lipid_df_L3 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/serum lipid
samples/L3_lipid_aq_class_study1_log-transformed.csv')
lipid_df_L3 = subset(lipid_df_L3, select = -c(ID_common))
lipid_df_L4 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS
datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/serum lipid
samples/L4_lipid_aq_study1_log-transformed.csv')
```

```
lipid_df_L4 = subset(lipid_df_L4, select = -c(ID_common))
lipid_df_L5 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/serum lipid samples/L5_lipid_p_class_study1_log-transformed.csv')
lipid_df_L5 = subset(lipid_df_L5, select = -c(ID_common))
lipid_df_L6 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/serum lipid samples/L6_lipid_p_study1_log-transformed.csv')
lipid_df_L6 = subset(lipid_df_L6, select = -c(ID_common))

lipid_df = cbind(lipid_df_L1, lipid_df_L2, lipid_df_L3, lipid_df_L4, lipid_df_L5, lipid_df_L6)
lipid_df <- lipid_df[order(lipid_df$ID_Common, decreasing=FALSE),, drop=FALSE]

#making the urine dataset
urine_df_U1 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/urine samples/24hr urine samples/U1_v2_gcms_24hr_study1_log-transformed.csv')
urine_df_U2 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS datasets/Ms4932p ASPERA Sent/metabolites/Study 1 (NPAAS-FS)/urine samples/24hr urine samples/U2_nmr_24hr_study1_log-transformed.csv')
urine_df_U2 = subset(urine_df_U2, select = -c(ID_common))

urine_df = cbind(urine_df_U1, urine_df_U2)
urine_df <- urine_df[order(urine_df$ID_common, decreasing=FALSE),, drop=FALSE]

#finding the IDs in the metabolite dfs
list_ID_met = as.list(urine_df$ID_common)

#getting rid of the two IDs that are not in the met dfs
df <- data_var_select_FS
df <- df[df$WHI_COMMON_ID %in% list_ID_met,]
demographic_df <- df
#changing some of the variables to factors
demographic_df <- mutate(demographic_df, agev1_FS = as.factor(agev1_FS))
demographic_df <- mutate(demographic_df, BMI_V1 = as.factor(BMI_V1))
demographic_df <- mutate(demographic_df, anysupp = as.factor(anysupp))
demographic_df <- mutate(demographic_df, ethnicity = as.factor(ethnicity))
demographic_df <- mutate(demographic_df, cignow = as.factor(cignow))
demographic_df <- mutate(demographic_df, season = as.factor(season))
demographic_df <- mutate(demographic_df, education = as.factor(education))
demographic_df <- mutate(demographic_df, PA_wk = as.factor(PA_wk))

#outcome df
df <- read.csv('/Users/cnondin/Documents/THESIS/dataset/PKG - Ms4932p NPAAS datasets/Ms4932p ASPERA Sent/NPAAS-FS(AS272)_NPFS_consumed_menu_avg_common_ID.csv')
```

```
#sorting the df by ID
df <- df[order(df$WHI_common_ID, decreasing=FALSE),, drop=FALSE]
#selecting variables
df <- df %>% select(WHI_common_ID, fat, percent_fat)
outcome_df <- df

###############making df for FFQ (this is for dataset 4)###################
FFQ_df <- read.csv(
"/Users/cnondin/Documents/THESIS/dataset/FFQ_baseline/FFQ_baseline.csv" )

#selecting rows to only get IDs from the FS dataset
list_ID_FS = as.list(demographic_df$WHI_COMMON_ID)
FFQ_df <- FFQ_df[FFQ_df$ID %in% list_ID_FS,]
FFQ_select <- FFQ_df %>%
  group_by(ID) %>%
  slice(1)

############ making the datasets ############
#dataset 1
#metabolomic data only: adding ID, serum df, lipid df, urine df and outcome df

lipid_df = subset(lipid_df, select = -c(ID_Common))
urine_df = subset(urine_df, select = -c(ID_common))
outcome_df = subset(outcome_df, select = -c(WHI_common_ID))
dataset_1 = cbind(serum_df, lipid_df, urine_df, outcome_df)

#dataset 2
#dataset 1 + participant characteristics

#getting rid of Ein and urine nitrogen from dem data
diet_biom_df <- demographic_df %>% select(WHI_COMMON_ID, Ein, Nitrogen)

demographic_df = subset(demographic_df, select = -c(WHI_COMMON_ID, Ein, Nitrogen))
dataset_2 <- cbind(dataset_1, demographic_df)

#changing the NA values
impute_data <- dataset_2
dataset2_imputed <- tibble::as_tibble(
  randomForest::rfImpute(fat ~ ., ntree = 200, iter = 5, data = impute_data)
) %>% select(anysupp, cignow, season, education,
             ethnicity, fat)

#removing the imputed variables from dataset 2
dataset_2 = subset(dataset_2, select = -c(anysupp, cignow, season, education, ethnicity))
#removing fat from the imputed dataset
dataset2_imputed = subset(dataset2_imputed, select = -c(fat))
```

```
#combining the two
dataset_2 = cbind(dataset_2, dataset2_imputed)

#dataset 3
#dataset 2 + diet-related biomarkers

diet_biom_df = subset(diet_biom_df, select = -c(WHI_COMMON_ID))
dataset_3 <- cbind(dataset_2, diet_biom_df)

#changing the NA values
impute_data <- dataset_3
dataset3_imputed <- tibble::as_tibble(
  randomForest::rfImpute(fat ~ ., ntree = 200, iter = 5, data = impute_data)
) %>% select(Ein, Nitrogen, fat)

#removing the imputed variables from dataset 2
dataset_3 = subset(dataset_3, select = -c(Ein, Nitrogen))
#removing fat from the imputed dataset
dataset3_imputed = subset(dataset3_imputed, select = -c(fat))
#combining the two
dataset_3 = cbind(dataset_3, dataset3_imputed)

#putting the outcomes as the last columns in dataset 2 and 3
outcome <- dataset_2 %>% select(fat, percent_fat)
dataset_2 = subset(dataset_2, select = -c(fat, percent_fat))
dataset_2 <- cbind(dataset_2, outcome)

outcome <- dataset_3 %>% select(fat, percent_fat)
dataset_3 = subset(dataset_3, select = -c(fat, percent_fat))
dataset_3 <- cbind(dataset_3, outcome)

#dataset 4
#dataset 3 + FFQ
dataset_4 <- cbind(dataset_3, FFQ_select)
outcome <- dataset_4 %>% select(fat, percent_fat)
dataset_4 = subset(dataset_4, select = -c(fat, percent_fat))
dataset_4 <- cbind(dataset_4, outcome)
#exporting the datasets
setwd("~/Documents/THESIS/dataset/datasets for analysis ")

write.csv(dataset_1, 'dataset_1.csv')
write.csv(dataset_2, 'dataset_2.csv')
write.csv(dataset_3, 'dataset_3.csv')
write.csv(dataset_4, 'dataset_4.csv')

#exporting the dempgraphic df
```

```
write.csv(demographic_df,
'/Users/cnondin/Documents/THESIS/visuals/table_1/demographic_df.csv')
write.csv(diet_biom_df, '/Users/cnondin/Documents/THESIS/visuals/table_1/diet_biom_df.csv')
write.csv(outcome, '/Users/cnondin/Documents/THESIS/visuals/table_1/outcome_df.csv')

######## sample code using CV to predict fat content #######
### (in this example, to predict total fat content using dataset 1) ####
library(caret)
library(rpart.plot)
library(tidyverse)
library(bartMachine)
library(glmnet)

#opening the datasets
dataset_1 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/datasets for analysis
/dataset_1.csv')
dataset_2 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/datasets for analysis
/dataset_2.csv')
dataset_3 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/datasets for analysis
/dataset_3.csv')
dataset_4 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/datasets for analysis
/dataset_4.csv')

#removing the percent fat column
dataset <- subset(dataset_1, select = -percent_fat)

set.seed(1)

# Train Test Split
split_index <- createDataPartition(dataset$fat, p = .8,
                                   list = FALSE,
                                   times = 5)
#split_index_mat <- matrix(unlist(split_index), ncol = 1)
num_iterations <- ncol(split_index)
MSE_df <- data.frame(matrix(NA, nrow = num_iterations, ncol = 6))
colnames(MSE_df) <- c('pruning', 'rf', 'boosted', 'BART', 'LASSO', 'fold')
fold_numbers <- 1:num_iterations
var_df <- data.frame(matrix(NA, nrow = num_iterations, ncol = 2))
colnames(var_df) <- c('test var', 'fold')

#grid for the BART model
bartGrid <- expand.grid(num_trees = c(10, 15, 20, 100), k = 2, alpha = 0.95, beta = 2, nu = 3)

for (i in 1:num_iterations) {
  # use ith column of split_index to create feature and target training/test sets
  train <- dataset[ split_index[,i], ]
```

```
test  <- dataset[-split_index[,i], ]
features_train <- dataset[ split_index[,i], !(names(dataset) %in% c('fat'))]
features_test  <- dataset[-split_index[,i], !(names(dataset) %in% c('fat'))]
target_train <- dataset[ split_index[,i], "fat"]
target_test <- dataset[-split_index[,i], "fat"]

#prunned tree
ctrl<- trainControl(method="cv", number=5)

d.tree <- train(fat ~.,
                data =train,
                trControl = ctrl,
                metric = "Rsquared",
                method = "rpart")

#random forest
r.forest <- train(fat ~.,
                  data = train,
                  trControl = ctrl,
                  metric = "Rsquared",
                  method = "ranger")

#boosted tree
xg.boost <- train(fat ~.,
                  data = train,
                  trControl = ctrl,
                  metric = "Rsquared",
                  verbose =TRUE,
                  method = "xgbTree")

#Bayesian Additive Regression Tree
BART <- train(fat ~.,
              train,
              trControl = ctrl,
              metric = "Rsquared",
              method = "bartMachine",
             preProc = c("center", "scale"),
             tuneGrid = bartGrid,
             num_burn_in = 2000,
             num_iterations_after_burn_in = 2000,
             serialize = T)

#LASSO
lasso<-train(y= target_train,
             x = features_train,
             trControl = ctrl,
```

```
            metric = "Rsquared",
            method = 'glmnet',
            tuneGrid = expand.grid(alpha = 1, lambda = 1)

  )

  #predicting for test

  # Decision Tree
  pruned_pred = predict(d.tree, test)
  # Random Forest
  rf_pred = predict(r.forest, test)
  # XGBoost
  boost_pred = predict(xg.boost, test)
  #BART
  bart_pred = predict(BART, test)
  #LASSO
  lasso_pred = predict(lasso, test)

  # Print R squared scores
  prune_MSE = mean((pruned_pred - target_test)^2)
  rf_MSE = mean((rf_pred - target_test)^2)
  boost_MSE = mean((boost_pred - target_test)^2)
  bart_MSE = mean((bart_pred - target_test)^2)
  lasso_MSE = mean((lasso_pred - target_test)^2)

  MSE_df[i,'pruning'] <- prune_MSE
  MSE_df[i,'rf'] <- rf_MSE
  MSE_df[i,'boosted'] <- boost_MSE
  MSE_df[i,'BART'] <- bart_MSE
  MSE_df[i,'LASSO'] <- lasso_MSE
  MSE_df[i,'fold'] <- i

  #saving the variance of the test set (to calculate R2 later)
  var_df[i,'test var'] <- var(target_test)
  var_df[i, 'fold'] <- i

  print(paste(i, "th fold is finished"))
}

#exporting the datasets
setwd("~/Users/cnondin/Documents/THESIS/CV_results/fat_ds1")

write.csv(MSE_df, '/Users/cnondin/Documents/THESIS/CV_results/fat_ds1/MSE_fat_ds1.csv')
write.csv(var_df, '/Users/cnondin/Documents/THESIS/CV_results/fat_ds1/var_fat_ds1.csv')
```

```
######## variable importance for BART dataset 2############
library(caret)
library(rpart.plot)
library(tidyverse)
library(bartMachine)
library(glmnet)

#opening the datasets
dataset_2 <- read.csv('/Users/cnondin/Documents/THESIS/dataset/datasets for analysis
/dataset_2.csv')

#removing the percent fat column
dataset <- subset(dataset_2, select = -percent_fat)

set.seed(1)

# Train Test Split
split_index <- createDataPartition(dataset$fat, p = .8,
                                   list = FALSE,
                                   times = 5)
#split_index_mat <- matrix(unlist(split_index), ncol = 1)
num_iterations <- ncol(split_index)
MSE_df <- data.frame(matrix(NA, nrow = num_iterations, ncol = 1))
colnames(MSE_df) <- c('BART') #remember to add BART if you figure out how it works
fold_numbers <- 1:num_iterations
var_df <- data.frame(matrix(NA, nrow = num_iterations, ncol = 2))
colnames(var_df) <- c('test var', 'fold')

#grid for the BART model
bartGrid <- expand.grid(num_trees = c(10, 15, 20, 100), k = 2, alpha = 0.95, beta = 2, nu = 3)

for (i in 1:num_iterations) {
  # use ith column of split_index to create feature and target training/test sets
  train <- dataset[ split_index[,i], ]
  test  <- dataset[-split_index[,i], ]
  features_train <- dataset[ split_index[,i], !(names(dataset) %in% c('fat'))]
  features_test  <- dataset[-split_index[,i], !(names(dataset) %in% c('fat'))]
  target_train <- dataset[ split_index[,i], "fat"]
  target_test <- dataset[-split_index[,i], "fat"]
```

```
  #Bayesian Additive Regression Tree
  ctrl<- trainControl(method="cv", number=5)
  BART <- train(fat ~.,
                train,
                trControl = ctrl,
                metric = "Rsquared",
                method = "bartMachine",
                preProc = c("center", "scale"),
                tuneGrid = bartGrid,
                num_burn_in = 2000,
                num_iterations_after_burn_in = 2000,
                serialize = T)


  #predicting for test

  #BART
  bart_pred = predict(BART, test)


  # Print R squared scores

  bart_MSE = mean((bart_pred - target_test)^2)


  MSE_df[i,'BART'] <- bart_MSE


  #saving the variance of the test set (to calculate R2 later)
  var_df[i,'test var'] <- var(target_test)
  var_df[i, 'fold'] <- i

  print(paste(i, "th fold is finished"))
}

#saving the model
saveRDS(BART, file = '/Users/cnondin/Documents/THESIS/R code
/ML_models/fat_ds2_BART.rda')

BART <- readRDS('/Users/cnondin/Documents/THESIS/R code
/ML_models/fat_ds2_BART.rda')

BARTImp <- varImp(BART, scale = FALSE)
BARTImp
```

```
##### Dotplot Visuals #####
library(ggplot2)

fat_ds1 <- read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/fat_ds1.csv')
fat_ds2 <- read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/fat_ds2.csv')
fat_ds3 <- read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/fat_ds3.csv')
fat_ds4 <- read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/fat_ds4.csv')
prcnt_fat_ds1 <-
read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/prcnt_fat_ds1.csv')
prcnt_fat_ds2 <-
read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/prcnt_fat_ds2.csv')
prcnt_fat_ds3 <-
read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/prcnt_fat_ds3.csv')
prcnt_fat_ds4 <-
read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/prcnt_fat_ds4.csv')

fat_ds1_plot=ggplot(data= fat_ds1, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
fat_ds1_plot

fat_ds2_plot=ggplot(data= fat_ds2, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
fat_ds2_plot

fat_ds3_plot=ggplot(data= fat_ds3, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
fat_ds3_plot

fat_ds4_plot=ggplot(data= fat_ds4, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
fat_ds4_plot

prcnt_fat_ds1_plot=ggplot(data= prcnt_fat_ds1, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
prcnt_fat_ds1_plot
```

```
prcnt_fat_ds2_plot=ggplot(data= prcnt_fat_ds2, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
prcnt_fat_ds2_plot

prcnt_fat_ds3_plot=ggplot(data= prcnt_fat_ds3, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
prcnt_fat_ds3_plot

prcnt_fat_ds4_plot=ggplot(data= prcnt_fat_ds4, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
prcnt_fat_ds4_plot

#making a plot with the best ones

best_fat <-
read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/best_models/best_fat.csv')

best_fat_plot=ggplot(data= best_fat, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
best_fat_plot

best_prcnt_fat <-
read.csv('/Users/cnondin/Documents/THESIS/CV_results/boxplot_R2/best_models/best_prcnt_f
at.csv')

best_prcnt_fat_plot=ggplot(data= best_prcnt_fat, mapping=aes(x=model, y=R2))+
  geom_dotplot(binaxis= 'y', stackdir = 'center', dotsize =0.75, stackratio = 1.25, fill =
"lightgray") +
  stat_summary(fun.data = "mean_sdl", fun.args = list(mult=1))
best_prcnt_fat_plot
```

```
##### Table 1 #####

library(table1)
demographic_df <-
read.csv('/Users/cnondin/Documents/THESIS/visuals/table_1/demographic_df.csv')
diet_biom_df <-
read.csv('/Users/cnondin/Documents/THESIS/visuals/table_1/diet_biom_df.csv')
outcome_df <- read.csv('/Users/cnondin/Documents/THESIS/visuals/table_1/outcome_df.csv')

table1_df <- cbind(demographic_df, diet_biom_df, outcome_df)

table1_df$agev1_FS <-
  factor(table1_df$agev1_FS, levels=c(0,1,2),
         labels=c("60-69",
                  "70-79",
                  "80-85"))
table1_df$BMI_V1 <-
  factor(table1_df$BMI_V1, levels=c(0,1,2),
         labels=c("Normal (<25.0)",
                  "Overweight (25-30)",
                  "Obese (>30)"))

table1_df$anysupp <-
  factor(table1_df$anysupp, levels=c(0,1),
         labels=c("No",
                  "Yes"))

table1_df$ethnicity <-
  factor(table1_df$ethnicity, levels=c(0,1),
         labels=c("Caucasian",
                  "Non-Caucasian"))

table1_df$cignow <-
  factor(table1_df$cignow, levels=c(0,1),
         labels=c("No",
                  "Yes"))

table1_df$season <-
  factor(table1_df$season, levels=c(0,1,2,3),
         labels=c("Spring",
                  "Summer",
                  "Fall",
                  "Winter"))

table1_df$education <-
```

```
  factor(table1_df$education, levels=c(0,1,2),
        labels=c("High school/General Educational Development diploma",
                 "Schooling after high school",
                 "College degree or higher"))
table1_df$PA_wk <-
  factor(table1_df$PA_wk, levels=c(0,1,2,3),
        labels=c("0–5.5",
                 "5.6–12.25",
                 "12.3–24.0",
                 "> 24"))

label(table1_df$agev1_FS)      <- "Age"
label(table1_df$BMI_V1)       <- "BMI"
label(table1_df$anysupp)      <- "Use of any Dietary Supplements"
label(table1_df$ethnicity)      <- "Race/Ethnicity"
label(table1_df$cignow)       <- "Currently Smoking"
label(table1_df$season)       <- "Season of study participation"
label(table1_df$education)      <- "Years of Education"
label(table1_df$PA_wk)       <- "Recreational physical activity"
label(table1_df$Ein)      <- "Total Energy Intake"
label(table1_df$fat)      <- "Fat"
label(table1_df$percent_fat)  <- "Fat"


units(table1_df$agev1_FS) <- "years"
units(table1_df$BMI_V1)    <- "kg/m2"
units(table1_df$PA_wk)      <- "MET/week"
units(table1_df$Ein)      <- "kcal/d"
units(table1_df$fat)      <- "g/d"
units(table1_df$percent_fat) <- "%E"

     table1(~ agev1_FS + ethnicity + education + BMI_V1 + anysupp + cignow + season +
     PA_wk + Ein + fat + percent_fat, data=table1_df
```

References


1. Naska A, Lagiou A, Lagiou P. Dietary assessment methods in epidemiological research: current state of the art and future prospects. *F1000Research*. 2017;6:926. doi:10.12688/f1000research.10703.1

2. Neuhouser ML, Pettinger M, Lampe JW, et al. Novel Application of Nutritional Biomarkers From a Controlled Feeding Study and an Observational Study to Characterization of Dietary Patterns in Postmenopausal Women. *Am J Epidemiol*. 2021;190(11):2461-2473. doi:10.1093/aje/kwab171

3. Zheng C, Gowda GAN, Raftery D, et al. Evaluation of potential metabolomic-based biomarkers of protein, carbohydrate and fat intakes using a controlled feeding study. *Eur J Nutr*. 2021;60(8):4207-4218. doi:10.1007/s00394-021-02577-1

4. Prentice RL, Pettinger M, Zheng C, et al. Biomarkers for Components of Dietary Protein and Carbohydrate with Application to Chronic Disease Risk in Postmenopausal Women. *J Nutr*. 2022;152(4):1107-1117. doi:10.1093/jn/nxac004

5. Prentice RL, Pettinger M, Neuhouser ML, et al. Biomarker-Calibrated Macronutrient Intake and Chronic Disease Risk among Postmenopausal Women. *J Nutr*. 2021;151(8):2330-2341. doi:10.1093/jn/nxab091

6. González-Peña D, Brennan L. Recent Advances in the Application of Metabolomics for Nutrition and Health. *Annu Rev Food Sci Technol*. 2019;10(1):479-519. doi:10.1146/annurev-food-032818-121715

7. Rafiq T, Azab SM, Teo KK, et al. Nutritional Metabolomics and the Classification of Dietary Biomarker Candidates: A Critical Review. *Adv Nutr*. 2021;12(6):2333-2357. doi:10.1093/advances/nmab054

8. Kirk D, Kok E, Tufano M, Tekinerdogan B, Feskens EJM, Camps G. Machine Learning in Nutrition Research. *Adv Nutr*. 2022;13(6):2573-2589. doi:10.1093/advances/nmac103

9. Russo S, Bonassi S. Prospects and Pitfalls of Machine Learning in Nutritional Epidemiology. *Nutrients*. 2022;14(9):1705. doi:10.3390/nu14091705

10. Huang L, Wang L, Hu X, et al. Machine learning of serum metabolic patterns encodes early-stage lung adenocarcinoma. *Nat Commun*. 2020;11(1):3556. doi:10.1038/s41467-020-17347-6

11. Louca P, Tran TQB, Toit C du, et al. Machine learning integration of multimodal data identifies key features of blood pressure regulation. *eBioMedicine*. 2022;84:104243. doi:10.1016/j.ebiom.2022.104243

12. Lampe JW, Huang Y, Neuhouser ML, et al. Dietary biomarker evaluation in a controlled feeding study in women from the Women's Health Initiative cohort. *Am J Clin Nutr*. 2017;105(2):466-475. doi:10.3945/ajcn.116.144840

13. RPubs - How to Replace Missing Data Using a Random Forest. Accessed July 27, 2023. https://rpubs.com/david-deming-tung/missing_data

14. Kuhn M. *The Caret Package*. Accessed July 27, 2023. https://topepo.github.io/caret/

15. predict function - RDocumentation. Accessed July 27, 2023. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/predict

16. varImp function - RDocumentation. Accessed July 27, 2023. https://www.rdocumentation.org/packages/caret/versions/6.0-92/topics/varImp

17. Kapelner A, Bleich J. **bartMachine** : Machine Learning with Bayesian Additive Regression Trees. *J Stat Softw*. 2016;70(4). doi:10.18637/jss.v070.i04

18. Lee SA, Wen W, Xiang YB, et al. Stability and reliability of plasma level of lipid biomarkers and their correlation with dietary fat intake. *Dis Markers*. 2008;24(2):73-79.

19. Furtado JD, Beqari J, Campos H. Comparison of the Utility of Total Plasma Fatty Acids Versus those in Cholesteryl Ester, Phospholipid, and Triglyceride as Biomarkers of Fatty Acid Intake. *Nutrients*. 2019;11(9):2081. doi:10.3390/nu11092081

20. Eichmann TO, Lass A. DAG tales: the multiple faces of diacylglycerol—stereochemistry, metabolism, and signaling. *Cell Mol Life Sci*. 2015;72:3931-3952. doi:10.1007/s00018-015-1982-3

21. Chatterjee S, Bedja D, Mishra S, et al. Inhibition of glycosphingolipid synthesis ameliorates atherosclerosis and arterial stiffness in apolipoprotein E-/- mice and rabbits fed a high-fat and -cholesterol diet. *Circulation*. 2014;129(23):2403-2413. doi:10.1161/CIRCULATIONAHA.113.007559

22. Wang W, Wu Z, Dai Z, Yang Y, Wang J, Wu G. Glycine metabolism in animals and humans: implications for nutrition and health. *Amino Acids*. 2013;45(3):463-477. doi:10.1007/s00726-013-1493-1

23. Iqbal J, Hussain MM. Intestinal lipid absorption. *Am J Physiol - Endocrinol Metab*. 2009;296(6):E1183-E1194. doi:10.1152/ajpendo.90899.2008

24. Mierziak J, Burgberger M, Wojtasik W. 3-Hydroxybutyrate as a Metabolite and a Signal Molecule Regulating Processes of Living Organisms. *Biomolecules*. 2021;11(3):402. doi:10.3390/biom11030402

25. Sinha RA, Singh BK, Yen PM. Direct effects of thyroid hormones on hepatic lipid metabolism. *Nat Rev Endocrinol*. 2018;14(5):259-269. doi:10.1038/nrendo.2018.10

26. Hodson L, Skeaff CM, Fielding BA. Fatty acid composition of adipose tissue and blood in humans and its use as a biomarker of dietary intake. *Prog Lipid Res*. 2008;47(5):348-380. doi:10.1016/j.plipres.2008.03.003

27. Zock PL, Mensink RP, Harryvan J, De Vries JHM, Katan MB. Fatty Acids in Serum Cholesteryl Esters as Quantitative Biomarkers of Dietary Intake in Humans. *Am J Epidemiol*. 1997;145(12):1114-1122. doi:10.1093/oxfordjournals.aje.a009074

28. Marchioni DM, De Oliveira MF, Carioca AAF, et al. Plasma fatty acids: Biomarkers of dietary intake? *Nutrition*. 2019;59:77-82. doi:10.1016/j.nut.2018.08.008

29. Inoue M, Senoo N, Sato T, et al. Effects of the dietary carbohydrate–fat ratio on plasma phosphatidylcholine profiles in human and mouse. *J Nutr Biochem*. 2017;50:83-94. doi:10.1016/j.jnutbio.2017.08.018

30. Gareth James, Trevor Hastie, Robert Tibshirani, Daniela Witten. *An Introduction to Statistical Learning with Applications in R Second Edition*.

31. A gentle introduction to decision trees using R. Eight to Late. Published February 16, 2016. Accessed October 25, 2023. https://eight2late.wordpress.com/2016/02/16/a-gentle-introduction-to-decision-trees-using-r/

32. Xu R, Nettleton D, Nordman DJ. Case-Specific Random Forests. *J Comput Graph Stat*. 2016;25(1):49-65. doi:10.1080/10618600.2014.983641