Predictive Modeling of Blood Pressure Categories: Integrating Demographic and Dietary Factors for Personalized Management

A Preprint

Alexander Hawthorne
University of Michigan
Ann Arbor, MI
hawthoal@umich.edu

Evelyn Paskhaver
University of Michigan
Ann Arbor, MI
evelynpa@umich.edu

Jeanne Yang
University of Michigan
Ann Arbor, MI
jeayang@umich.edu

Li Yuan
University of Michigan
Ann Arbor, MI
leeyuan@umich.edu

December 3, 2023

1 Introduction

Hypertension, a prevalent and frequently asymptomatic health condition, persists as a global health concern, significantly contributing to the burden of cardiovascular diseases (Forouzanfar et al. 2016). In response to this, our study focuses on constructing predictive models to predict blood pressure levels and aims to identify influential predictors associated with elevated blood pressure and hypertension in order to answer two questions.

Firstly, can we predict blood pressure level categories through the use of easily accessible and non-invasive data such as a patient's demographic and dietary information? While methods to find blood pressure already exist, they require the assitance of medical personnel and specialized equimpent. Thus, our study seeks to explore an alternate path, to bridge the gap between accessibility and accuracy in blood pressure assessment. As a baseline, we employ logistic regression with lasso regularization and 10-fold cross-validation, which has advantages in interpretability. Further, we leverage the extreme gradient boosting (XGB) algorithm to surpass this baseline.

We also aim to identify an answer to the following question: what are the most influential predictors associated with blood pressure level? Understanding these factors may lead to potential strategies, such as such as dietary modifications, to to manage and mitigate hypertension. In order to deduce how influential each feature is, we utilize am importance matrix, as well as a comparison of the features selected by the two algorithms.

2 Data

2.1 Overview

NHANES employs a complex, multistage, probability sampling design to select a sample representative of the civilian, non-institutionalized US population (CDC 2023b). In 2017-2018, 16,211 persons were selected from 30 survey locations, with 9,254 completing interviews and 8,704 undergoing examinations. Each participant has a unique identification number `SEQN`. According to the CDC (2023a), NHANES field operations were suspended in March 2020 because of COVID-19. Consequently, we only use the data collected in the 2017-2018 cycle.

In this study, the data we use to build our response variable is the examination data of blood pressure (`BPX_J`), which "provides data for three consecutive blood pressure (BP) measurements and other methodological measurements to obtain an accurate BP. Heart rate or pulse, depending on age, are also reported (CDC 2020a)." This data contains 4 readings of systolic blood pressure and 4 readings of diastolic blood pressure for each participant. In order to create a response variable about blood pressure level (BPXLEVEL), we first average the 4 readings of systolic blood pressure and diastolic blood pressure of each participant respectively. Systolic and diastolic blood pressure may be correlated, and predicting both levels simultaneously through a single model allows for capturing potential dependencies and may result in a model that more successfully generalizes the underlying patterns. Thus, we follow the definition of normal, elevated, and hypertension provided by CDC (2021) to divide our average systolic blood pressure and diastolic blood pressure into two blood pressure levels shown in table 1.

Table 1: Blood Pressure Levels Divided by Systolic and Diastolic Blood Pressure

Blood Pressure Levels	Systolic Blood Pressure		Diastolic Blood Pressure
Normal (BPXLEVEL = 0)	< 120 mmHg	and	< 80 mmHg
Elevated or Hypertension (BPXLEVEL = 1)	≥ 120 mmHg	or	≥ 80 mmHg

Our decision to predict blood pressure as classes rather than directly is motivated by several factors. Medical terminology regarding blood pressure may be challenging for individuals without medical training to interpret. By outputting in classes, not only do we allow for easier understanding, but there may also be a possibility of integrating such a model into health apps that do not require the assistance of medical professionals. Additionally, the choice to group elevated and hypertension as one class is due to the risk of a false identification of normal blood pressure being much more harmful than a false identification of abnormal (elevated/hypertension) blood pressure.

After getting the blood pressure levels (BPXLEVEL), we merged two other datasets from the NHANES. The first is Demographic Variables and Sample Weights (DEMO_J) which "provides individual, family, and household-level information (CDC 2020b)." Additionally, we use Dietary Interview - Total Nutrient Intakes, First Day (DR1TOT_J) which contains "detailed dietary intake information from NHANES participants (CDC 2020c)." These are merged based on participants' unique identification number SEQN.

By merging data, selecting relevant predictors, and removing some of the blank data entries, we were left with a data frame with 6125 observations and 16 variables. Details regarding these 16 variables are shown in table 2.

Table 2: Variable Names and Labels in the Curated Data Frame

Name	Label	Source	Name	Label	Source
BPXLEVEL	Blood pressure levels	Derived	DR1TCARB	Carbohydrate (gm)	BPX_J
BPXPLS	60 sec. pulse	DR1TOT_J	DR1TPROT	Protein (gm)	DR1TOT_J
RIAGENDR	Gender	DEMO_J	DR1TFIBE	Dietary fiber (gm)	DR1TOT_J
RIDAGEYR	Age in years at screening	DEMO_J	DR1TTFAT	Total fat (gm)	DR1TOT_J
RIDRETH3	Race/Hispanic origin	DEMO_J	DR1TCHOL	Cholesterol (mg)	DR1TOT_J
DR1TRET	Retinol (mcg)	DR1TOT_J	DR1TVARA	Vitamin A, RAE (mcg)	DR1TOT_J
DR1TACAR	Alpha-carotene (mcg)	DR1TOT_J	DR1TBCAR	Beta-carotene (mcg)	DR1TOT_J
DR1TCRYP	Beta-cryptoxanthin (mcg)	DR1TOT_J	DR1TLYCO	Lycopene (mcg)	DR1TOT_J

2.2 Visualization

In this section, we present two scatterplot matrices that provide a comprehensive visual exploration of the dataset. The first matrix focuses on demographic information, offering insights into the relationships and distributions among key demographic variables. The second matrix encompasses macronutrient intakes, health care details, insulin usage, and the presence of diabetes. These visualizations aim to reveal potential patterns, correlations, and trends within the dataset.

In Figure 1, we utilize a color-coded scheme to represent different blood pressure levels: red for normal and blue for elevated or hypertension. By examining the relationship between blood pressure levels (BPXLEVEL) and gender (RIAGENDR), noteworthy patterns emerge. The plot reveals a higher prevalence of elevated blood pressure and hypertension among male participants (coded as 1) compared to their female counterparts (coded as 2).

Further exploration of blood pressure levels against age (RIAGEYR) reveals intriguing insights. The distributions indicate a skewed pattern, with individuals younger than 20 predominantly exhibiting normal blood pressure

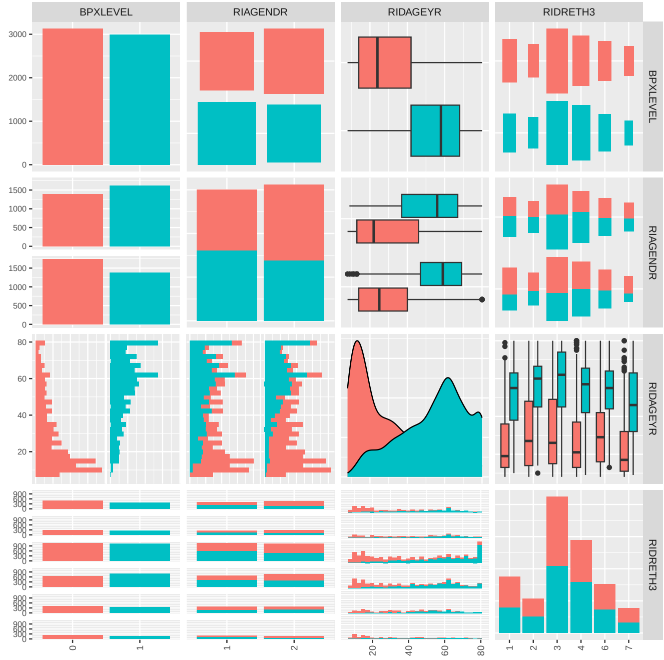

Figure 1: Scatterplot Matrix of BPXLEVEL Against Some Demographic Information

levels. A concerning trend is observed among those around 60 years old, who are more likely to have evevated blood pressure or hypertension. Thus, age emerges as a potential influential factor for predicting blood pressure levels in future models.

Analyzing blood pressure levels against race (RIDRETH3) also unvails an interesting observation. While Mexican American (coded as 1), Non-Hispanic White (coded as 3), Non-Hispanic Asian (coded as 6), and other Hispanic and other races (including multi-racial) (coded 2 and 7) all have more occurences of normal blood pressure than elevated or hypertension, this is not true for Non-Hispanic Black individuals (coded as 4).

Figure 2 uses the same color coding as previously described, and presents a scatter plot matrix investigating the potential impact of macro nutrient intake, including carbohydrates, proteins, fats, and cholesterol (USDA 2022), on blood pressure levels. Histograms of macro nutrient distributions across all three blood pressure levels reveal right-skewed patterns, suggesting no single macro nutrient significantly influences blood pressure.

Notably, the analysis highlights substantial correlations among the macro nutrient variables. The highest correlation is observed between protein intake (DR1TPROT) and fat intake (DR1TTFAT), reaching 0.729. Additional pairs, such as protein intake (DR1TPROT) and daily fiber intake (DR1TFIBE) with a correlation of 0.684, indicate potential multicollinearity among predictor variables. This observation prompts caution when employing certain parametric modeling methods, such as logistic regression, which may be sensitive to multicollinearity issues.

These findings allow for an intricate understanding of the dataset and emphasize the importance of considering demographic and nutritional factors in predicting blood pressure levels. Subsequent sections will delve deeper into statistical analyses and modeling techniques to derive actionable insights from the presented visualizations.

3 Methods

3.1 One-Hot Encoding of Categorical Predictors

Categorical predictors often require transformation into numerical format for compatibility with many machine learning algorithms. We employ one-hot encoding to convert categorical variables, BPACSZ (4 levels), BPXPTY (2 levels), RIAGENDR (2 levels), and RIDRETH3 (6 levels), into a binary matrix, where each category is represented by a binary column. This technique ensures that the categorical nature of the variables is

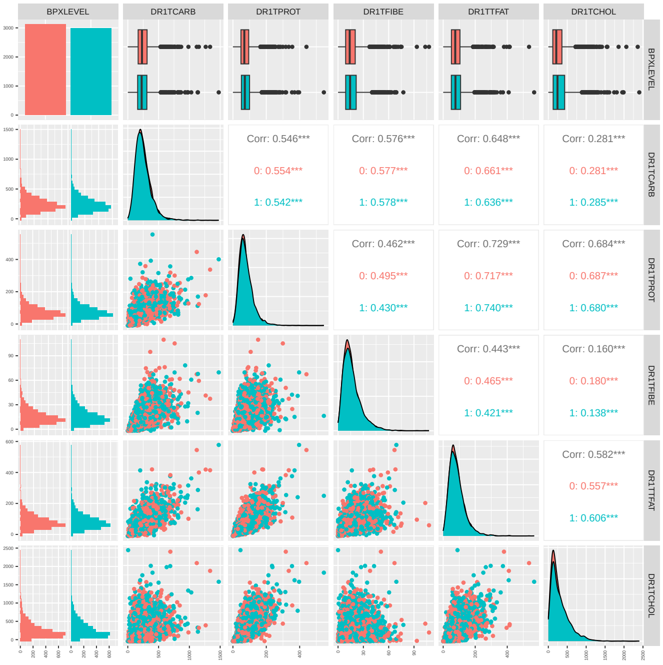

Figure 2: Scatterplot Matrix of BPXLEVEL Against Marco Nutrient Intakes

preserved in the analysis. After applying one-hot encoding to these categorical predictors, our data frame has 21 columns of predictors in total.

3.2 Logistic Regression with Lasso Regularization and 10-Fold Cross Validation as a Baseline

We chose logistic regression with Lasso regularization for the baseline algorithm because of its efficacy in handling diverse data characteristics, and its suitability in this case particularly for binary classification between Normal ($\texttt{BPXLEVEL} = 0$) and Elevated or Hypertension ($\texttt{BPXLEVEL} = 1$) blood pressure levels as classes. Lasso regularization is also important to help prevent overfitting and control model complexity; with the lasso penalty, we will be able to identify which predictors are the most influential in predicting the blood pressure class. Both will benefit the interpretability of the model, by focusing on the most influential predictor variables through regularization, and the interpretability of logistic regression makes it easier to understand the impact of each predictor variable.

To rigorously evaluate our model, we implement a 10-fold cross-validation strategy. This involves dividing the dataset into 10 subsets, training and testing the model 10 times, with each subset serving as the test set exactly once. This approach provides a robust estimate of the model's performance.

The model's performance is systematically assessed through accuracy and Area Under the Curve (AUC) metrics across the 10-fold cross-validation. This thorough evaluation ensures the reliability of our logistic regression baseline in predicting blood pressure categories.

3.3 XGBoost Model with 10-Fold Cross Validation

Extreme Gradient Boosting (XGBoost) is a powerful and flexible ensemble learning method recognized for its predictive strength. The XGBoost algorithm systematically constructs a collection of weak learners, often in the form of decision trees, and amalgamates their predictions to enhance accuracy and generalize well to unseen data. Each sequential learner corrects errors made by its predecessor. Additionally, XGBoost incorporates regularization terms to manage model complexity and prevent overfitting, which is a reason why we chose to utilize this method, as robustness to overfitting is especially advantageous for making accurate predictions.

In our analysis, we employ XGBoost to improve upon the performance achieved by the baseline logistic regression model with Lasso regularization. We will assess the performance of the XGBoost model based on accuracy and the area under the curve (AUC) metrics. We employ 10-fold cross-validation to ensure robust estimation of these metrics across different data subsets, enhancing model reliability and generalizability.

3.4 Feature Selection with XGBoost Feature Importance

One distinctive feature of XGBoost is its ability to provide valuable insights into feature importance. Feature selection is achieved through the computation of importance scores assigned to each predictor, utilizing Gain as the metric, which represents the improvement in accuracy attributed to a specific feature across the model's trees (XGBoost 2022). The higher the importance score assigned to a feature, the more impactful it is considered in the overall predictive capacity of the model.

Leveraging the XGBoost-derived importance scores to determine the most influential predictors is an important advantage that XGBoost provides. The incorporation of these scores and the resultant focus on only the relevant predictors can enhance the predictive performance of our model, providing improved accuracy and AUC scores, and can also improve the interpretability of our model.

4 Results

4.1 Logistic Regression Model with Lasso Regularization

The Logistic regression model was trained with various lasso regularization strengths, spanning a range from low to high values, using a 10-fold cross-validation strategy. The model's logistic deviance was documented for each regularization strength.

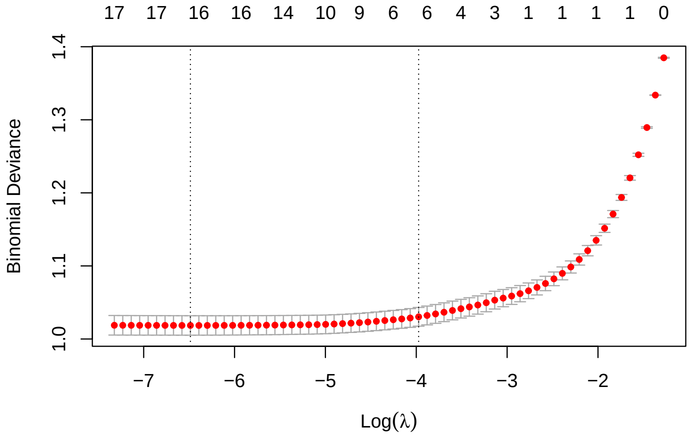

Figure 3: 10-Fold Cross Validation Findinig the Best Lasso Penalty Term

Figure 3 helps identify the optimal λ for the lasso penalty term of the logistic regression. We aim to get the λ which minimizes the logistic deviance. By looking at the figure, we got the smallest logistic deviance when $\lambda = 0.001525$ and only 16 predictors were selected by the lasso penalty, which is indicated by the vertical dash line on the left.

Table 3 shows all coefficients of our predictors and the intercept of the model. Our logistic model chose 16 features, and looks like:

$$\Pr(\texttt{BPXLEVEL} = 1|X) = \frac{1}{1 + e^{-X\beta}}$$

where:

$$X = \begin{bmatrix} 1 & X_{1\ \texttt{BPXPLS}} & \cdots & X_{1\ \texttt{DR1TLYCO}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_{4900\ \texttt{BPXPLS}} & \cdots & X_{4900\ \texttt{DR1TLYCO}} \end{bmatrix}_{4900\times17} \qquad \beta = \begin{bmatrix} \beta_{\texttt{BPXPLS}} \\ \vdots \\ \beta_{\texttt{DR1TLYCO}} \end{bmatrix}$$

The values of all entries in β can be found in the table below:

Table 3: Coefficients of Selected Predictors

PREDICTOR	$\beta_{\text{PREDICTOR}}$	PREDICTOR	$\beta_{\text{PREDICTOR}}$	PREDICTOR	$\beta_{\text{PREDICTOR}}$
Intercept	-3.604823e+00	BPXPLS	8.554252e-03	DR1TCARB	4.086890e-04
RIDRETH32	-2.559511e-01	DR1TPROT	4.613140e-04	DR1TFIBE	-6.618181e-03
RIDRETH33	-3.537319e-01	DR1TTFAT	6.099947e-04	DR1TCHOL	5.028398e-04
RIDRETH34	2.037675e-01	DR1TRET	-3.170559e-04	RIAGENDR1	5.020180e-01
RIDRETH36	-8.467646e-02	DR1TACAR	2.394273e-05	RIAGENDR2	-4.685533e-13
RIDAGEYR	6.508277e-02	DR1TLYCO	-2.473133e-06		

In our logistic regression model, we achieved a test accuracy of 74.45% and the Area Under the Curve (AUC) was calculated as 0.7442811, demonstrating the model's robust discriminative ability across the classes. These results establish a baseline for our future modeling efforts, showcasing the effectiveness of the logistic regression approach in capturing and understanding underlying patterns within the dataset.

4.2 XGBoost Model

The XGBoost model was trained using Extreme Gradient Boosting with exact tree method, a powerful ensemble learning method. The following hyperparameters were modified and utilized in the model:

- Learning Rate (eta): 0.005
- Subsample: 0.75
- Column Subsample: 0.8
- Maximum Depth: 10
- Number of Trees (Rounds): 35

With these hyperparameters, we used 10-fold cross-validation to get a test accuracy of 74.94% and an AUC of 0.754929 The test accuracy is 0.49% higher than that of logistic regression model with lasso regularization, and the test AUC is 0.0106479 higher than that of logistic model. These indicate that the XGBoost model is slightly better than logistic regression model with lasso regularization when classifying the blood pressure levels.

4.3 XGBoost Model with Selected Predictors

Figure 4 shows the Gain scores of the predictors used in the XGBoost model. A higher bar (higher Gain score) represents more important the predictor is. Notably, key predictors such as RIDAGEYR (age), DR1TRET (retinol intake), and DR1TTFAT (total fat intake) emerged as significant contributors to the predictive power of the model.

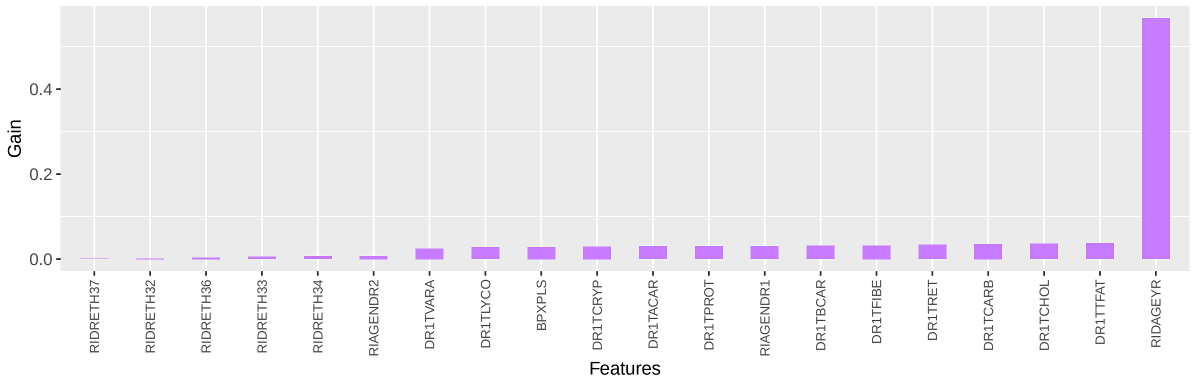

Figure 4: Bar Plots of Gain Score of Each Feathure in the XGBoost Model

In our pursuit of refining the model and unveiling the most impactful predictors, we executed a meticulous feature selection process. We initiated this process by systematically eliminating predictors, starting with the least important (the one with the lowest Gain score), and subsequently assessed the impact on both test accuracy and AUC. This methodical stepwise elimination allowed us to pinpoint a subset of predictors that consistently upheld optimal predictive performance. During this process, we keep using the same hyperparameters we used in the original XGBoost model with 10-fold cross validation at each step.

The results of this feature selection journey revealed a compelling trade-off between the number of predictors and predictive accuracy. Significantly, in figure 5, the model showcased a remarkable test accuracy of 0.7453061% and an AUC of 0.7501558 even with just the top 11 most important predictors. This underscores the efficiency of the selected predictors in encapsulating crucial information for the accurate prediction of health outcomes. As indicated by the red dash line in figure 5, the model achieved the highest test accuracy of 0.7542857% and the highest AUC of 0.7613304 with the top 18 predictors.

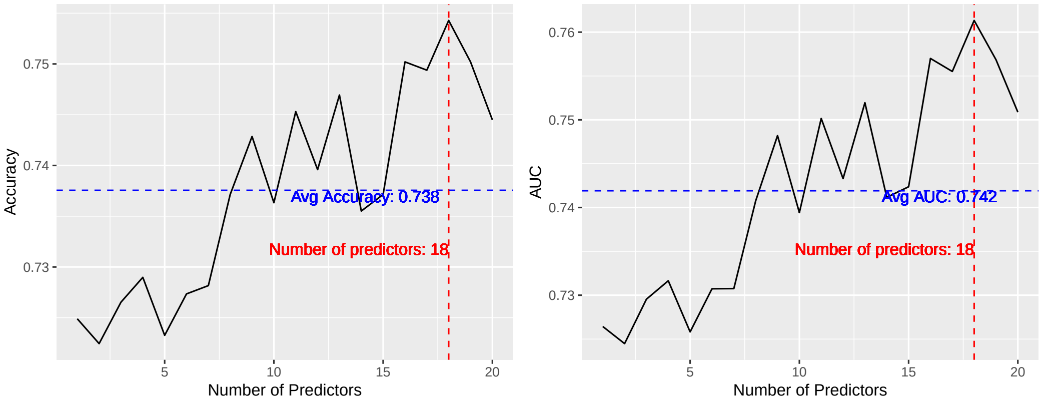

Figure 5: XGBoost Model Test Accuracy and AUC from 1 Predictor to 88 Predictors

Table 4 provides a comprehensive overview of the top predictors identified by the XGBoost model with test accuracy higher than 74.44898%, presenting their corresponding threshold Gain scores, accuracy, and AUC values. The table is thoughtfully organized, with entries sorted based on descending test accuracy, prioritizing higher accuracy models. In cases of ties, the sorting is further refined by considering descending AUC values and, if necessary, the top number of predictors in descending order.

# of features	Threshold Importance	Test Accuracy	AUC
18	0.0037512	0.7542857	0.7613304
16	0.0064800	0.7502041	0.7569965
19	0.0013171	0.7502041	0.7568501
17	0.0054985	0.7493878	0.7555147
13	0.0276062	0.7469388	0.7519480
11	0.0295506	0.7453061	0.7501558
20	0.0007307	0.7444898	0.7508705

Our focus lies on the accuracy and AUC metrics, and, based on these, the model with the top 18 most important predictors stands out as the preferred choice.

Among these 18 selected predictors, there are 16 predictors selected by both the logistic model and the XGBoost model. These predictors are `RIDAGEYR`, `DR1TTFAT`, `DR1TCHOL`, `DR1TCARB`, `DR1TRET`, `DR1TFIBE`, `RIAGENDR1`, `DR1TPROT`, `DR1TACAR`, `BPXPLS`, `DR1TLYCO`, `RIAGENDR2`, `RIDRETH34`, `RIDRETH33`, and `RIDRETH36`. Since both models selected these predictors, indicating the importance of these predictors on predicting blood pressure levels. The other 4 predictors selected by the XGBoost model but not the logistic model are `DR1TBCAR`, `DR1TCRYP`, `DR1TVARA`.

This model exhibits higher increases in accuracy, with -0.0097959% and 0.0097959% improvements compared to the logistic regression model with lasso regularization and the XGBoost model using all predictors, respectively. Moreover, it demonstrates a 0.0104598 and 0.0170493 increase in AUC compared to the logistic regression model with lasso regularization and the XGBoost model using all predictors, respectively.

Our systematic approach to feature selection not only fine-tuned the model but also provided insightful perspectives on the pivotal factors influencing its predictive power. This enhanced interpretability contributes to a more robust and effective health outcome prediction system.

5 Conclusion

These findings offer valuable insights into predicting blood pressure levels and identifying predictors associated with hypertension. However, it is crucial that we acknowledge and address the potential issue of multicollinearity surfaced in our visualization. This presence of collinearity may introduce challenges for the interpretation and stability of the logistic regression model. Given the degree of correlation found between the variables, caution is warranted regarding attributing individuals impacts to any single variable. Thus, in future refinements, further feature selection or regularization techniques may be explored to mitigate the effects of multicollinearity and enhance the robustness of the predictive models. Ultimately, while the results found shed light on predicting blood pressure, this drawback needs to be addressed in further work.

6 Reproducibility

To reproduce this analysis and results, the appendix contains all of our code, also included in the R Markdown file attached to the submission. Running this code should successfully generate the figures and results provided in this report.

7 Appendix: R Script

```r
rm(list = ls())
knitr::opts_chunk$set(echo = TRUE)
set.seed(88)
library(caret)
library(dplyr)
library(GGally)
library(ggpubr)
library(glmnet)
library(grid)
library(gridExtra)
library(haven)
library(knitr)
library(patchwork)
library(pROC)
library(tidyr)
library(xgboost)
knitr::opts_chunk$set(fig.pos = "ht", out.extra = "")
findNonNAColumns = function(data) {
  return(colnames(data)[apply(data, 2, function(col) all(!is.na(col)))])
}
# Import 2017 - 2018 blood pressure data
# Doc: https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/BPX_J.htm
BPX_J = read_xpt("https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/BPX_J.XPT")
BPX_J = BPX_J %>%
  mutate(BPXSYAVG = rowMeans(select(., starts_with("BPXSY")), na.rm = TRUE),
         BPXDIAVG = rowMeans(select(., starts_with("BPXDI")), na.rm = TRUE)) %>%
  filter(complete.cases(BPXSYAVG, BPXDIAVG)) %>%
  mutate(BPXLEVEL = case_when(
    # Normal blood pressure
    BPXSYAVG < 120 & BPXDIAVG < 80 ~ 0,
    # Elevated blood pressure or Hypertension
    (BPXSYAVG >= 120) | (BPXDIAVG >= 80) ~ 1
  ))
columns_no_na = findNonNAColumns(BPX_J)
BPX_J = BPX_J[c("BPXLEVEL", "SEQN","BPXPLS")]
BPX_J = BPX_J %>%
  mutate(
    BPXLEVEL = as.factor(BPXLEVEL)
  )
# Import 2017 - 2018 demographic data
# Doc: https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DEMO_J.htm
DEMO_J = read_xpt("https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DEMO_J.XPT")
currentVar = colnames(BPX_J)
FULLDATA = BPX_J %>% left_join(DEMO_J, by = "SEQN")
columns_no_na1 = setdiff(findNonNAColumns(FULLDATA), currentVar)
FULLDATA = FULLDATA[c(currentVar, "RIAGENDR", "RIDAGEYR", "RIDRETH3")]
FULLDATA = FULLDATA %>%
  mutate(
    RIAGENDR = as.factor(RIAGENDR),
    RIDRETH3 = as.factor(RIDRETH3)
  )

# Import 2017 - 2018 Total Nutrient Intakes, First Day
# Doc: https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DR1TOT_J.htm
currentVar = colnames(FULLDATA)
DR1TOT_J = read_xpt("https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DR1TOT_J.XPT")
FULLDATA = FULLDATA %>% left_join(DR1TOT_J, by = "SEQN")
FULLDATA = FULLDATA[c(currentVar, "DR1TCARB", "DR1TPROT", "DR1TFIBE","DR1TTFAT", "DR1TCHOL", "DR1TRET",
FULLDATA = na.omit(FULLDATA)

# Prepare data for training and testing
FULLDATA = FULLDATA[c("BPXLEVEL", "BPXPLS", "DR1TCARB", "DR1TPROT", "DR1TFIBE","DR1TTFAT", "DR1TCHOL",
n.obs = nrow(FULLDATA)
index.train = sample(seq(n.obs), floor(n.obs * 0.8), replace = FALSE)
# Data frame train and test
train = FULLDATA[index.train, ]
test = FULLDATA[-index.train, ]
train.X = train[, -1]
train.Y = train$BPXLEVEL
test.X = test[, -1]
test.Y = test$BPXLEVEL
# Model matrix train and test
train.X.mm = model.matrix(~ . - 1, train.X)
test.X.mm = model.matrix(~ . - 1, test.X)
# Data frame train and test with dummy
train.X.dummy = cbind(train.Y, as.data.frame(train.X.mm))
colnames(train.X.dummy)[1] = "BPXLEVEL"
test.X.dummy = cbind(test.Y, as.data.frame(test.X.mm))
colnames(test.X.dummy)[1] = "BPXLEVEL"
scatter.matrix1 = ggpairs(FULLDATA[c("BPXLEVEL", "RIAGENDR", "RIDAGEYR", "RIDRETH3")],
        aes(color = BPXLEVEL)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1, size = 8),
        axis.text.y = element_text(size = 7))
pdf(file = "report_files/figure-latex/scattermatrix1.pdf", width = 8, height = 8)
print(scatter.matrix1)
dev.off()
knitr::include_graphics("report_files/figure-latex/scattermatrix1.pdf")
scatter.matrix2 = ggpairs(FULLDATA[c("BPXLEVEL", "DR1TCARB", "DR1TPROT", "DR1TFIBE",
                                     "DR1TTFAT", "DR1TCHOL")],
        aes(color = BPXLEVEL)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1, size = 5),
        axis.text.y = element_text(size = 5))
pdf(file = "report_files/figure-latex/scattermatrix2.pdf", width = 10, height = 10)
print(scatter.matrix2)
dev.off()
knitr::include_graphics("report_files/figure-latex/scattermatrix2.pdf")
set.seed(88)
# Softmax classification with 10 fold CV
cvfit = cv.glmnet(train.X.mm, as.matrix(train.Y), family = "binomial", alpha=1)
plot(cvfit)
coef.df = as.data.frame(as.matrix(coef(cvfit, s = cvfit$lambda.min)))
colnames(coef.df) = c("Coefficients")
coef.df = coef.df %>% filter_all(all_vars(. != 0))
cvfit$lambda.min
cvfit$index
softmax.predictions = predict(cvfit, newx = test.X.mm,
                              s = cvfit$lambda.min, type = "class")
mean(softmax.predictions == test.Y)
# ROC - AUC
softmax.rocr = roc(as.numeric(test.Y) - 1, as.numeric(softmax.predictions) - 1)
softmax.rocr$auc
confusionMatrix(as.factor(softmax.predictions), test.Y)
lr.acc = round(confusionMatrix(as.factor(softmax.predictions), test.Y)$overall[['Accuracy']]*100, digits
# Gradient boosting with 10 fold CV
xgboost.train = function(col) {
  set.seed(88)
  xgb.data = xgb.DMatrix(data = as.matrix(train.X.dummy[col]),
                         label = recode(train.X.dummy$BPXLEVEL,'0'=0, '1'=1))
  xgb.test.X = data.matrix(test.X.dummy[col])
  hyperparameters = list(
    eta = 0.005,
    subsample = 0.75,
    col_subsample = 0.8,
    max_depth = 10
  )

  params = list(
    eta = hyperparameters$eta,
    subsample = hyperparameters$subsample,
    colsample_bytree = hyperparameters$col_subsample,
    max_depth = hyperparameters$max_depth,
    tree_method = "exact",
    objective = "binary:logistic"
  )

  cv.xgb = xgb.cv(
    params = params,
    data = xgb.data,
    nfold = 10,
    metrics = "error",
    verbose = 0,
    nrounds = 35
  )

  eval.log = as.data.frame(cv.xgb$evaluation_log)
  min.merror = min(eval.log[, 4])
  min.merror.index = which.min(eval.log[, 4])
  xgb.model = xgboost(params = params,
                      data = xgb.data,
                      nrounds = min.merror.index,
                      verbose = 0)
  xgb.predictions = predict(xgb.model, xgb.test.X)
  xgb.predictions = ifelse(xgb.predictions > 0.5, 1, 0)
  return(list(xgb.model, xgb.predictions))
}
xgboost.all = xgboost.train(colnames(train.X.dummy)[2:dim(train.X.dummy)[2]])
mean(xgboost.all[[2]] == test.Y)
xgb.auc = multiclass.roc(as.numeric(test.Y) - 1,
               as.numeric(xgboost.all[[2]]) - 1)$auc
confusionMatrix(as.factor(xgboost.all[[2]]), test.Y)
importance.matrix = xgb.importance(colnames(train.X.dummy)[2:dim(train.X.dummy)[2]], model = xgboost.all
importance.matrix
xgb.acc = round(confusionMatrix(as.factor(xgboost.all[[2]]), test.Y)$overall[['Accuracy']]*100, digits
importance.plot = xgb.ggplot.importance(importance.matrix, measure = 'Gain') +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1, size = 8),
        axis.text.y = element_text(size = 10)) +
  geom_bar(aes(fill = "blue"), stat = "identity") +
  ylab("Gain") + coord_cartesian() + ggtitle("")
pdf(file = "report_files/figure-latex/importance_plot.pdf", width = 10, height = 3.5)
print(importance.plot)
dev.off()
knitr::include_graphics("report_files/figure-latex/importance_plot.pdf")
threshold = sort(importance.matrix$Gain)
n.feature = seq(length(importance.matrix$Gain))
threshold.importance = c()
test.accuracy = c()
test.auc = c()
for (i in 1:length(threshold)) {
  selected.features = importance.matrix$Feature[which(importance.matrix$Gain >= threshold[i])]
  xgboost.all = xgboost.train(selected.features)
  accuracy = mean(xgboost.all[[2]] == test.Y)
  auc = multiclass.roc(as.numeric(test.Y) - 1,
                       as.numeric(xgboost.all[[2]]) - 1)$auc
  threshold.importance[i] = threshold[i]
  test.accuracy[i] = accuracy
  test.auc[i] = auc
  print(i)
}
df.xgb = data.frame(n.feature = sort(n.feature, decreasing = TRUE),
                    threshold.importance = threshold.importance,
                    test.accuracy = test.accuracy,
                    test.auc = test.auc)
xgb.accuracy = ggplot(df.xgb, aes(x = n.feature, y = test.accuracy)) +
  geom_line() +
  geom_vline(xintercept = df.xgb$n.feature[which.max(df.xgb$test.accuracy)],
             linetype = "dashed", color = "red") +
  geom_hline(yintercept = mean(df.xgb$test.accuracy), linetype = "dashed", color = "blue") +
  geom_text(aes(label = sprintf("Avg Accuracy: %.3f", mean(df.xgb$test.accuracy))),
            x = max(df.xgb$threshold.importance), y = mean(df.xgb$test.accuracy),
            vjust = 1, hjust = -1.5, color = "blue") +
  geom_text(aes(label = sprintf("Number of predictors: %d", df.xgb$n.feature[which.max(df.xgb$test.accu
            x = df.xgb$n.feature[which.max(df.xgb$test.accuracy)],
            y = max(df.xgb$test.accuracy),
            vjust = 20, hjust = 1, color = "red") +
  labs(x = "Number of Predictors",
       y = "Accuracy")
xgb.auc = ggplot(df.xgb, aes(x = n.feature, y = test.auc)) +
  geom_line() +
  geom_vline(xintercept = df.xgb$n.feature[which.max(df.xgb$test.auc)],
             linetype = "dashed", color = "red") +
  geom_hline(yintercept = mean(df.xgb$test.auc), linetype = "dashed", color = "blue") +
  geom_text(aes(label = sprintf("Avg AUC: %.3f", mean(df.xgb$test.auc))),
            x = max(df.xgb$threshold.importance), y = mean(df.xgb$test.auc),
            vjust = 1, hjust = -2.5, color = "blue") +
  geom_text(aes(label = sprintf("Number of predictors: %d", df.xgb$n.feature[which.max(df.xgb$test.auc)]
            x = df.xgb$n.feature[which.max(df.xgb$test.auc)],
            y = max(df.xgb$test.auc),
            vjust = 20, hjust = 1, color = "red") +
  labs(x = "Number of Predictors",
       y = "AUC")
xgb.plots = (xgb.accuracy | xgb.auc)
pdf(file = "report_files/figure-latex/xgb_plots.pdf", width = 10, height = 4)
print(xgb.plots)
dev.off()
knitr::include_graphics("report_files/figure-latex/xgb_plots.pdf")
df.xgb.sorted <- df.xgb %>% arrange(desc(test.accuracy), desc(test.auc), desc(n.feature))
colnames(df.xgb.sorted) <- c("# of features", "Threshold Importance", "Test Accuracy", "AUC")
knitr::kable(df.xgb.sorted[1:7,])
intersection = intersect(unname(unlist(importance.matrix[1:18, 1])),
                         rownames(coef.df))
intersection
setdiff(unname(unlist(importance.matrix[1:18, 1])), intersection)
```

CDC. 2020a. "2017-2018 Data Documentation, Codebook, and Frequencies Blood Pressure (BPX_j)." wwwn.cdc.gov. https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/BPX_J.htm.

———. 2020b. "2017-2018 Data Documentation, Codebook, and Frequencies Demographic Variables and Sample Weights (DEMO_j)." wwwn.cdc.gov. https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DEMO_J.htm.

———. 2020c. "2017-2018 Data Documentation, Codebook, and Frequencies Dietary Interview - Total Nutrient Intakes, First Day (DR1TOT_j)." wwwn.cdc.gov. https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DR1TOT_J.htm.

———. 2021. "Facts about Hypertension." Centers for Disease Control; Prevention. https://www.cdc.gov/bloodpressure/facts.htm.

———. 2023a. "2019-2020 Examination Data - Continuous NHANES." wwwn.cdc.gov. https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?BeginYear=2019.

———. 2023b. "NHANES Tutorials - Sample Design Module." wwwn.cdc.gov. https://wwwn.cdc.gov/nchs/nhanes/tutorials/sampledesign.aspx.

Forouzanfar, Mohammad H, Ashkan Afshin, Lily T Alexander, H Ross Anderson, Zulfiqar A Bhutta, Stan Biryukov, Michael Brauer, et al. 2016. "Global, Regional, and National Comparative Risk Assessment of 79 Behavioural, Environmental and Occupational, and Metabolic Risks or Clusters of Risks, 1990–2015: A Systematic Analysis for the Global Burden of Disease Study 2015." *The Lancet* 388 (October): 1659–1724. https://doi.org/10.1016/s0140-6736(16)31679-8.

USDA. 2022. "Macronutrients | National Agricultural Library." Usda.gov. https://www.nal.usda.gov/human-nutrition-and-food-safety/food-composition/macronutrients.