Valentinas Sungaila

# Math 6388 Final Project: Multivariate Adaptive Regression Splines

## Introduction

Multivariate adaptive regression splines better know as MARS was first introduces in 1991 By Jerome H, Friedman. The term MARS is trademarked and licensed by Salford Systems company and thus many other statistical software call the implementation of MARS as “Earth”. MARS is a non-parametric regression method that can model multiple nonlinear relationships in the data. Thus, the method can be applied to many various datasets and studies. As such this paper will explore what MARS models are, how MARS model can be applied to a case study, and how well do different MARS models can predict the “health” of an individual.

## Case Study Description

To see how the MARS model performs The Behavioral Risk Factor Surveillance System (BRFSS) was chosen as the dataset to apply the method. The survey collects state data about U.S. residents regarding their health-related risk behaviors, chronic health conditions, and use of preventive services. The dataset can be found on the Center for Disease Control website. BRFSS is a health-related telephone survey that is gathered every year. The dataset contains 231,396 observations of individuals with 33 variables. The variables are all categorical as shown in the examples below.

Value represents the assigned category for each variable and its meaning. Frequency in dataset is the number of that observation for each variable category.

Variable: Adults with good or better health

| Value | Value Meaning | Frequency in Dataset |
|---|---|---|
| 1 | Good or Better Health | 4057 |
| 2 | Fair or Poor Health | 696 |

Variable: Income Categories

| Value | Value Meaning | Frequency in Dataset |
|---|---|---|
| 1 | Less than $15,000 | 268 |
| 2 | $15,000 to less than $25,000 | 541 |
| 3 | $25,000 to less than $35,000 | 365 |
| 4 | $35,000 to less than $50,000 | 559 |
| 5 | $50,000 or more | 2473 |
| 9 | Don't know/Not sure/Missing | 556 |

Variable: Smoking Status

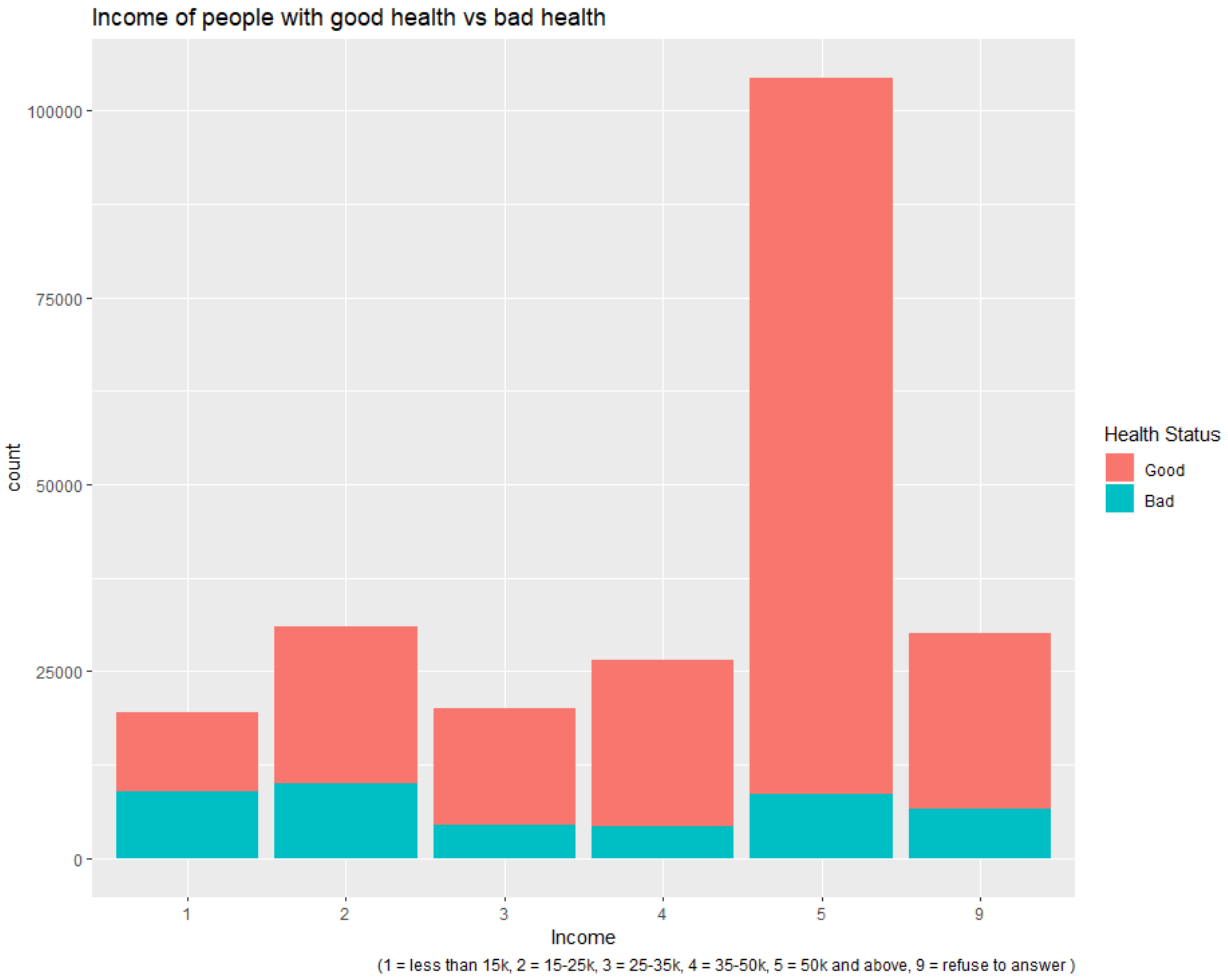
| Value | Value Meaning | Frequency in Dataset |
|---|---|---|
| 1 | Current smoker - now smokes every day | 462 |
| 2 | Current smoker - now smokes some days | 220 |
| 3 | Former smoker | 1238 |
| 4 | Never smoked | 2524 |
| 9 | Don't know/Refused/Missing | 318 |

The goal of the MARS method is to see how well the method can model "Adults with good or better health" variable, given other behavioral related aspects from the BRFSS survey.

First split the data into 75% training set and 25% test set and compare how well the MARS model predicts health vs a logistic regression by looking at the mean squared error (MSE).

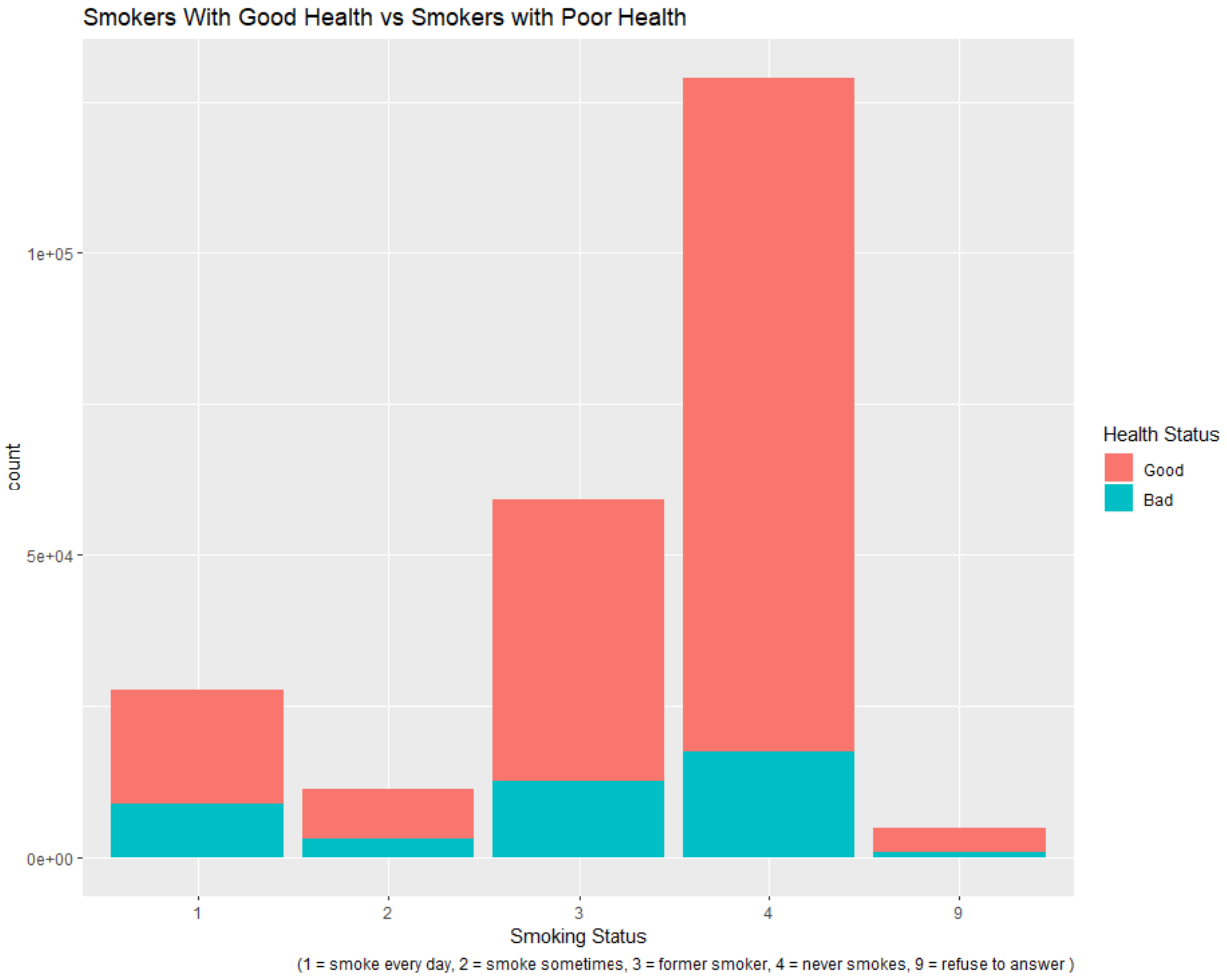
To do some data exploration lets see how peoples income breaks down depending on their health.

Graph 1




When people make above 50k it seems like they are reporting good health status. Also the majority of people with bad health are in the bottom two income tiers.

Graph 2

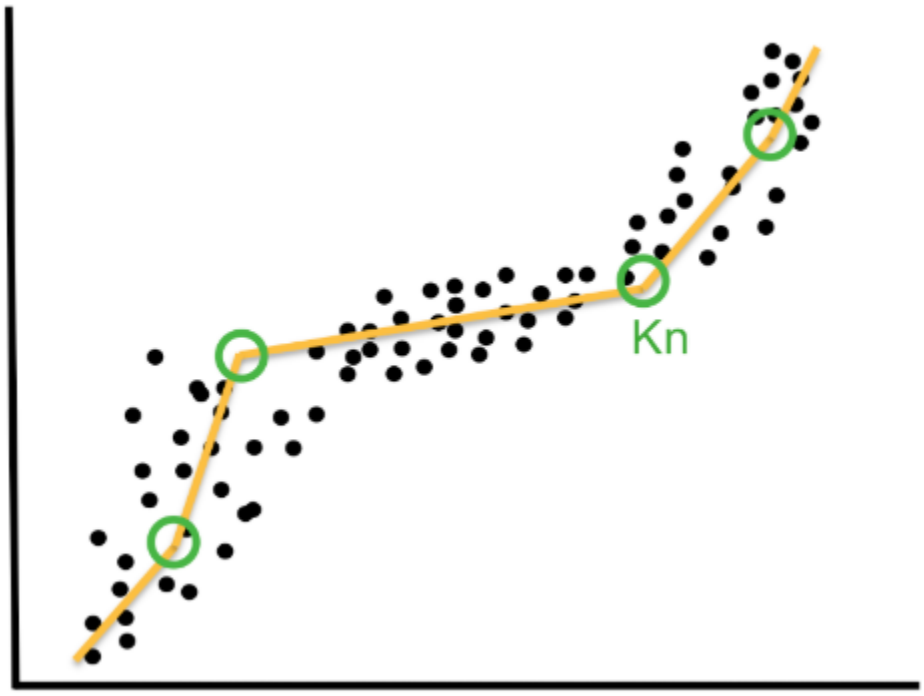



It seems like people who are reporting good health status are either former smokers or never smoked.

**Model and Methods**

The best way to understand MARS models is to look at how the model fits some data visually.

Figure 1




https://support.bccvl.org.au/support/solutions/articles/6000118097-multivariate-adaptive-regression-splines

As it can be seen in figure 1, the MARS algorithm partitions the data into a few linear lines. The kinks in the graph where the line changes slope are know as knots or cut points. As the number of knots increases, so does the flexibility of the model. But a linear relationship is still held between the two knots.

Since MARS models are so flexible, the models can deal with both, quantitative and discrete variables. MARS is also a supervised method because the method needs a response variable to operate. MARS does not assume any distributional assumptions. One fault that the MARS method has is that the model can not run if there are missing values in the predictors. There is a technique called a surrogate split that allows missing values, but most statistical software implementations do not perform such operation because that is used more for regression tree analysis rather than MARS.

To fit the data the MARS model creates a pair of hinge functions for each cut point $\alpha$, where a pair of hinge functions are $h(x-\alpha)$ and $h(\alpha-x)$. The hinge functions are defined below.

Equation 1

$$h(x-\alpha) = \begin{cases}(x-\alpha), & x > \alpha \\ 0, & x \leq \alpha\end{cases} = (x-\alpha) * I(x > \alpha)$$

$$h(\alpha-x) = \begin{cases}(\alpha-x), & x < \alpha \\ 0, & x \geq \alpha\end{cases} = (\alpha-x)$$

*In a linear regression context, it would look this this.*

Equation 2

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1\big((x-\alpha) * I(x > \alpha)\big) + \hat{\beta}_2\big((\alpha-x) * I(x < \alpha)\big) = \hat{\beta}_0 + \hat{\beta}_1 h(x-\alpha) + \hat{\beta}_2 h(\alpha-x)$$

For $h(x-\alpha)$ if x > α then the line will have a $\hat{\beta}_1$slope. If x < α then $h(x-\alpha) = 0$ and the slope would be $\hat{\beta}_2$ because the new hinge function $h(\alpha-x)$ would be used. Hinge functions give the range for each slope. So, the range for the slope of $\hat{\beta}_2$ when looking at

equation 2 goes from 0 to α. Once x reaches α (where the knot is in the graph), the slope change to $\hat{\beta}_1$ for the range of α to infinity.

To fit the data, the MARS model looks at each data point for each predictor as a knot. The MARS model first starts with a process called “forward pass”. Where it looks thought each variable and finds a place in the data that gives the maximum reduction in residual sum of squares (RSS). This method can be considered a greedy algorithm because it is possible for the MARS method to find a local optimum but not a global one when searching for the right place for the hinge functions because of the high number of covariates and the pairwise correlation structure of those covariates. Terms in the MARS model are defined as the hinge functions and the intercept. Once a term is found, a linear regression is applied to find the coefficient of that term. This process of adding terms continues until the change in RSS is very small (usually user defined) or until the maximum number of terms is reached. Statistical software R by default uses expected change in $R^2$ of less than 0.001 to decide when to stop adding more terms.

After this, the MARS model runs a process called “backward pass” where MARS removes terms one by one, deleting the least effective term at each step until it finds the best sub model. Since with “forward pass” once a variable is included it cannot be excluded anymore. But with “backward pass” any variable can be excluded even if it was the first one included by the “forward pass”. Also, with “forward pass” when a knot is found a pair of hinge functions is added but “backwards pass” can throw out one of the two hinge functions if one part of the hinge function does not help with RSS. This gives MARS the power to do automatic variable selection because if a variable does not add anything to the RSS it will not be included in the

final model. Cross validation can be used by running the “forward pass” and the “backward pass” multiple times to make sure that global optimum is reached.

**Analysis and Results**

Using R statistical software with the earth function from the earth package, four different MARS models were built for the BRFSS dataset. Model 1 uses the earth function in R and allows R to decide the optimal number of terms using change in $R^2$ as less than 0.001. Model 2, still uses the change in $R^2$ of 0.001 to decide the optimal model, but uses 10-fold cross validation that is run ten times. Since the outcome variable is binary 1 for great health and 2 for poor health, Model 3 imposes a binomial family on the MARS model. Meaning that the MARS model estimates a generalized linear model (GLM) using the MARS process. Model 4 is just model 3 estimated using 10-fold cross validation that is run ten times. Both models 3 and 4 use change in $R^2$ of 0.001 to decide the optimal model as well. The resulting output of the four models can be seen below.

Table 1

| Model 1 Output | | Model 2 Output | |
|---|---|---|---|
| Hinge Function | Coefficients | Hinge Function | Coefficients |
| (Intercept) | 1.36447394 | (Intercept) | 1.36428068 |
| h(X.MENT14D-2) | 0.11068088 | h(X.MENT14D-2) | 0.11069374 |
| h(3-X.MENT14D) | -0.06779827 | h(3-X.MENT14D) | -0.06779839 |
| h(X.MENT14D-3) | -0.12828564 | h(X.MENT14D-3) | -0.12833287 |
| h(2-X.TOTINDA) | -0.11511948 | h(2-X.TOTINDA) | -0.11492606 |
| h(3-X.ASTHMS1) | 0.05481577 | h(X.TOTINDA-2) | 0.00618667 |
| h(X.ASTHMS1-3) | 0.01566009 | h(3-X.ASTHMS1) | 0.05481279 |
| h(2-X.EXTETH3) | -0.04899384 | h(X.ASTHMS1-3) | 0.01565178 |
| h(2-X.DENVST3) | -0.04236109 | h(2-X.EXTETH3) | -0.04898930 |
| h(58-X.AGE80) | -0.00396815 | h(2-X.DENVST3) | -0.04234912 |
| h(X.AGE80-58) | -0.00062390 | h(58-X.AGE80) | -0.00396817 |
| h(2-X.EDUCAG) | 0.10445282 | h(X.AGE80-58) | -0.00062512 |
| h(X.EDUCAG-2) | -0.01349715 | h(2-X.EDUCAG) | 0.10446278 |
| h(5-X.INCOMG) | 0.04270898 | h(X.EDUCAG-2) | -0.01350161 |
| h(X.INCOMG-5) | 0.01764412 | h(5-X.INCOMG) | 0.04271284 |
| h(4-X.SMOKER3) | 0.01590706 | h(X.INCOMG-5) | 0.01763062 |
| h(X.SMOKER3-4) | 0.00355082 | h(4-X.SMOKER3) | 0.01591369 |
| h(7-DROCDY3. ) | 0.00731214 | h(X.SMOKER3-4) | 0.00354748 |
| h(DROCDY3.-7) | 0.00001390 | h(7-DROCDY3.) | 0.00731293 |
| - | - | h(DROCDY3.-7) | 0.00001381 |

MARS model 1 estimated 19 terms with 10 predictors while MARS model 2 using 10-fold cross validation estimates 20 terms with 10 predictors. For both models three most important variables were as estimated by the MARS model using residual sum of squares (RSS). X.INCOMG which is people's income where 1 is less than 15k earned, 2 is between 15k and 25k earned, 3 is between 25k and 35k earned, 4 is between 35k and 50k earned, 5 is 50k or more earned, and 9 is refuse to answer . X.MENT14D which is mental health status where 1 is 0 days mental health not good, 2 is 1-13 days mental health not good, 3 is 14+ days mental health not good, 9 is

refuse to answer. X.TOTINDA which is if people exercise where 1 had physical activity, 2 no physical activity in the last 30 days, 9 refuse to answer.

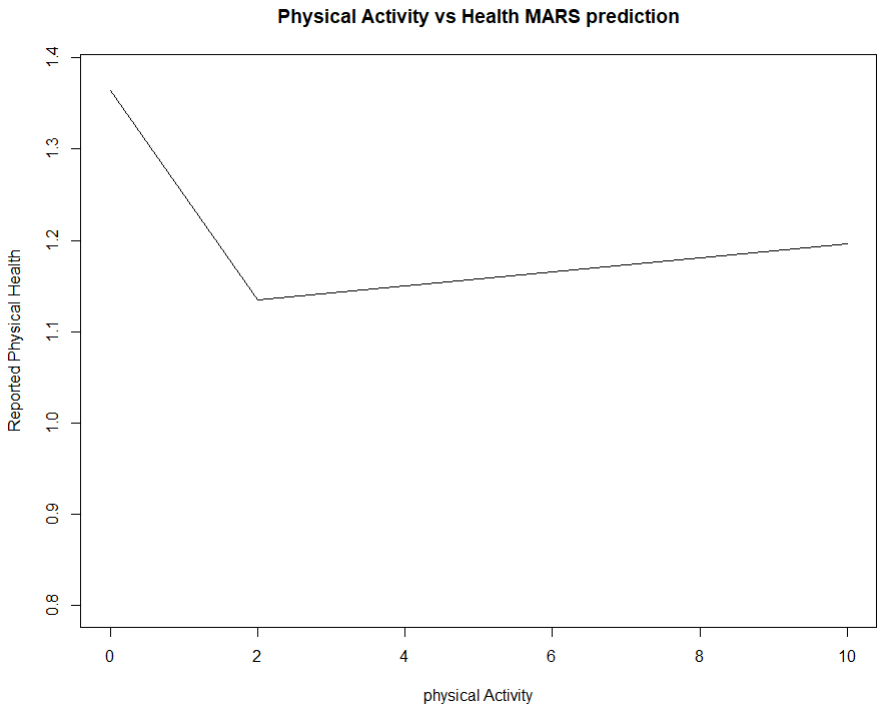
When looking at table 1, a hinge function for X.TOTINDA can be read as

(2 – physical activity) coefficient =-0.115

(physical activity – 2) coefficient = 0.006

With MARS these coefficients can still be interpreted. (2 – physical activity) can be interpreted as your physical level approaches 1 on the x axis, your health gets better because 1 is being approached on the y axis as well. But, for (physical activity – 2) as physical activity start decreasing x approaches 2, 3, 4 your health also decreases starts approaching 2 on the y axis. A graphical representation of the hinge functions can be seen in graph 3.

Graph 3



To show how MARS model selects the amount of terms and variables graph 4 shows us a model selection plot that shows the Generalized cross-validation (GCV) R^2 (left-hand y-axis and solid black line) based on the number of terms retained in the model (x-axis) which are constructed from a certain number of original predictors (right-hand y-axis). The vertical dashed lined at 20 tells us the optimal number of terms retained where marginal increases in GCV R^2 are less than 0.001. RSq refers to R-squared and GRSq refers to the GRS statistic is the Gibbons et al. (1989) statistic that tests whether the estimated intercepts from a multiple regression model are jointly zero.

Graph 4

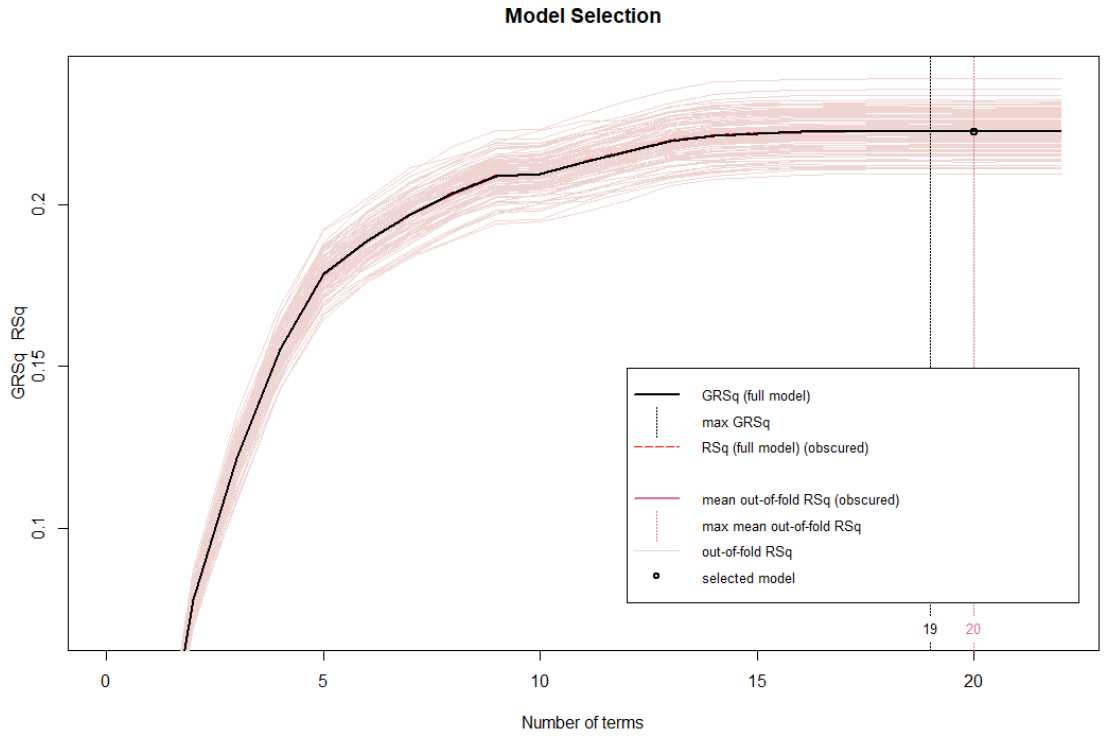


Table 2

| Model 3 output | | Model 4 output | |
|---|---|---|---|
| Hinge Function | Coefficient | Hinge Function | Coefficient |
| (Intercept) | -0.33945308 | (Intercept) | -0.33599062 |
| h(X.MENT14D-2) | 0.31331409 | h(X.MENT14D-2) | 0.31405017 |
| h(3-X.MENT14D) | -0.59799306 | h(3-X.MENT14D) | -0.59707654 |
| h(X.MENT14D-3) | -0.42535505 | h(X.MENT14D-3) | -0.42594224 |
| h(2-X.TOTINDA) | -0.77061042 | h(2-X.TOTINDA) | -0.77063924 |
| h(3-X.ASTHMS1) | 0.35837001 | h(3-X.ASTHMS1) | 0.35821925 |
| h(X.ASTHMS1-3) | 0.10242588 | h(X.ASTHMS1-3) | 0.10247152 |
| h(2-X.EXTETH3) | -0.40561941 | h(2-X.EXTETH3) | -0.40627161 |
| h(2-X.DENVST3) | -0.31235654 | h(2-X.DENVST3) | -0.31225371 |
| h(58-X.AGE80) | -0.03145729 | h(58-X.AGE80) | -0.03164506 |
| h(X.AGE80-58) | 0.00072018 | h(2-X.EDUCAG) | 0.44335107 |
| h(2-X.EDUCAG) | 0.44359652 | h(X.EDUCAG-2) | -0.15931583 |
| h(X.EDUCAG-2) | -0.15934402 | h(5-X.INCOMG) | 0.29924685 |
| h(5-X.INCOMG) | 0.29905552 | h(X.INCOMG-5) | 0.16246137 |
| h(X.INCOMG-5) | 0.16214115 | h(4-X.SMOKER3) | 0.12288437 |
| h(4-X.SMOKER3) | 0.12330712 | h(X.SMOKER3-4) | 0.04788493 |
| h(X.SMOKER3-4) | 0.04794701 | h(7-DROCDY3. ) | 0.06181258 |
| h(7-DROCDY3. ) | 0.06176134 | | |

MARS model 3 which imposed a binomial distribution on the response variable estimated 18 terms with 10 predictors while MARS model 4 which imposed a binomial distribution on the response variable and used 10-fold cross validation estimates 17 terms with 10 predictors. For both models three most important variables were the same as for model 1 and model 2.

To check how well the MARS model does a logistic regression was ran on the same data. The mean squared error (MSE) was calculated and is shown in the table below.

Table 3

| MARS Model 1 | MARS Model 2 | MARS Model 3 | MARS Model 4 | Logistic regression |
|---|---|---|---|---|
| MSE = 0.1172956 | MSE = 0.1172968 | MSE = 5.798832 | MSE = 5.798224 | MSE = 5.080228 |

It seems like the MARS model 1 where are does all the background work and MARS model 2 where cross-validation was used had the best MSE.

**Discussion**

Overall, it seems like MARS performed really well on the BRFSS dataset. The model predicted that the 3 biggest factors that effect how healthy you feel are income, your mental health, and exercise, which makes a lot of sense in real life terms. When compared to a logistic regression using MSE MARS models blew it out of the water if a binomial distribution was not specified. Unlike many other machine learning methods MARS models are still interpretable, so for future research getting rid of all the "did not answer" answers and only focus on people who answered all the questions might lead to interesting results.

As seen in graph 4, $R^2$ is around 20% meaning that this data is not doing the best job at explaining the variation in health. Or there is just too much noise. Nonetheless MARS model gives the best of both worlds a flexible model and an interpretable result which allowed for a model to be built with a very small MSE.

References

"CDC - BRFSS." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 5 Nov. 2019, www.cdc.gov/brfss/index.html.

Greenwell, B. "Hands-On Machine Learning with R." *Chapter 7 Multivariate Adaptive Regression Splines*, 1 Feb. 2020, bradleyboehmke.github.io/HOML/mars.html.

"Multivariate Adaptive Regression Spline." *Wikipedia*, Wikimedia Foundation, 10 Apr. 2020, en.wikipedia.org/wiki/Multivariate_adaptive_regression_spline.

"Multivariate Adaptive Regression Splines." *ARDC Nectar Support*, 2019, support.bccvl.org.au/support/solutions/articles/6000118097-multivariate-adaptive-regression-splines.

"Multivariate Adaptive Regression Splines." *Multivariate Adaptive Regression Splines · UC Business Analytics R Programming Guide*, uc-r.github.io/mars.

Prabhakaran, S. "Caret Package - A Complete Guide to Build Machine Learning in R." *Machine Learning Plus*, 20 May 2018, www.machinelearningplus.com/machine-learning/caret-package/.

# Appendix

```
# Machine Learning final project

# Valentinas Sungaila

# Loading all my packages

library(SASxport)

library(plyr)

library(earth)     # for fitting MARS models

library(caret)

library(ISLR)

library(dplyr)

library(tidyverse)

library(ggplot2)

# Data cleaning

#set the working directory



setwd("C:/Users/sunga/Desktop/machine learning final project")

# Read in an xpt file



lookup.xport("LLCP2018.xpt")

brfss_Survey_Data_2018  <- read.xport("LLCP2018.xpt")

# how to open .xpt files found here https://www.phusewiki.org/wiki/index.php?title=Open_XPT_File_with_R

# Limit my data a little to be more focused

# let me focus on people who only live in a private residence, (By private residence, we mean someplace like a house or
        apartment.)

brfss_Survey_Data_2018 = subset(brfss_Survey_Data_2018, PVTRESD3 == 1)

# focus on people currently living in the state they took the survey at

brfss_Survey_Data_2018 = subset(brfss_Survey_Data_2018, CSTATE1 == 1)
```

```
# focus on people who are either male or female at birth insterad of people who refused to answer

brfss_Survey_Data_2018 = subset(brfss_Survey_Data_2018, SEX1 <= 2)

# make it so the weight is only in pounds

brfss_Survey_Data_2018 = subset(brfss_Survey_Data_2018, WEIGHT2 <= 0999)

# make it so height is only in feet and inches

brfss_Survey_Data_2018 = subset(brfss_Survey_Data_2018, HEIGHT3 <= 711)

# replace the number of children from 88 for 0 children to 0 for 0 children

brfss_Survey_Data_2018$CHILDREN[brfss_Survey_Data_2018$CHILDREN == 88] = 0

# cleaning up data for variables I do not need because they are duplicates or some variation of the same question

brfss_Survey_Data_2018 <- brfss_Survey_Data_2018[ -c(2:219) ]

brfss_Survey_Data_2018 <- brfss_Survey_Data_2018[ -c(13, 42:43, 47:57) ]

brfss_Survey_Data_2018 <- brfss_Survey_Data_2018[ -c(1,3) ]

# seems to have worked properly but now I have some NA means for some of my variables which means I will need to remove
    them

# keeping only the columns without na values

# found https://r.789695.n4.nabble.com/how-to-delete-columns-with-NA-values-td1839902.html

# 4th comment

brfss_Survey_Data_2018 = brfss_Survey_Data_2018[,colSums(is.na(brfss_Survey_Data_2018)) == 0]

# make my response which is adults with good or better health into a binary 1,2 variable

brfss_Survey_Data_2018 = subset(brfss_Survey_Data_2018, X.RFHLTH <= 2)

# 1 means people are in good or better health

# 2 means people are in fair or poor health

table(brfss_Survey_Data_2018$X.RFHLTH)

# looks good

# finished with data cleaning
```

```
##############################################################################################

# Since I have a binary response variable if I want to use a binary GLM in my MARS function to check for model fit I need to
        make my response

# which is adults with good or better health into a binary 0,1 variable so I can see if there is any effect if I specify a distribution
        for my response

brfss_Survey_Data_2018_binomial = mutate(brfss_Survey_Data_2018, X.RFHLTH_binary = X.RFHLTH - 1)

# function to move columns around

moveme <- function (invec, movecommand) {

 movecommand <- lapply(strsplit(strsplit(movecommand, ";")[[1]],

                        ",|\\s+"), function(x) x[x != ""])

 movelist <- lapply(movecommand, function(x) {

  Where <- x[which(x %in% c("before", "after", "first",

                    "last")):length(x)]

  ToMove <- setdiff(x, Where)

  list(ToMove, Where)

 })

 myVec <- invec

 for (i in seq_along(movelist)) {

  temp <- setdiff(myVec, movelist[[i]][[1]])

  A <- movelist[[i]][[2]][1]

  if (A %in% c("before", "after")) {

   ba <- movelist[[i]][[2]][2]

   if (A == "before") {

    after <- match(ba, temp) - 1

   }

   else if (A == "after") {

    after <- match(ba, temp)

   }
```

```
    }
    else if (A == "first") {
      after <- 0
    }
    else if (A == "last") {
      after <- length(myVec)
    }
    myVec <- append(temp, values = movelist[[i]][[1]], after = after)
  }
  myVec
}
# function creator found here
# https://stackoverflow.com/questions/3369959/moving-columns-within-a-data-frame-without-retyping/18540144#18540144
# The basic options are:
#
#  first
#  last
#  before
#  after
# Compounded moves are separated by a semicolon.
# example
# moveme(names(df), "g first")
# moveme(names(df), "g first; a last; e before c")
# move my variable of interest which is X.RFHLTH first then X.RFHLTH_binary
```

```
brfss_Survey_Data_2018_binomial = brfss_Survey_Data_2018_binomial[moveme(names(brfss_Survey_Data_2018_binomial),
          'X.RFHLTH_binary after X.RFHLTH')]

# remove the unnecessary column

brfss_Survey_Data_2018_binomial <- brfss_Survey_Data_2018_binomial[ -1 ]

###-----------------------------------------------------------------------------------------------------------------------------------

# Exploratory analysis

ggplot(data = brfss_Survey_Data_2018, aes(x = as.factor(brfss_Survey_Data_2018$X.INCOMG),

                                  fill = as.factor(brfss_Survey_Data_2018$X.RFHLTH))) +

  geom_bar(stat="count") +

  labs(x = "Income", title = "Income of people with good health vs bad health",

       colour = "Health",

       caption = "(1 = less than 15k, 2 = 15-25k, 3 = 25-35k, 4 = 35-50k, 5 = 50k and above, 9 = refuse to answer )") +

  scale_fill_discrete(name="Health Status",

                      labels=c("Good", "Bad"))

# Histogram of income

hist(brfss_Survey_Data_2018$X.INCOMG,freq = F)

ggplot(data = brfss_Survey_Data_2018, aes(x = as.factor(brfss_Survey_Data_2018$X.SMOKER3),

                                  fill = as.factor(brfss_Survey_Data_2018$X.RFHLTH))) +

  geom_bar(stat="count") +

  labs(x = "Smoking Status", title = "Smokers With Good Health vs Smokers with Poor Health",

       colour = "Health",

       caption = "(1 = smoke every day, 2 = smoke sometimes, 3 = former smoker, 4 = never smokes, 9 = refuse to answer )") +

  scale_fill_discrete(name="Health Status",

                      labels=c("Good", "Bad"))




####-----------------------------------------------------------------------------------------------------------------------------------
```

```
# First, split the data into a train set and test set for the normal data.

# 75% Train

# 25% Test

set.seed(1)

n = length(brfss_Survey_Data_2018$X.RFHLTH) # number of observations

tr = sample(1:n, size = floor(n*.75))

t = (1:n)[-tr]

train_set = data.frame(brfss_Survey_Data_2018[tr, ])

test_set = data.frame(brfss_Survey_Data_2018[t, ])

train_n = length(tr)

test_n = length(t)

###

# my data with a 0,1 response variable  into a train set and test set.

# 75% Train

# 25% Test

n_binomial = length(brfss_Survey_Data_2018_binomial$X.RFHLTH_binary) # number of observations

tr_binomial = sample(1:n_binomial, size = floor(n_binomial*.75))

t_binomial = (1:n_binomial)[-tr_binomial]

train_set_binomial = data.frame(brfss_Survey_Data_2018_binomial[tr_binomial, ])

test_set_binomial = data.frame(brfss_Survey_Data_2018_binomial[t_binomial, ])

train_n_binomial = length(tr_binomial)

test_n_binomial = length(t_binomial)

###-----------------------------------------------------------------------------------------------------------


# running the mars model where the number of knots is decided by R
```

```
mars1_train = earth(X.RFHLTH ~ ., data = train_set)

print(mars1_train)

summary(mars1_train)

plot(mars1_train, which = 1)

# The model selection plot that shows us the Generalized cross-validation (GCV) R^2

#(left-hand y-axis and solid black line) based on the number of terms retained in the model

#(x-axis) which are constructed from a certain number of original predictors (right-hand y-axis).

#The vertical dashed lined at 14 tells us the optimal number of terms retained where marginal increases

#in GCV R^2 are less than 0.001.

# summarize the importance of input variables

evimp(mars1_train)

#Show the variables that are important

## make predictions on training data

predictions <- predict(mars1_train, train_set)

#summarize accuracy of training

mse <- mean((predictions - train_set$X.RFHLTH)^2)

print(mse)

## make predictions on test data

predictions2 <- predict(mars1_train, test_set)

#summarize accuracy of test

mse2 <- mean((predictions2 - test_set$X.RFHLTH)^2)

print(mse2)



###-----------------------------------------------------------------------------
```

```
# running the mars model where the number of knots is decided by R but we use 10-fold cross validation

mars2_train = earth(X.RFHLTH ~ ., data = train_set, nfold = 10, ncross = 10, trace = .5, pmethod="cv")

print(mars2_train)

summary(mars2_train)

plot(mars2_train, which = 1)

plotd(mars2_train)

# The model selection plot that shows us the Generalized cross-validation (GCV) R^2

#(left-hand y-axis and solid black line) based on the number of terms retained in the model

#(x-axis) which are constructed from a certain number of original predictors (right-hand y-axis).

#The vertical dashed lined at 14 tells us the optimal number of terms retained where marginal increases

#in GCV R^2 are less than 0.001.

# summarize the importance of input variables

evimp(mars2_train)

#Show the variables that are important

## make predictions on training data

predictions3 <- predict(mars2_train, train_set)

#summarize accuracy of training

mse3 <- mean((predictions3 - train_set$X.RFHLTH)^2)

print(mse3)

## make predictions on test data

predictions4 <- predict(mars2_train, test_set)

#summarize accuracy of test

mse4 <- mean((predictions4 - test_set$X.RFHLTH)^2)

print(mse4)

# still get the same thing


# compare MARS models
```

```
plot.earth.models(list(mars2_train,mars1_train))


#-----------------------------------------------------------------------------------------------------

# let me specify a binomial family glm for the model since my response is 1 and 2 to see if that makes any difference


# running the mars model where the number of knots is decided by R


mars3_train = earth(X.RFHLTH_binary ~ ., data = train_set_binomial, glm=list(family=binomial))

print(mars3_train)

summary(mars3_train)

plot(mars3_train, which = 1)

# The model selection plot that shows us the Generalized cross-validation (GCV) R^2

#(left-hand y-axis and solid black line) based on the number of terms retained in the model

#(x-axis) which are constructed from a certain number of original predictors (right-hand y-axis).

#The vertical dashed lined at 14 tells us the optimal number of terms retained where marginal increases

#in GCV R^2 are less than 0.001.

# summarize the importance of input variables

evimp(mars3_train)

#Show the variables that are important

## make predictions on training data

predictions5 <- predict(mars3_train, train_set_binomial)


#summarize accuracy of training

mse5 <- mean((predictions5 - train_set_binomial$X.RFHLTH_binary)^2)

print(mse5)
```

```
## make predictions on test data

predictions6 <- predict(mars3_train, test_set_binomial)

#summarize accuracy of test

mse6 <- mean((predictions6 - test_set_binomial$X.RFHLTH_binary)^2)

print(mse6)

###-------------------------------------------------------------------------

# running the mars model where the number of knots is decided by R but we use 10-fold cross validation

memory.limit(size=51000)

mars4_train = earth(X.RFHLTH_binary ~ ., data = train_set_binomial, nfold = 10, trace = .5, pmethod="cv",
        glm=list(family=binomial))

print(mars4_train)

summary(mars4_train)

plot(mars4_train, which = 1)

plot(mars4_train)

# The model selection plot that shows us the Generalized cross-validation (GCV) R^2

#(left-hand y-axis and solid black line) based on the number of terms retained in the model

#(x-axis) which are constructed from a certain number of original predictors (right-hand y-axis).

#The vertical dashed lined at 14 tells us the optimal number of terms retained where marginal increases

#in GCV R^2 are less than 0.001.

# summarize the importance of input variables

evimp(mars4_train)

#Show the variables that are important

## make predictions on training data

predictions7 <- predict(mars4_train, train_set_binomial)



#summarize accuracy of training

mse7 <- mean((predictions7 - train_set_binomial$X.RFHLTH_binary)^2)
```

```
print(mse7)

## make predictions on test data

predictions8 <- predict(mars4_train, test_set_binomial)

#summarize accuracy of test

mse8 <- mean((predictions8 - test_set_binomial$X.RFHLTH_binary)^2)

print(mse8)

# still get the same thing

plot.earth.models(list(mars3_train,mars4_train))



#-----------------------------------------------------------------------------------------

# foud how to  train MARS models with different values of nk, thresh and span (including minspan and endspan)

# nk = Maximum number of model terms before pruning

# thresh = Forward stepping threshold

# span = minspan: Minimum number of observations between knots. endspan: Minimum number of observations before the first
        and after the final knot

# A simulation can be conducted to show how different values of nk, thresh, minspan and endspan affects the model-training
        process

# https://blog.zenggyu.com/en/post/2018-06-16/multivariate-adaptive-regression-splines-in-a-nutshell/



results <- crossing(nk = c(5, 10, 20),

                    thresh = c(0, 0.01, 0.1),

                    span = c(1, 5, 10)) %>%

  pmap(function(nk, thresh, span, train_set) {

    fit <- earth(X.RFHLTH ~ ., data = train_set,

                 degree = 1, nprune = NULL,

                 nk = nk, thresh = thresh, minspan = span, endspan = span)

    mutate(train_set, nk = nk, thresh = thresh, span = span,

           y_predicted = predict(fit)[,1], p = length(coef(fit)))
```

```
  }, train_set = train_set) %>%
  bind_rows()


# the number of terms included in each model with different parameters (the output is attached below)
results %>%
  select(nk, thresh, span, p) %>%
  distinct() %>%
  mutate(nk = as.factor(nk) %>% fct_relabel(function(x) {paste0("nk=", x)})) %>%
  split(.$nk) %>%
  map(function(x) {
    x %>%
      select(-nk) %>%
      spread(key = span, value = p, sep = "=") %>%
      as.data.frame() %>%
      `rownames<-`(paste0("thresh=", .$thresh))  %>%
      select(-thresh)
  })
3##############################################
# creating segments to show for MARS model


plot(0, 0, col = "white",
     main = "Physical Activity vs Health MARS prediction",
     xlab = "Physical Activity",
     ylab = "Reported Physical Health",
     xlim = c(0, 10),
     ylim = c(0.8, 1.38))
```

```
segments(x0 = c(0,2),
         y0 = c(1.36428068,1.134429),
         x1 = c(2,10),
         y1 = c(1.134429,1.196296))


####################################----------------------------------------------------------------------------------


# lets model the data with a logistic regression and see how it compares.
logit = glm(X.RFHLTH_binary ~ ., data = train_set_binomial, family=binomial)
summary(logit)
## make predictions on training data
predictions9 <- predict(logit, train_set_binomial)
#summarize accuracy of training
mse9 <- mean((predictions9 - train_set_binomial$X.RFHLTH_binary)^2)
print(mse9)
## make predictions on test data
predictions10 <- predict(logit, test_set_binomial)
#summarize accuracy of test
mse10 <- mean((predictions10 - test_set_binomial$X.RFHLTH_binary)^2)
print(mse10)
```