Debt Modeling and Prediction

Casey Liu

March 2019

1 Summary

Consumer debt was approaching 14-trillion in the third quarter of 2018, according to the New York Federal Reserve. Understanding how debt associate with each individual has become an urgent topic for nowadays financial situation. In this project, we are modeling and making predictions to total debt in an American household. Generalized Linear Model(GLM) and Generalized Additive Partial Linear Model was compared with five Non-parametric Machine Learning Model method (K Nearest Neighbour (KNN), Random Forest, Decision Tree, Neural Network (NN), Support Vector Machine (SVM)). The best model was selected based on mean square error(MSE).

2 Introduction

In this project, we are focusing on modeling and predicting debt of American household based on household general information. Data set includes 8690 records with 13 features for each records (as shown in appendix). Data set was further cleaned with removing unnecessary columns and imputation. GLM with log link is used instead of log transformation of response variable. Generalized linear model with logistic regression was performed as a method of outlier detection. Data set was split into training group and testing grouping before further analysis (detailed information is included in appendix).

3 Modeling

The Data set was split into training set, with 5781 observations and 9 features, and test set, with 2889 observations and 9 features. Models were trained based on training set, and measured standard error (MSE) based on test set.

3.1 GLM

A basic GLM logistic regression with respect to all 9 predictors was fitted. There is 25 parameters in the full logistic model. We decide to proceed with AIC selection, which will reduce number of predictors and increase degree of freedom. The resulting model excludes nFu(number of people in a household), eduWife (education level of wife), headMarital (the marital status of domain person). (model formula and assumption check is in appendix)

3.1.1 Generalized Additive Partial Linear Model (GAPLM)

Since it is only meaningful to fit the model with significant predictors, we choose to proceed to GAM procedure with 5 features selected by AIC. Among those 5 predictors, 3 are continuous, which are ageHead (age of domain person),FamilyIncome and eduExp(education expense). We fit GAM with all 3 continuous predictors with logit link and all categorical predictors as linear predictor function using gam package in R (as shown in appendix).
When using package gam, we choose to use cubic smoothing splines and local regression. The estimated additive function f_j are plotted (as shown in appendix, included detailed discussion).
We could see that for cubic spline, ageHead have nonlinear pattern, FamilyIncome and eduExp have linear pattern. For local regression, all three have linear pattern. Thus we compute a potential GAPLM model by adding non-parametric cublic spline smoothing function to ageHead to AIC selected GLM model (as shown in appendix).

3.1.2 Modified GLM based on GAM observation

Based on graph of GAM (as shown in appendic for previous subsection), we could observe a nonlinear pattern for ageHead. Thus we decide to add a quadratic term to GLM model selected by AIC procedure. The likelihood ratio test between with quadratic term model and without quadratic term model is 0.007532, which suggest it rejected null hypothesis and showed evidence that adding a quadratic term is more preferable.

Moreover, interaction between continuous variable was examined. And the likelihood ratio test between with interaction and without interaction is < 0.05, which suggest it rejected null hypothesis and showed evidence that adding interaction terms is more preferable.

3.2 Machine Learning Model

All Machine Learning Model was fitted using Scikit-Learn Package in python.

1. Decision Tree

A decision tree is a tree which each node represents a feature (Bhukya, 2010), each link represents a decision and each leaf represents an outcome. The whole idea is to create a tree like this for the entire data and process a single outcome at every lead (or minimize the error in every leaf.) Decision Tree is good for us to visualize the result (as shown in graph, included detailed discussion).

2. Random Forest

Random forest has many classification trees. When a new object need to be classify, just simply put the new object as an input vector down to each of the trees in the forest (Afanador, 2016). Each tree will gibe a classification, the class with most "votes" stands for the final class random forest choose for that new object. Here, we set our number of tree as 100. Since there is too many trees, it is merely impossible for us to visualize random forest.

3. K Nearest Neighbour

K Nearest Neighbour (KNN) algorithm is to separate points in data into several classes based on distance to predict the classification of a new sample point(Rithesh, 2017). Here we set the number of neighbours to 100, in order to achieve a reasonable MSE. Note that 100 neighbours is a lot when our observation number is around 8000.

4. Neural Network

Neural Network (NN) is an algorithms predicting results by keep updating weights of every nodes in multiple hidden layers (DeWeese, 1996). The most simple NN update weight backwards by calculating error. Here we set the hidden layer size as 10, and alpha is 0.8, in order to converge within 1000 iteration.

5. Support Vector Machine

Support Vector Machine is to plot each data item as a point in n-dimensional space (n is number of predictors, in this case, $n = 11$) with the value of each predictor being the value of a particular coordinate (Rithesh, 2017). Then, perform classification by finding the hyper-plane that differentiate the two classes very well. Here linear kernel was used, thus we are trying to find a linear hyper-plane between classes. This is the

most time consuming algorithm, since it involves with multiple matrix calculation.

4 Discussion

Mean square error (MSE) was calculated, and a histogram for MSE was generated based on seven different prediction models (as shown in appendix). Based on the graph we could see that all GLM models has smaller MSE than all machine learning models. In those two GLM models, glm with interaction has the smallest MSE.

Thus we are choosing glm with interaction and quadratic term as our final model as shown below (see summaries and coefficient in appendix).

$$log(E(totalDebt|X)) = \eta = \beta_0 + \beta_1 ageHead + \beta_2 ageHead^2 + \beta_{3-6} 1_{lifeSat} + \beta_{7-12} 1_{educHead} + \beta_{13} FamilyIncome + \beta_{14} eduExp + \beta_{15} ageHead * educExp + \beta_{16} FamilyIncome * educExp \quad (1)$$

Based on two tables of summaries of effect, we can observe that all predictors are significant. Based on coefficient table, we can observe that(with completely satisfied as a baseline) if life satisfaction are very satisfied (lifeSat = 2) or somewhat satisfied (lifeSat = 3), their tendency to have large amount of debt will decrease by 0.08252 and 0.1437 respectively, whereas not very satisfied (lifeSat = 4) and not at all satisfied (lifeSat = 5) will increase by 0.0987 and 0.3764 respectively. Notice that the magnitude of change for different lifeSat is different. Compared with family satisfied with life, family not satisfied with life is more likely to have a large debt. Thus, we might conclude that if a family is not satisfied with life, they might have large debt. For educHead (education level of head) predictor, the magnitude of coefficient is increasing in following order: AA, MD, others, BS, MS, JD, PhD. It is ambiguous to say higher or education lower degree will serve as factor for large debt. Since person with higher degree does not mean he/she will end up with a higher pay job. It is also bizarre to observe PhD will have larger debt than MD and JD. Since tuition for PhD are majorly sponsored by school, but MD and JD are often self-pocket. In conclusion, purely based on coefficient of different education levels, we could say, people with associated of art (AA) will have the lowest debt, whereas people with PhD will have the largest debt.

For ageHead predictor, we could set all other predictor as 0, then we will get a $y = ax^2 + bx + c$ curve, with x = ageHead, a = -0.0003529, b = 0.02018, c = 9.436 (as

shown in appendix). We can observe the maximum is around 30, which means total Debt of a family unit will increase before the head of family is 30, then decrease after 30. This is reasonable, since we would expected a person will only have student loan around 20s, and might have car loan and house loan on top of student loan around 30s, then start paying back slowly after 30s. Moreover, for the same age, a increase educExp will decrease the amount of debt, which might indicate for families with the same ageHead, family with high education expenditure might have lower debt.

For educExp (education expenditure in 2008) and FamilyIncome (family income in 2008) predictor, we can see amount of debt will increase with large value of educExp or FamilyIncome (since $\beta > 0$) . Moreover, notice the interaction term between educExp and ageHead, and between FamilyIncome and educExp are positive. This means for the same educExp, a increasing value in ageHead and familyincome would decrease the amount of debt. For the same family income, a increasing value in educExp will decrease the amoung of debt. This is reasonable since we would expect family with older parents and high income are more likely to have small debt. In summary, although FamilyIncome and educExp has increasing effect on debt, debt will decrease for large value of FamilyIncome*educExp and ageHead*educExp.

Since interaction terms involves in the final model, it is difficult to interpret contribution for toalDebt with respect to each predictor. However, based on categorical predictors and ageHead, we could roughly say that family with Head older than 30 , high life satisfaction rate and head education level is Associate of Art tends to have small amount debt. Family with Head younger than 30, low life satisfaction rate, head education level is PhD tends to have large amount of debt. More detailed information is needed when interpreting FamilyIncome and educExp predictors.

5 Conclusion

We select GLM model with interaction and quadratic term as our final model, by comparing accuracy and recall for 1 GLM model, 1 GAPLM model and 5 non-parametric machine model. GLM and GAPLM models both have smaller MSE comapred to machine learning models. This might suggest machine learning model are over fitting, since our data set only has lower than 10k observation with only 9 features. Moreover, by comparing among GLM and GAPLM models, GAPLM model with local regression has the higher MSE. Thus GLM model with interaction and quadratic term is the best choice.

As is discussion part, we observe abnormal trend for people with PhD has more debt than people with medical or law degree. Since we would assmue medical and law student has more education debt than PhD student . This problem can be further investigated by adding another interaction term: educHead*eduExp. Additional interaction might be important for further anlysis, such as ageHead*lifeSat, educHead*FamilyIncome. Moreover, notice even our best model has a large MSE, this might caused by we have not centering our data at the beginning. Moreover, notice we have a interaction part in the final model, centering might help with interpretation for interaction terms.

6 Reference

Afanador, N., Smolinska, A., Tran, T. and Blanchet, L. (2016). Unsupervised random forest: a tutorial with case studies. Journal of Chemometrics, 30(5), pp.232-241.
Bhukya, D. and Ramachandram, S. (2010). Decision Tree Induction: An Approach for Data Classification Using AVL-Tree. International Journal of Computer and Electrical Engineering, pp.660-665.
DeWeese, M. (1996). Optimization principles for the neural code. Network: Computation in Neural Systems, 7(2), pp.325-331.
Rithesh, R. (2017). SVM-KNN: A Novel Approach to Classification Based on SVM and KNN. International Research Journal of Computer Science, 4(8).

7 Appendix

7.1 Introduction

7.1.1 Data Clean and Missing Values

Below is a graph of column information in raw data.

```
ER42001         RELEASE NUMBER
ER42002         2009 FAMILY INTERVIEW (ID) NUMBER
ER42016         # IN FU
ER42017         AGE OF HEAD
ER42020         # CHILDREN IN FU
ER42023         HEAD MARITAL STATUS
ER42024         A3 LIFE SATISFACTION
ER43612         W39 VALUE ALL DEBTS
ER46474         K55 HGHST COLLEGE DEGREE RECD-WF
ER46568         L55 HGHST COLLEGE DEGREE RECD-HD
ER46851         HEAD AND WIFE TAXABLE INCOME-2008
ER46935         TOTAL FAMILY INCOME-2008
ER46971D1       EDUCATION EXPENDITURE 2008
```

Figure 1: Column Information

Note the first two column (Release number and Family ID) are irrelevant for this data analysis, thus were removed from data set. headMarital, lifeSat, educWife and eduHead were converted to categorical variable. In the original dataset, NA was replace by specific numbers, such as 999 or 98. Thus we replace those number by NA and proceed with imputation steps using mice package in R.

Below is a graph showing missing values in each column. Notice that only eduWife, eduHead, totalDebt and lifeSat have missing values. Notice both of them has less than 10 percentage missing value, thus imputation step computed on those columns using predictive mean matching method in mice.

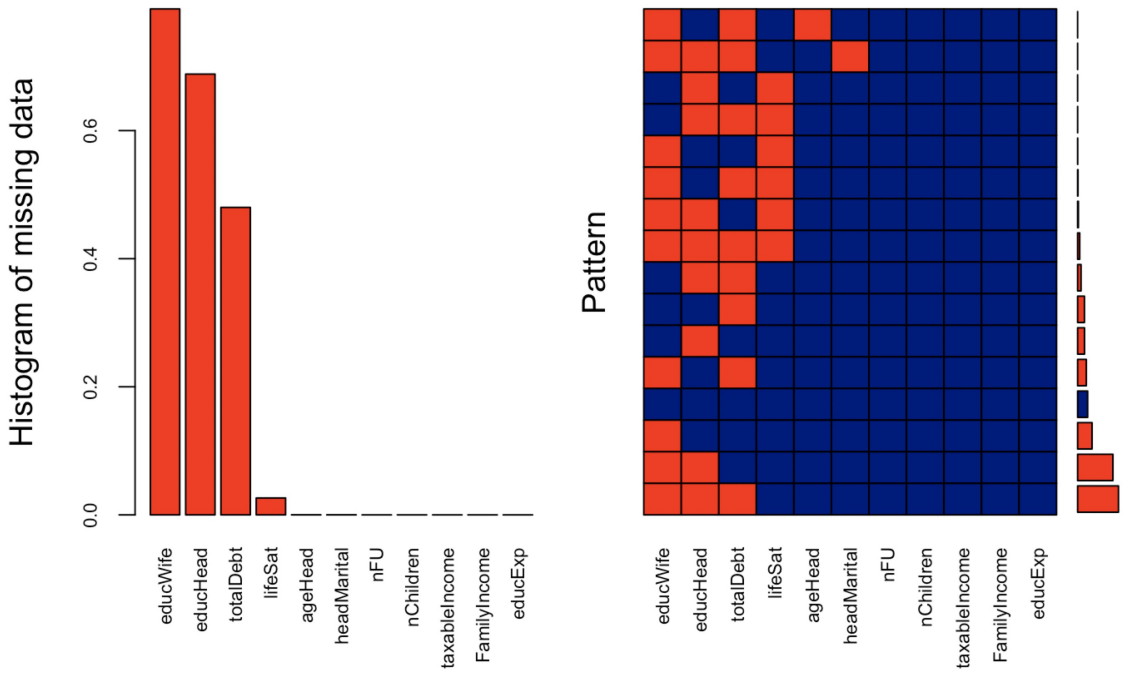

Figure 2: Missing Value

Based on correlation plot (as shown below), nFU nChildern, FamilyIncome taxableIncome are highly correlated, thus remove nChildern and taxableIncome from dataset.

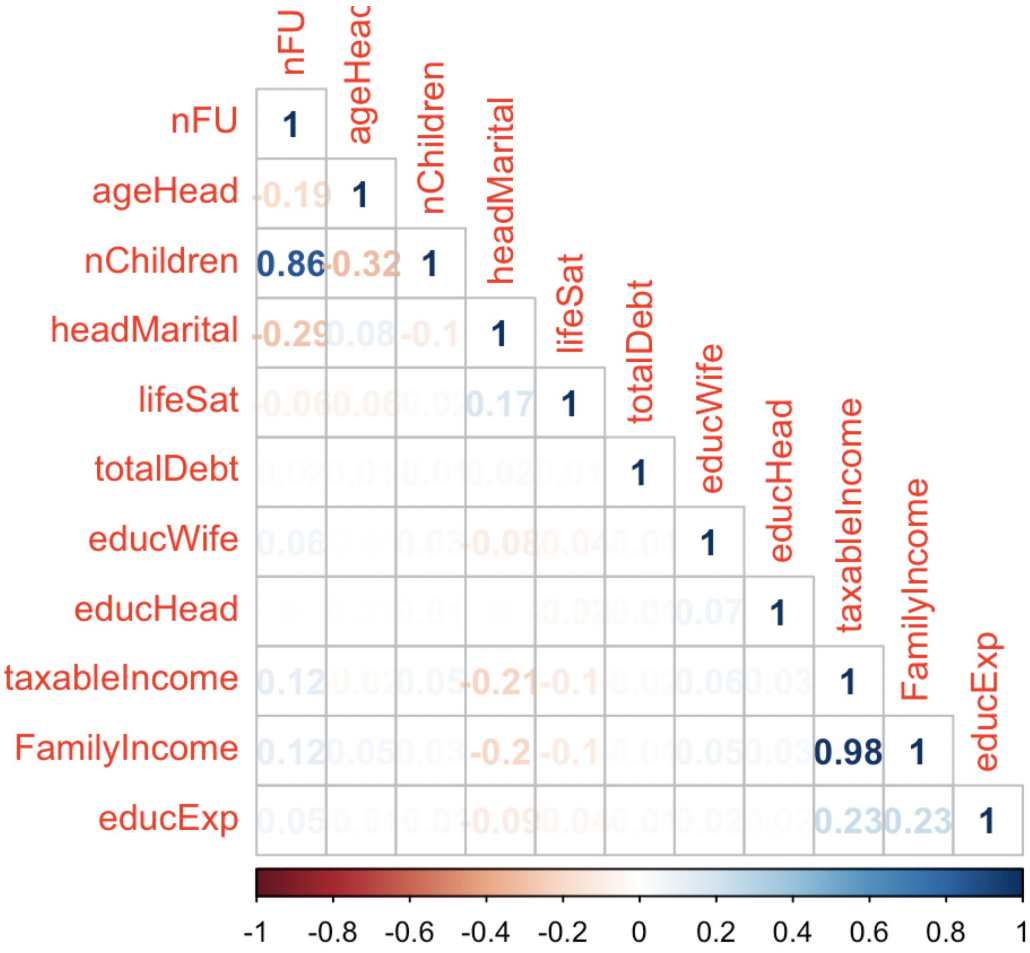

Figure 3: Correlation Plot

Note that distribution of totalDebt is extremely skewed with lots of outliers, thus transformation is needed before outlier detection. A log transformation would make distribution of totalDebt normal-like (as shown below).

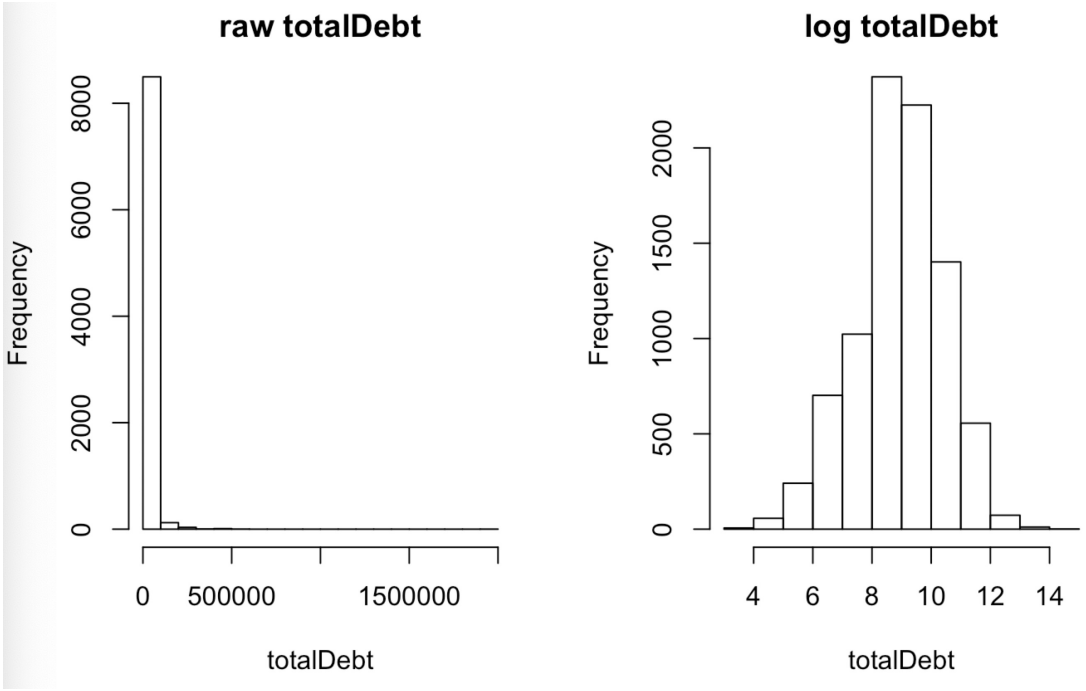

Figure 4: Correlation Plot

However, it is risky for us to assume it is a log-normal model, and to quote Geroge Box :"All models are wrong, some are useful". If log-normal is not the true distribution, log transformation would introduce more error. Further, if you transform the response variable, then you transform also the variance of you response variable. Thus, a log transformation will lead to the assumption that your variance is in fact log-normally distributed. (Christoph Scherber,2012). Generalized linear models are more flexible than transformations of the response, in that they allow a separate modeling of linearity and variance relationships. Thus in this case, we are choosing Gaussian GLM with log link instead of transformation response variable directly.

We continue our analysis based on imputed data set, with 8690 observations and 9 columns.

7.1.2 Outlier Detection

Gaussian GLM with log link was for 9 predictors was used to compute Leverage and Cook's distance plot (as shown in graph). We can observe several influential observations, with large leverage value (leverage over threshold $2p/n$, here p means

number of features, which is 9, and n means numbers of observations, which is 8690). However for cook's distance, there is only a few of high leverage value are actually considered as outliers. For the sake of variability in measurement, we are trying to keep as many observations as possible, and only delete those with both very high leverage and cook's distance value (> 0.02).

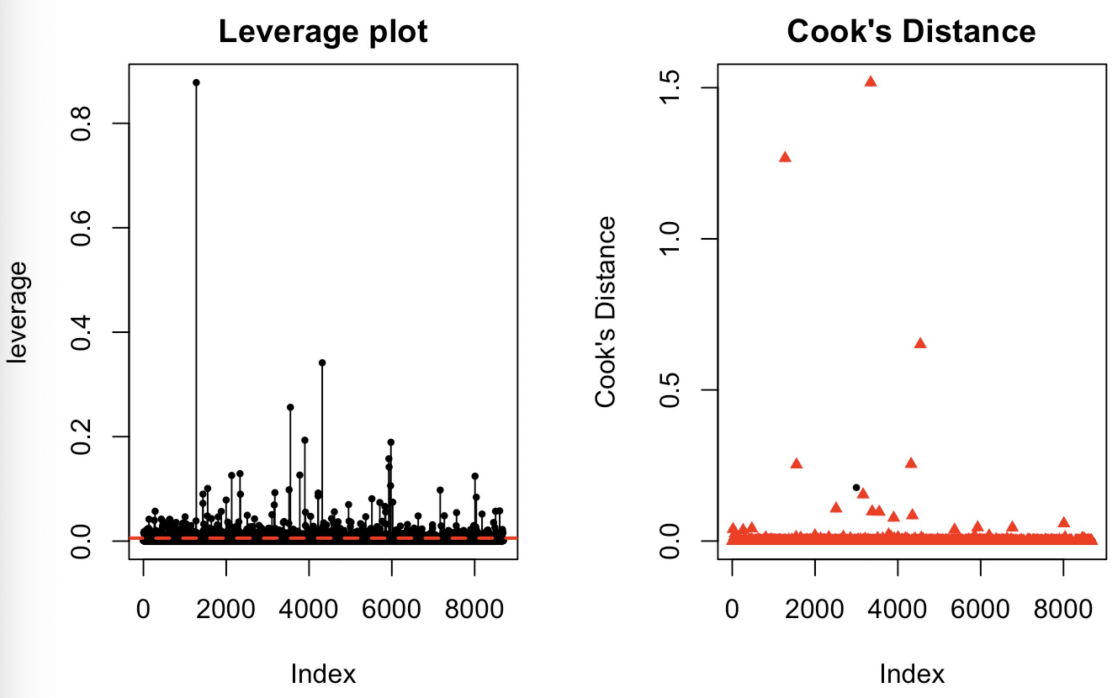

Figure 5: Outlier Detection

The remaining data set has 8678 observations with 9 features.

Below is a table with column information after imputation, data clean and removing outlier.

predictors	Description	Range
nFU	people in a family unit	1 - 12
ageHead	Age of husband	17 - 104
headMarital	husband Marital Status	Married,Never married,widowed,divorced,separated
lifeSat	Life Satisfaction	1 completely satisfied - 5 not at all satisfied,
totalDebt	Value all debts	42 - 2200000
educWife	education degree for wife	AA, BS, MS, PhD, JD, MD, honorary, other
eduHead	education degree for husband	AA, BS, MS, PhD, JD, MD, honorary, other
FamilyIncome	total family income for 2008	-100 - 6317099 (in USD)
eduExp	education expenditure	0 - 120000 (in USD)

Table 1: Column information after data clean

7.2 Modeling

7.2.1 GLM

Below is model formula for GLM after AIC procedure

$$log(E(totalDebt|X)) = \eta = \beta_0 + \beta_1 ageHead + \beta_{2-5} 1_{lifeSat} + \beta_{6-11} 1_{educHead} + \beta_{12} FamilyIncome + \beta_{13} eduExp \quad (2)$$

Below is Pearson and deviance residual Box plot and residual plot for GLM model (after AIC procedure)

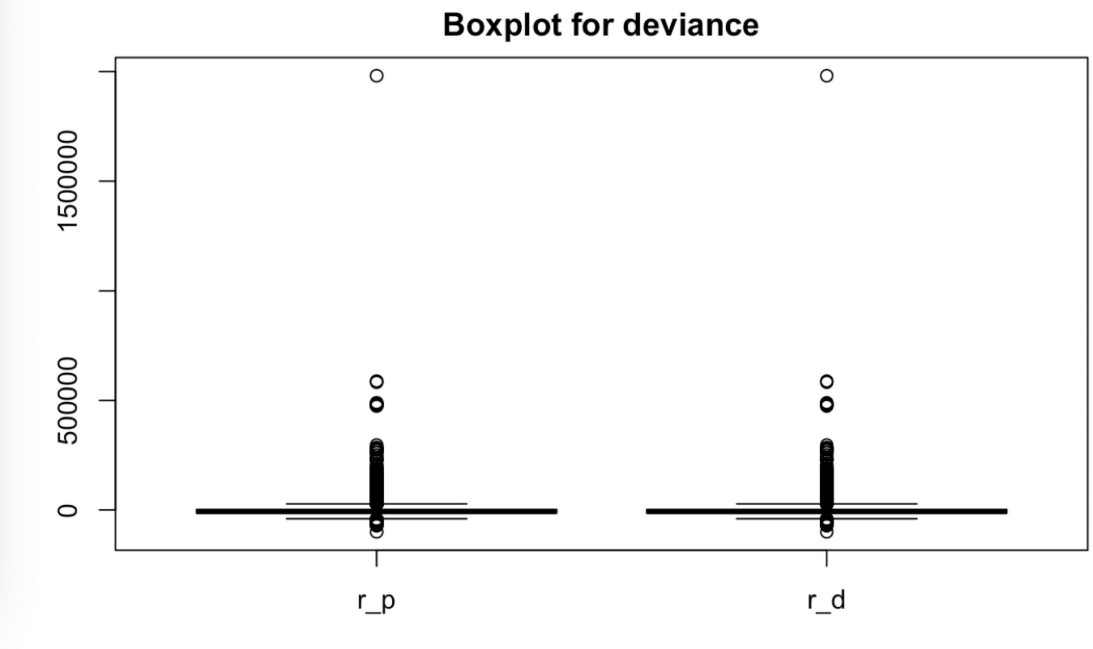

Figure 6: GLM AIC box

7.2.2 GLM

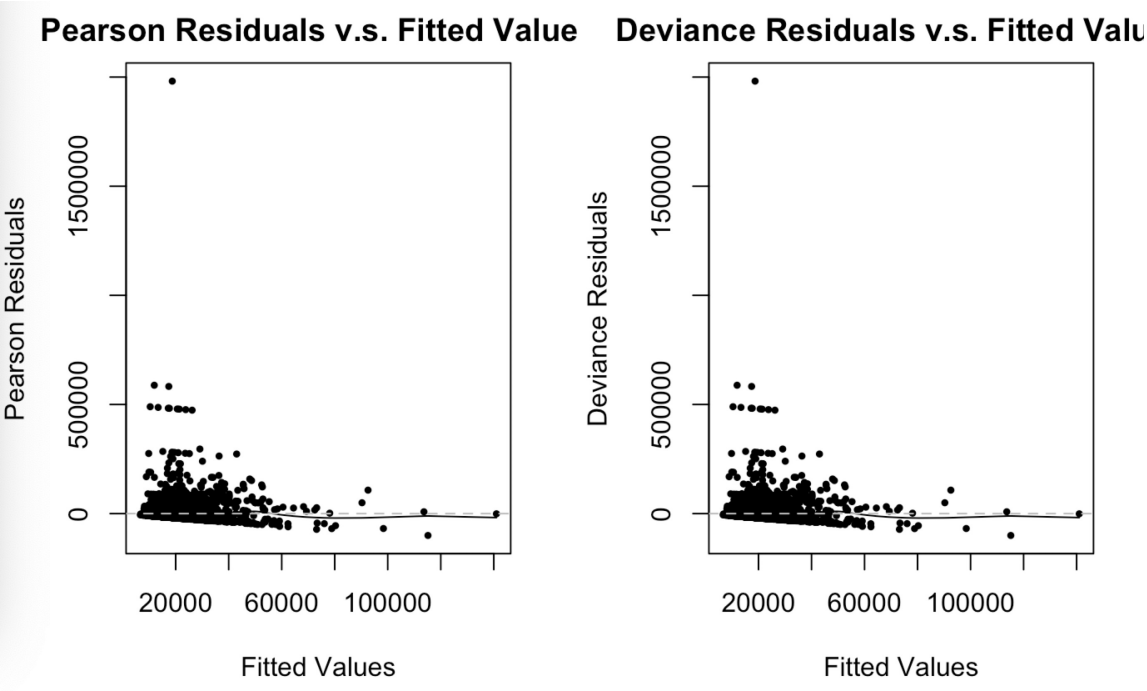

Figure 7: GLM AIC residual

Together with runs-test ($p = 0.08974 > 0.05$), we fail to see any evidence of lack of fit for this model, indicating that the model is good.

7.2.3 GAPLM

Notice for the plots of fitted function, we could see that some gaps exists in the range of those continuous variables, which can alter the results for smoothing in an influential way. The reason is that the minimum bandwidth must be large enough to cover the gap to give well defined estimates at the gaps, which may lead to over-smoothing for other regions. This is also the reason that we observe some sudden changes at the boundaries in some of the estimated functions. Since the boundary estimates may be not reliable, we focus our interpretations on the regions without gaps or boundary effects.
We ignore the unstable parts of fitted function caused by gaps within observed predictor values.

Below is GAM plot using gam package in R, using cubic smoothing. Notice only ageHead shows non-linear patter.

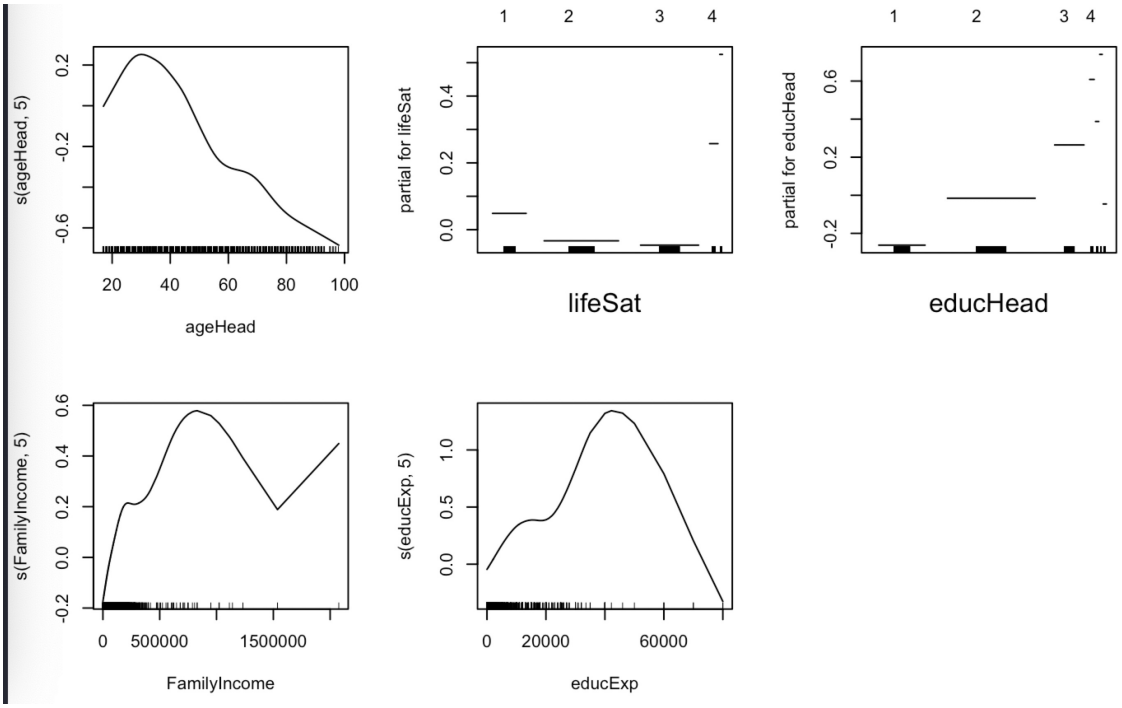

Figure 8: GAM cubic

Below is GAM plot using gam package in R, using local linear. Notice non of the continuous variable showed non-linear pattern.

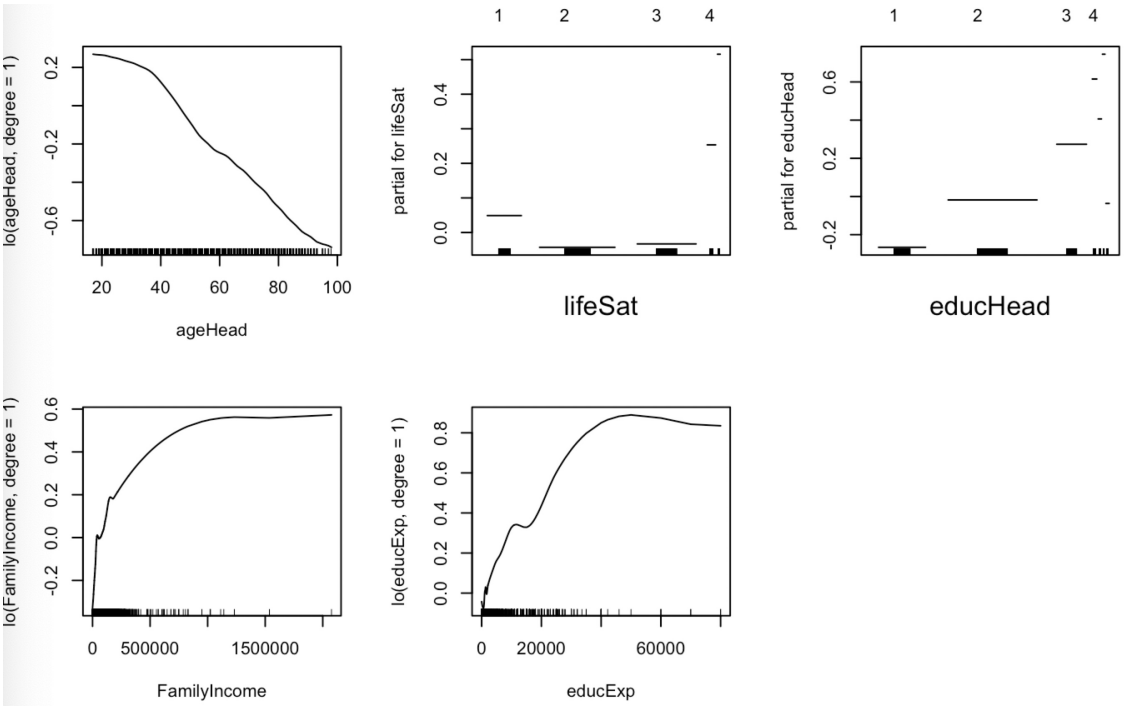

Figure 9: GAM local

Below is model formula for GAPLM cubic smoothing spline model

$$log(E(totalDebt|X)) = \eta = \beta_0 + f(ageHead) + \beta_{3-6}1_{lifeSat} + \beta_{7-12}1_{educHead} + \beta_{13}FamilyIncome + \beta_{14}eduExp \quad (3)$$

Below is Pearson and deviance residual Box plot and residual vs fitted plot for GAPLM cubic smoothing spline model

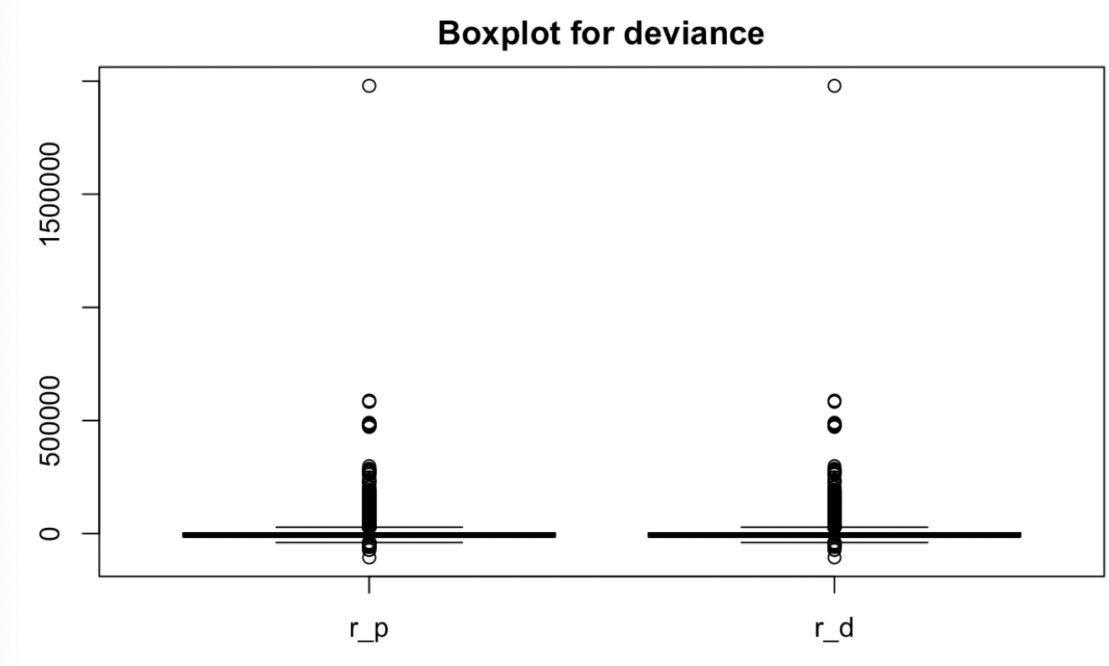

Figure 10: GAPLM cubic smoothing spline residual Box Plot

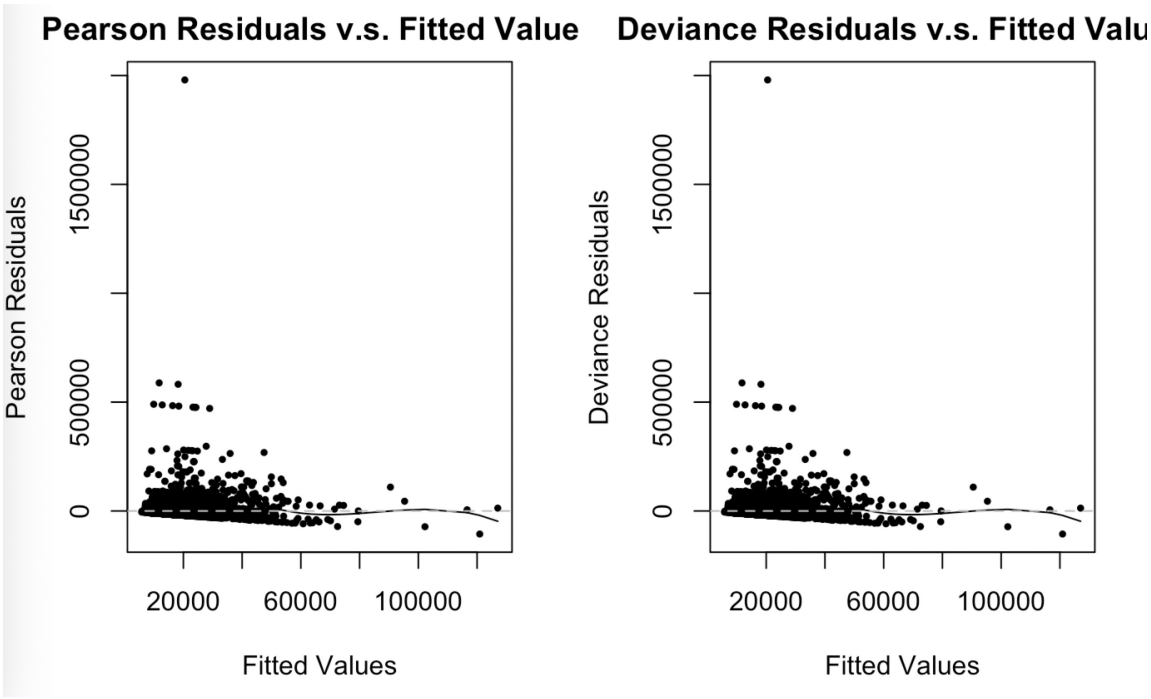

Figure 11: GAPLM cubic smoothing spline Residual vs Fitted Plot

Together with runs-test ($p = 0.337 > 0.05$), we fail to see any evidence of lack of fit for this model, indicating that the model is good.

7.2.4 Modified GLM with GAM observation

Below is model formula for modified GLM with GAM observation

$$log(E(totalDebt|X)) = \eta = \beta_0 + \beta_1 ageHead + \beta_2 ageHead^2 + \beta_{3-6} 1_{lifeSat} + \beta_{7-12} 1_{educHead} + \beta_{13} FamilyIncome + \beta_{14} eduExp + \beta_{15} ageHead * educExp + \beta_{16} FamilyIncome * educExp \quad (4)$$

Below is Pearson and deviance residual Box plot and residual v.s. fitted plot for GLM with quadratic and interact term.

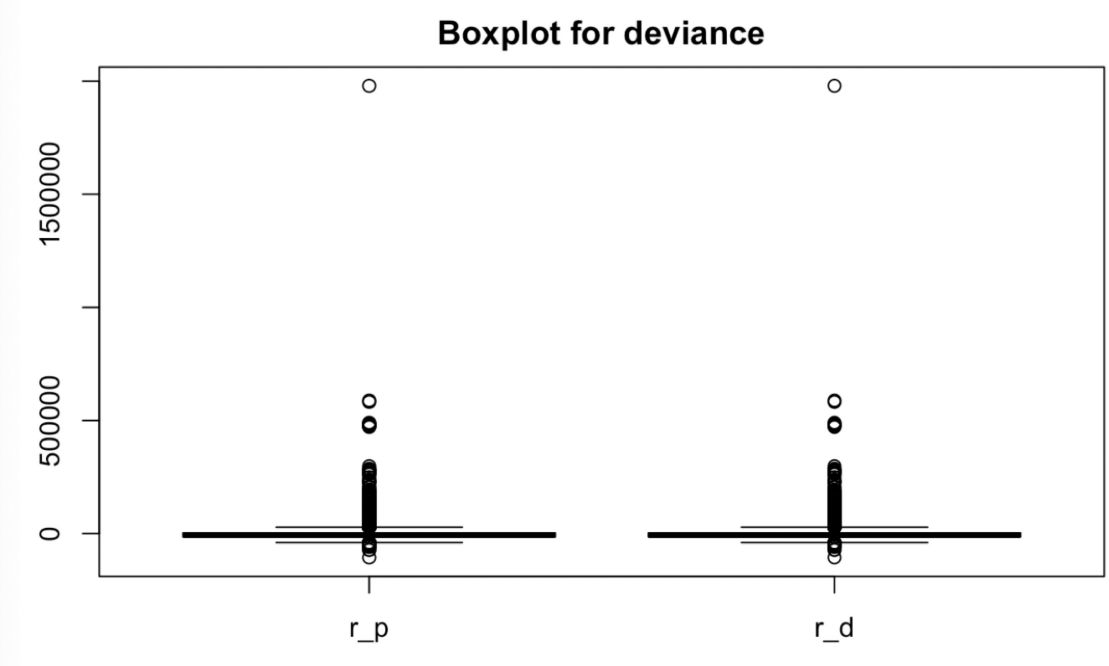

Figure 12: Modified GLM Box Plot

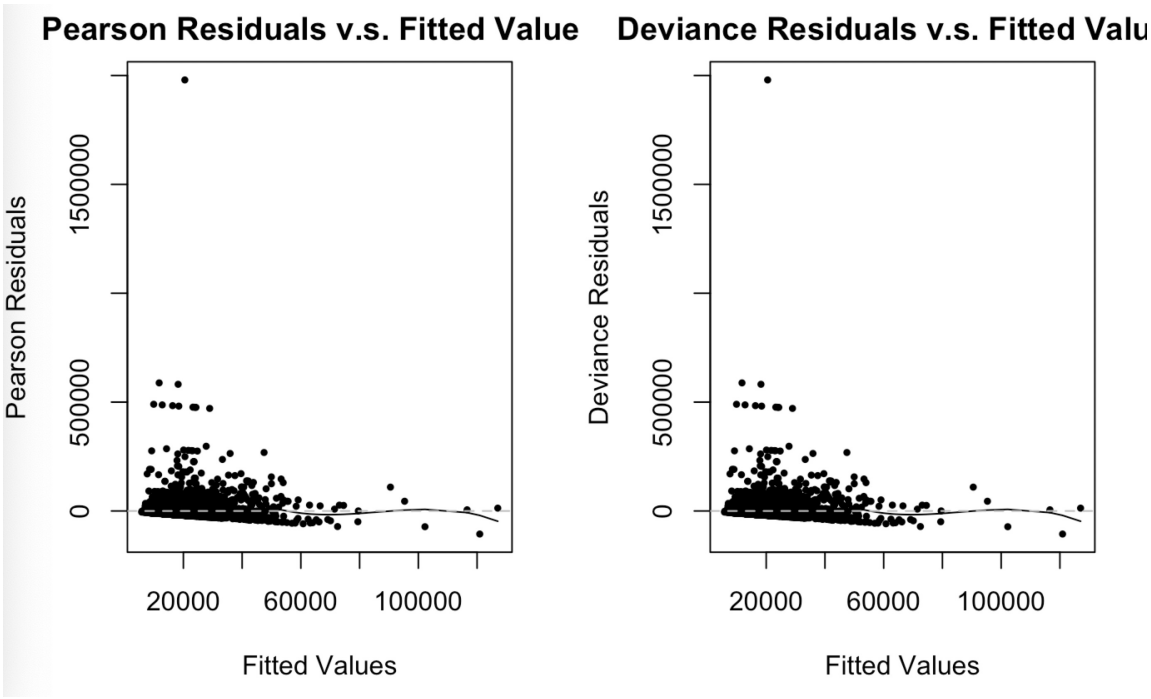

Figure 13: Modified GLM Residual vs Fitted Plot

Together with runs-test ($p = 0.07154 > 0.05$), we fail to see any evidence of lack of fit for this model, indicating that the model is good.

7.2.5 Machine Learning

Below is a graph represent visualization of decision tree.

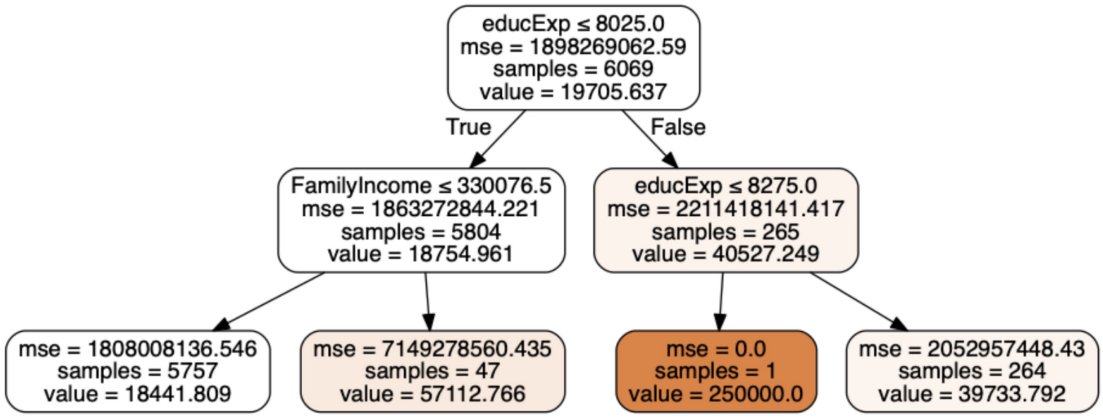

Figure 14: Decision Tree

From graph, we could observe that this decision tree is making decision based on educExp (education expenditure), FamilyIncome and educExp. Since we set tree depth as 2, there is only 2 feature decision layer and the 3rd layer is MSE (one of decision tree metric, selected when constructing tree) decision layer.

7.3 Discussion

A model with small MSE will serve as a good prediction model .

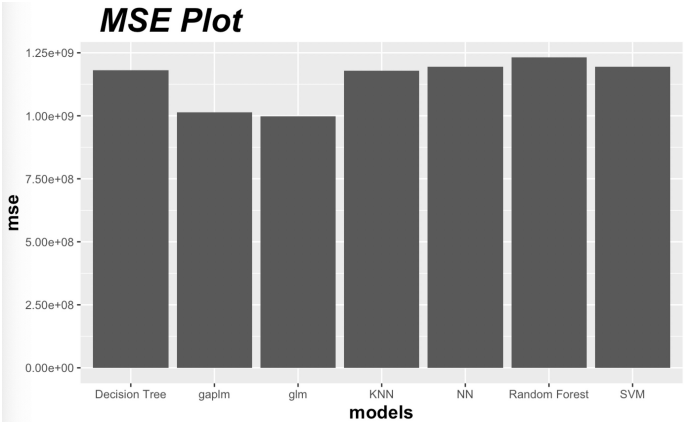

Figure 15: MSE Plot

The MSE plot was generated based on seven different prediction models. Based on this plot, we can see GLM model with quadratic and interact term has the smallest MSE.

Below is coefficients for parametric of GLM with quadratic and interaction term

	coefficient
β_0	9.436
$\beta_{ageHead}$	0.02018
$\beta_{ageHead^2}$	-0.000353
$\beta_{lifeSat=Verysatisfied}$	-0.08252
$\beta_{lifeSat=Somewhatsatisfied}$	-0.01437
$\beta_{lifeSat=Notverysatisfied}$	0.0987
$\beta_{lifeSat=Notatallsatisfied}$	0.376
$\beta_{educHead=Bachelor}$	0.2529
$\beta_{educHead=Master}$	0.5272
$\beta_{educHead=PhD}$	0.9084
$\beta_{educHead=JD;LLB}$	0.6752
$\beta_{educHead=MD;DDS;DVM;DO}$	1.012
$\beta_{educHead=Other}$	0.2236
$\beta_{FamilyIncome}$	0.0000009343
$\beta_{educExp}$	0.00005165
$\beta_{ageHead:educExp}$	-0.000000565
$\beta_{FamilyIncome:educExp}$	-0.00000000004917

Table 2: coefficient for parametric term

Below is a graph for $y = ax^2 + bx + c$ curve, with x = ageHead, a = -0.0003529, b = 0.02018, c = 9.436.

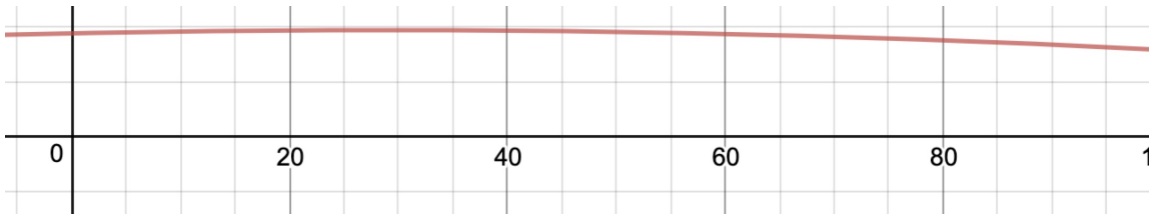

Figure 16: Curve for ageHead predictor

7.4 Code

7.4.1 R

```
#' ---
#' title: "STA 260 Final Report"
#' author: "Casey Liu"
#' date: "2/21/2019"
#' output: pdf_document
#' ---
#'
## ----setup, include=FALSE-----------------------------------------------------
knitr::opts_chunk$set(echo = TRUE)

#'
#'
#'
#' # preprocess
## -----------------------------------------------------------------------------
setwd("/Users/yilanliu/Desktop/UCD/sta260/final/")
mydata=read.table("J254627.csv",sep=",",header=TRUE)

colnames(mydata)=c("releaseNumber","familyID","nFU",
                   "ageHead","nChildren","headMarital",
                   "lifeSat","totalDebt","educWife",
                   "educHead","taxableIncome","FamilyIncome","educExp")
dim(mydata)

# all release number is 5, which means it's the most recent record with im

#mydata=mydata[(mydata$totalDebt<00000)&(mydata$totalDebt>0),]
mydata = mydata[,-c(1,2)]
# nFU column -- good
# age column 999 = NA
mydata$ageHead[mydata$ageHead==999]<-NA

# nChildren --- good ----- too many/ zero-inflation?
# headMarital ---- NA
mydata$headMarital[mydata$headMarital==9 | mydata$headMarital==8]<-NA
```

```
# LifeSat
mydata$lifeSat[mydata$lifeSat == 8 | mydata$lifeSat == 9 | mydata$lifeSat =
# totalDebt
mydata$totalDebt[mydata$totalDebt == 999999998 | mydata$totalDebt == 99999
# EduWife
mydata$educWife[mydata$educWife == 98 | mydata$educWife == 99 | mydata$edu
# EduHead
mydata$educHead[mydata$educHead == 98 | mydata$educHead == 99 | mydata$edu
# taxableIncome ---- dubious
# FamilyIncome ----dubious
# educExp --- okay

mydata$headMarital = as.factor(mydata$headMarital)
mydata$lifeSat = as.factor(mydata$lifeSat)
mydata$educWife = as.factor(mydata$educWife)
mydata$educHead = as.factor(mydata$educHead)

summary(mydata)
df = mydata


#'
#'
#' # data imputation
## ------------------------------------------------------------------------
library(readr)
library(mice)
#df = read.csv("debt_raw.csv")
#df = df[,c(-1)]
summary(df)
#colnames(df)=c("nFU","ageHead","nChildren","headMarital",
#                    "lifeSat","totalDebt","educWife",
#                    "educHead","taxableIncome","FamilyIncome","educExp")
library(VIM)
aggr_plot <- aggr(df, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE
                  , cex.axis=.7, gap=3, ylab=c("Histogram of missing data"

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

```
tempData <- mice(df,m=5,maxit=50,meth='pmm',seed=500)
stripplot(tempData, pch = 20, cex = 0.01)
completedData <- complete(tempData,1)
write.csv(completedData, "debt_raw.csv")
summary(completedData)
df = completedData

#'
#'
#' # analysis
## ------------------------------------------------------------------------
library(lawstat)
library(MASS)

assumption_check = function(model){
  par(mfrow=c(1,1))
  r_p = residuals(model, type="pearson")
  r_d = residuals(model, type="deviance")
  do.call(rbind, Map(data.frame, pearson_residual=r_p, deviance_residual=r
  boxplot(cbind(r_p, r_d), labels = c("Pearson residual", "Deviance residu

  y_hat = model$fitted.values
  par(mfrow=c(1,2))
  plot(y_hat, r_p, pch=16, cex=0.6, ylab='Pearson Residuals',
       xlab='Fitted Values',
       main = "Pearson Residuals v.s. Fitted Values")
  lines(smooth.spline(y_hat, r_p, spar=0.9))
  abline(h=0, lty=2, col='grey')
  plot(y_hat, r_d, pch=16, cex=0.6, ylab='Deviance Residuals',
       xlab='Fitted Values',
       main = "Deviance Residuals v.s. Fitted Values")
  lines(smooth.spline(y_hat, r_d, spar=0.9))
  abline(h=0, lty=2, col='grey')
  print(runs.test(y = r_p, plot.it = FALSE))
  print(runs.test(y = r_d, plot.it = FALSE))
}

outlier_check = function(model, dataset, standard_cooks){
```

```
  par(mfrow=c(1,2))
  # leverage points => influential points

  leverage = hatvalues(model)
  plot(names(leverage), leverage, xlab="Index", type="h", main = "Leverage
  points(names(leverage), leverage, pch=16, cex=0.6)
  p <- length(coef(model))
  n <- nrow(dataset)
  abline(h=2*p/n,col=2,lwd=2,lty=2)
  infPts <- which(leverage>2*p/n)

  cooks = cooks.distance(model)

  plot(cooks, ylab="Cook's Distance", pch=16, cex=0.6,main = "Cook's Dista
  points(infPts, cooks[infPts], pch=17, cex=0.8, col=2)
  #text(infPts, cooks[infPts],  cex= 0.7, pos = 2)
  infPts[cooks[infPts]>standard_cooks]
  #cooks[infPts]>standard_cooks
  #text(x=1:length(cooks)+1, y=cooks, labels=ifelse(cooks>standard_cooks,
# add labels
  #susPts <- as.numeric(names(sort(cooks[cooks>standard_cooks], decreasing=
  #print(susPts)
}

#'
#' a.) glm
## ------------------------------------------------------------------------
df = read.csv("debt_raw.csv")
df = df[,c(-1)]
summary(df)

df$headMarital = as.factor(df$headMarital)
df$lifeSat = as.factor(df$lifeSat)
df$educWife = as.factor(df$educWife)
df$educHead = as.factor(df$educHead)
continous_df = df[,c(1,2,3,6,9,10,11)]
summary(continous_df)
library(corrplot)
```

```
corrplot(cor(continous_df), type="lower", method = "number")
# notice nChildren and nFU has corr = 0.86, FamilyIncome and taxableincome
# thus we decide to delete nChildren and taxableIncome
df = df[,c(-3, -9)]
summary(df)
#corrplot(cor(df), type="lower", method = "number")
# there is still some correlation in data
# however to decide to continue computing our result based on this
# Note the response looks like a gamma distribution with lots of zeros

#'
#' # Data Transformation, histogram suggest to use log transformation
#' https://stats.stackexchange.com/questions/47840/linear-model-with-log-t
#' based on this, we decided to use log link in gaussian distribution, ins
## ------------------------------------------------------------------------
hist(df$totalDebt)
hist(log(df$totalDebt))
#df$totalDebt = log(log(df$totalDebt))


#'
#' # Outliers detection using Normal distribution
## ------------------------------------------------------------------------
summary(df)
continous_df = df[,c(1,2,5,8,9)]
plot(continous_df)

## ------------------------------------------------------------------------
summary(df)

#'
#'
## ------------------------------------------------------------------------
glm_1 = glm(formula = totalDebt~., family=gaussian(link = log), data = df)
assumption_check(glm_1) # runs = 0.1173
dim(df) # 8690    9
outlier_check(glm_1, df, 0.02)
# reomve those outliers
```

```
df = df[-c(20, 64, 263, 473, 1277, 1552, 2508, 3160, 3344, 3380, 3550, 377
dim(df) # 8678    9

glm_2 = glm(formula = totalDebt~., family=gaussian(link = log), data = df)
outlier_check(glm_2, df, 0.025)
assumption_check(glm_2) # runs =  0.01711
summary(glm_2)
write.csv(df, "debt.csv")

#'
#'
#' (Delete) Outliers detection using tweedie distribution
## ------------------------------------------------------------------------
#df = read.csv("debt.csv")
#df = df[,c(-1, -3, -9)]
library(statmod)
glm_1 = glm(formula = totalDebt~., family=tweedie(var.power=1.1, link.powe
assumption_check(glm_1)
dim(df) # 8690    9
outlier_check(glm_1, df, 0.005)
# thus observation 535, 197, 477, 309, 450, 478, 458, 197, 504 is influent
# reomve those outliers
df = df[-c(3068,4528,6925,7065,7418,7428,7698,7783,8316),]
dim(df) # 8681    9

glm_2 = glm(formula = totalDebt~., family=tweedie(var.power=1.1, link.powe
outlier_check(glm_2, df, 0.005)
assumption_check(glm_2)
summary(glm_2)
write.csv(completedData, "debt.csv")


#'
#' # split data into train and test
## ------------------------------------------------------------------------
df = read.csv("debt.csv")
df = df[,-c(1)]
```

```
df$headMarital = as.factor(df$headMarital)
df$lifeSat = as.factor(df$lifeSat)
df$educWife = as.factor(df$educWife)
df$educHead = as.factor(df$educHead)
summary(df)
require(caTools)
# https://rpubs.com/ID_Tech/S1
set.seed(123)   #  set seed to ensure you always have same random numbers
sample = sample.split(df,SplitRatio = 0.75) # splits the data in the ratio
train =subset(df,sample ==TRUE) # creates a training dataset named train1
test=subset(df, sample==FALSE)
length(train$totalDebt) # 5781
dim(train)
length(test$totalDebt) # 2889

#'
## ------------------------------------------------------------------------
summary(train$totalDebt)
summary(test$totalDebt)
hist(df$totalDebt)

#'
#'
#'
#' basic glm model
## ------------------------------------------------------------------------
glm_1 = glm(formula = totalDebt~., family=gaussian(link = log), data = tra
assumption_check(glm_1) # runs = 0.2867
summary(glm_1)
length(glm_1$coefficients)
stepAIC(glm_1)

# exclude nFu, ageHead, headMatital
glm_aic = glm(formula = totalDebt ~ ageHead + lifeSat + educHead + FamilyI
    educExp, family = gaussian(link = log), data = train)
assumption_check(glm_aic) # runs = p-value = 0.08974
summary(glm_aic)
```

```
#'
#'
#' gam::gam
#'
#' # check outliers for each parameter, individually.
## ------------------------------------------------------------------------
gam_cubic = gam::gam(totalDebt~ s(ageHead,5)+ lifeSat + educHead + s(Family
    s(educExp,5), family = gaussian(link = log), data = train )

outlier_check(gam_cubic, train, 0.05)
#plot(mydata)
par(mfrow=c(2,3))
plot(gam_cubic)

# thus include agehead as additive part

gaplm_1 = gam::gam(totalDebt~ s(ageHead,5)+ lifeSat + educHead + FamilyInco
    educExp, family = gaussian(link = log), data = train )
assumption_check(gaplm_1) # runs = 0.337
summary(gaplm_1)

#'
## ------------------------------------------------------------------------
library(gam)
gam_local <- gam::gam(totalDebt~ lo(ageHead,degree = 1)+ lifeSat + educHea
                      lo(FamilyIncome,degree = 1) +  lo(educExp,degree = 1
                    family = gaussian(link = log), data = train )
par(mfrow=c(2,3))
plot(gam_local)

#' # glm_aic2 include age^2
## ------------------------------------------------------------------------
glm_aic2 = glm(formula = totalDebt ~ ageHead + I(ageHead^2) + lifeSat + ed
    educExp, family = gaussian(link = log), data = train)
assumption_check(glm_aic2) # runs = p-value = 0.05649
summary(glm_aic2)

#'
```

```
#' # interaction check
## ------------------------------------------------------------------------
glm_aic2_interact= glm(formula = totalDebt ~ ageHead + I(ageHead^2)
                       + lifeSat + educHead + FamilyIncome + educExp
                       + (ageHead + FamilyIncome + educExp) * (ageHead + F
                       , family = gaussian(link = log), data = train)
assumption_check(glm_aic2_interact) # runs = 0.08022
summary(glm_aic2_interact)

# likelihood ratio test
# whether a reduced model is preferred
#testing the null hypothesis that the slope parameter for interaction term
#and the small p-value indicates that the null model is rejected.
#Therefore, the larger model is more appropriate.
anova( glm_aic2, glm_aic2_interact,test = "Chi")
stepAIC(glm_aic2_interact)

glm_interact = glm(formula = totalDebt ~ ageHead + I(ageHead^2) + lifeSat +
    educHead + FamilyIncome + educExp + ageHead:educExp + FamilyIncome:educ
    family = gaussian(link = log), data = train)
anova( glm_interact, glm_aic2_interact,test = "Chi") #0.2925 thus glm_inte
assumption_check(glm_interact) # runs = 0.07154

summary(glm_interact)

#'
#' # potential model: glm_interact, gaplm_1, compare mse using test set
#'
## ------------------------------------------------------------------------
library(caret)
prediction_check = function(model){
  pred_aic1 = predict(model, newdata = test, type="response")
   mean((test$totalDebt - pred_aic1)^2)
}
prediction_check(glm_interact) # 998,893,844 thus smaller
prediction_check(gaplm_1) # 1,014,720,020

#'
```

```
#'
#'
#'
```

7.4.2 Python

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


# Load libraries
import pandas as pd
from sklearn.model_selection import train_test_split # Import train_test_s
from sklearn import metrics #Import scikit-learn metrics module for accura


# In[6]:


df = pd.read_csv('debt.csv')
df = df.drop(["Unnamed: 0"],axis=1)
df.head()


# In[9]:


df['headMarital'] = df['headMarital'].astype('category')
df['lifeSat'] = df['lifeSat'].astype('category')
df['educWife'] = df['educWife'].astype('category')
df['educHead'] = df['educHead'].astype('category')
df = pd.get_dummies(df)


# In[10]:
```

```
df.head()


# In[11]:


X = df.drop(["totalDebt"],axis=1)
y =df[["totalDebt"]]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, r
# 70% training and 30% test
X_train.to_csv("X_train", sep='\t', index=False)
X_test.to_csv("X_test", sep='\t', index=False)
y_train.to_csv("y_train", sep='\t', index=False)
y_test.to_csv("y_test", sep='\t', index=False)


# In[ ]:


from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
mean_squared_error(y_true, y_pred)

y_true = [[0.5, 1], [-1, 1], [7, -6]]
y_pred = [[0, 2], [-1, 2], [8, -5]]
mean_squared_error(y_true, y_pred)


## Decision Tree

# In[18]:


#desicion trees
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
# Create Decision Tree classifer object
```

```
clf = DecisionTreeRegressor( max_depth=2)

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

mean_squared_error(y_test, y_pred) # 1,180,348,295.1599853


# In[19]:


from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus

dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = list(X),class_name
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
#graph


## Random Forest

# In[20]:


#Import Random Forest Model
from sklearn.ensemble import RandomForestRegressor

#Create a Gaussian Classifier
```

```
clf=RandomForestRegressor(n_estimators=100)

#Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train,y_train.values.ravel())

y_pred=clf.predict(X_test)

mean_squared_error(y_test, y_pred) # 1,231,921,022.4153643


## KNN

# In[28]:


from sklearn.neighbors import KNeighborsRegressor
#Create KNN Classifier
knn = KNeighborsRegressor(n_neighbors=100)

#Train the model using the training sets
knn.fit(X_train, y_train.values.ravel())

#Predict the response for test dataset
y_pred = knn.predict(X_test)

mean_squared_error(y_test, y_pred) #1,178,995,799


## Neural Network

# In[36]:


from sklearn.neural_network import MLPRegressor

#Create NN Classifier
nn = MLPRegressor(alpha=0.8, hidden_layer_sizes=(10,), max_iter = 1000)
```

```
#Train the model using the training sets
nn.fit(X_train, y_train.values.ravel())

#Predict the response for test dataset
y_pred = nn.predict(X_test)

mean_squared_error(y_test, y_pred) #1,194,438,189 not converge


## SVM

# In[37]:


#Import svm model
from sklearn.svm import SVR

#Create a svm Classifier
clf = SVR(kernel='linear') # Linear Kernel

#Train the model using the training sets
clf.fit(X_train, y_train.values.ravel())

#Predict the response for test dataset
y_pred = clf.predict(X_test)

mean_squared_error(y_test, y_pred) # 1,194,468,569


# In[ ]:
```