STAT 652 Final Project

Arin Ghosh

12/1/2018

1. Introduction

This is an analysis report of the data provided by the Canadian Community Health Survey (CCHS) – Healthy Aging module. The project is divided into 2 parts corresponding to two separate datasets provided to us. The first dataset has 20000 rows and 9 columns. In this dataset problem our task is to predict cognitive health index called HUIDCOG using 8 other health-utility-index (HUI) variables. In the second part of the project, we are tasked with building a regression model that predicts a real number called HUIDHSI, which is another measure of the HUI that provides a description of an individual's overall functional health. The second dataset is much bigger in size compared to the first dataset, having 590 variables with 10000 rows. A part of the dataset is held out for validation purpose which was released to the students at a later date.

Part 1: Predicting HUIDCOG (Classification Analysis)

1 Data

1.1 Data Loading

First step is to load the appropriate dataset into the R Studio environment. The dataset can be found on the project github repository. Once downloaded in to the working directory of the R Studio, we can load the data using read.csv() command.

```r
hui <- read.csv("hui.csv")
summary(hui)
```

```
##            DHHGAGE       DHH_SEX                 HUIDCOG
##  55 TO 59 YEARS:3085   FEMALE:11385   COG. ATT. LEVE 1:13949
##  60 TO 64 YEARS:2982   MALE  : 8615   COG. ATT. LEVE 2:  496
##  85 AND OLDER  :2602                  COG. ATT. LEVE 3: 3764
##  65 TO 69 YEARS:2595                  COG. ATT. LEVE 4: 1268
##  70 TO 74 YEARS:1958                  COG. ATT. LEVE 5:  429
##  75 TO 79 YEARS:1928                  COG. ATT. LEVE 6:   71
##  (Other)       :4850                  NOT STATED      :   23
##            HUIGDEX                 HUIDEMO                 HUIGHER
##  LIM. HANDS/F   :  252   EMOT. ATT. LEV.1:14912   NO PROBLEMS    :17335
##  NOT STATED     :   10   EMOT. ATT. LEV.2: 4067   NOT STATED     :  296
##  USE OF HANDS/F.:19738   EMOT. ATT. LEV.3:  749   PROB./CORR.    : 1579
##                          EMOT. ATT. LEV.4:  183   PROB./NOT CORR.:  790
##                          EMOT. ATT. LEV.5:   39
##                          NOT STATED      :   50
##
##            HUIGMOB                 HUIGSPE                 HUIGVIS
##  NEED MECH. SUPP: 1580   NO PROBLEMS     :19837   NO PROBLEMS     : 4210
##  NO AID REQUIRED:  322   NOT STATED      :   11   NOT STATED      :  142
##  NO PROBLEMS    :17496   PARTIAL/NOT UND.:  152   VISUAL P. UNCOR.:  658
##  NOT STATED     :   16                            VISUAL PROB. COR:14990
```

```
##  REQUIRES HELP  :  586
##
##
```

From the summary of the hui dataset, we can see that there are a total of 9 columns that exists. Further, we see from the summary that there are no NA or missing values, although there are several entries with value 'NOT STATED'. Also it is noteworthy to mention that none of the variables are continuous variables and the value which we are suppose to predict is a multivariate i.e. HUIDCOG can take one of the possible 7 values for each set of dependent variables. Although the dataset appear to be clean and variables are grouped, we will still need to do some analysis and take a look if we can reduce possible number of outcomes for each column without loosing a lot of quality in the dataset where it makes sense.

1.2 Exploratory Data Analysis & Data Grouping

Missing Data:

To start with our data analysis, we can notice that for some of the responses, we have an inconclusive response i.e. `hui$HUIDCOG== 'NOT STATED'`. We should first remove these records, since this leads to an observation where we don't know what is the outcome.

```
# remove NOT STATED from HUIDCOG
hui <- hui[hui$HUIDCOG!= 'NOT STATED',]
hui$HUIDCOG <- factor(hui$HUIDCOG)
```

Target Variabe:

The given dataset is a fully categorical mulltivariate dataset meaning there are no columns with real numbers. This is something that is not extinsively discussed in our class, although we've been tought how to deal with categorical data in general. Since multivariate analysis is not extensively covered in the coursework, I am going to reduce the possible outcome of our dependent variable HUIDCOG from 6 to 2. To come up with a meaningful division, we must refer to the original documentation provided by the instructor. If we take a look into page 53 of CCHS_HA_Derived_variables.pdf then we can find out how these 6 different classes of HUIDCOG came into existence. For our analysis purpose I have divided our target variable HUIDCOG into a binary response where 1 refers to the patient is healthy in terms of cognitive abilities and 0 is unhealthy. We assign 1 if the patient is able to think clearly and solve day to day problems (COG. ATT. LEVE 1) and assign a 0 otherwise since in any other case it indicates some kind of issues with the patient's cognitive health abilities.

```
hui$HUIDCOG = ifelse(hui$HUIDCOG=='COG. ATT. LEVE 1', 1, 0)
```

Dependent Variable Removal:

Next we are interested in seeing the different class distribution of the 8 dependent variables. Particularly the variables HUIGDEX and HUIGSPE.

```
## $HUIGDEX
## freq_dist
##    LIM. HANDS/F      NOT STATED USE OF HANDS/F.
##    0.0125644491    0.0001501727    0.9872853782
##
## $HUIGSPE
## freq_dist
##      NO PROBLEMS       NOT STATED PARTIAL/NOT UND.
##     0.9922410772     0.0002002303     0.0075586925
```

Both of these variables, in my opinion, lacks diversity and is biased heavily towards one class than the others, therefore I have decided not to include these 2 variables in my analysis.

```
hui = dplyr::select(hui, -c(HUIGDEX, HUIGSPE))
```

Dependent Variable Group Collapsing:

Furthermore, I have tried to reduce the number of groups for each of the remaining 6 variables to some degree where it makes sense. For example, instead of using 4 possible classes of HUIGHER, which classifies the respondents based on their hearing state, I have collapsed it into 2, marking HUIGHER as Good if there was a history of hearing problem (whether or not its corrected currently) or otherwise Bad there was no hearing complains ever for that particular patient. Below is a summary of the collapsing decisions that has been made in this analysis.

HUIDEMO : Emotional index

This variable classifies respondents based on emotional health status. The original record has 6 different levels based on different levels of emotional response. But we can reduce it to Happy or Unhappy based on broader definition. I have converted all of NOT STATED as Unhappy.

Table 1: Mapping Table of HUIDEMO

HUIDEMO	isHappy
EMOT. ATT. LEV.1	Happy
EMOT. ATT. LEV.2	Happy
EMOT. ATT. LEV.3	Unhappy
EMOT. ATT. LEV.4	Unhappy
EMOT. ATT. LEV.5	Unhappy
NOT STATED	Unhappy

HUIGHER : Hearing State

This variable classifies respondents based on hearing state of the patient. As explained earlier, the original 4 possible classes are reduced to a broader 2 general classes of Good or Bad indicating if the patient has a history of hearing issues.

Table 2: Mapping Table of HUIGHER

HUIGHER	hearingState
NO PROBLEMS	Good
NOT STATED	Bad
PROB./CORR.	Good
PROB./NOT CORR.	Bad

HUIGMOB : Mobility Trouble

This variable classifies the respondents based on their state of mobility trouble. We classify this as TRUE or FALSE indicating if the respondent indicated that (s)he cannot move freely without external help.

Table 3: Mapping Table of HUIGMOB

HUIGMOB	mobilityHelp
NEED MECH. SUPP	TRUE
NO AID REQUIRED	FALSE
NO PROBLEMS	FALSE
NOT STATED	FALSE
REQUIRES HELP	TRUE

HUIGVIS : Vision State

This variable classifies the respondents based on their vision state. Like HUIGMOB, I have mapped this to TRUE if the respondent has a history of vision problem and False otherwise.

Table 4: Mapping Table of HUIGVIS

HUIGVIS	visualProb
NO PROBLEMS	TRUE
NOT STATED	TRUE
VISUAL P. UNCOR.	FALSE
VISUAL PROB. COR	FALSE

DHHGAGE : Age

Instead of age groups, I have take the mean age of the group, although since I am not reducing the number of classes here, it is probably not going to add a lot of value in the complexity reduction of our final model, unless we decide not to use this variable.

Table 5: Mapping Table of DHHGAGE

DHHGAGE	meanAges
45 TO 49 YEARS	47
50 TO 54 YEARS	52
55 TO 59 YEARS	57
60 TO 64 YEARS	62
65 TO 69 YEARS	67
70 TO 74 YEARS	72
75 TO 79 YEARS	77
80 TO 84 YEARS	82
85 AND OLDER	87

Here is the glance of the final dataset after data cleaning that we will be using in our model building.

Table 6: Final Dataset To Be Used For Model Building

DHH_SEX	HUIDCOG	isHappy	hearingState	mobilityHelp	visualProb	meanAges
MALE	1	Happy	Good	FALSE	TRUE	47
MALE	1	Happy	Good	FALSE	TRUE	47
FEMALE	1	Happy	Good	FALSE	FALSE	47
MALE	1	Happy	Good	FALSE	TRUE	47

DHH_SEX	HUIDCOG	isHappy	hearingState	mobilityHelp	visualProb	meanAges
MALE	1	Happy	Good	FALSE	FALSE	47
FEMALE	1	Happy	Good	FALSE	TRUE	47

```
##    DHH_SEX         HUIDCOG          isHappy      hearingState
##  FEMALE:11372   Min.   :0.0000   Happy  :18970   Bad : 1073
##  MALE  : 8605   1st Qu.:0.0000   Unhappy: 1007   Good:18904
##                 Median :1.0000
##                 Mean   :0.6983
##                 3rd Qu.:1.0000
##                 Max.   :1.0000
##  mobilityHelp    visualProb         meanAges
##  Mode :logical   Mode :logical   Min.   :47.00
##  FALSE:17816     FALSE:15638     1st Qu.:57.00
##  TRUE :2161      TRUE :4339      Median :67.00
##  NA's :0         NA's :0         Mean   :66.96
##                                  3rd Qu.:77.00
##                                  Max.   :87.00
```

2 Methods

In this part of the project, I have used some of the classification techniques that were taught over the course from Logistic Regression to Support Vector Machines. However for focussing on one technique, **Logistic Regression** is preferred which is interpretable and gives a low missclassification error rate as well as decent Specificity and Sensitivity score. I have used the variable importance table from random forest models to select a subgroup of variables to further tune my models. I have attached the details into the **Appendix** section.

2.1 Logistic Regression

Logistic Regression uses the logistic function fitted by **maximum likelihood**. It performs well even if the predictors do not follow Gaussian distribution. The model is a linear model in the log-odds of success

$$\log\left(\frac{p(X)}{1-p(X)}\right) = X\beta = \beta_0 + X_1\beta_1 + \ldots + X_p\beta_p.$$

Since our dependent variable takes a 0/1 binary response, we can use this model. Unlike linear regression where one unit change in the predictor variable (X) results in one unit change in Y, here one unit increase in X_j , while holding all others fixed is associated with a β_j change in the log-odds.

Let's start with baseline model in logistic regression. Before fitting the model, dataset is split into training and testing set in random sampled fashion of ratio 70:30. The model is fitted to train data and then predict the test data to validate based on its accuracy, sensitivity, specificity, etc.

The coefficients must be estimated based on the available training data. For logistic regression, the more general method of maximum likelihood is preferred for its robust statistical properties. Basically, the algorithm tries to find coefficients that maximize the likelihood that the probabilities are closest to 1 for people who don't have any problem in terms of patient's cognitive abilities (i.e. the respondent is able to think clearly and can solve day to day problems), and close to zero for people who has some type of cognitive disability and cannot carry out their day-to-day activity without some degree of help. During my experiment I have found that isHappy, mobilityHelp, hearingState, and meanAges are the most important variables so I have included only these 4 variables in my final model. Below table shows summary of estimates, Std.Error and p-value in the order of significance after performing logistic regression to the training data.

Table 7: Summary of Final Logistic Regression and its odd-ratios

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.58051491889384	0.11923118860264	13.2558849527299	4.17e-40
isHappyUnhappy	-1.25415543187966	0.0682410074257045	-18.3783252796362	1.96e-75
mobilityHelpTRUE	-0.524639323044865	0.0506502883766866	-10.3580717871361	3.85e-25
hearingStateGood	0.577738692439939	0.0661056790755542	8.73962268475639	2.34e-18
meanAges	-0.0169982650062456	0.00137517800999301	-12.3607743017445	4.26e-35

And here is the confusion matrix for the model indicating various measurements including accuracy, its 95% confidence interval, sensitivity, specificity and balanced accuracy as well.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0  1607  4421
##          1  1608 12341
##
##                Accuracy : 0.6982
##                  95% CI : (0.6918, 0.7046)
##     No Information Rate : 0.8391
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1744
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.49984
##             Specificity : 0.73625
##          Pos Pred Value : 0.26659
##          Neg Pred Value : 0.88472
##              Prevalence : 0.16094
##          Detection Rate : 0.08044
##    Detection Prevalence : 0.30175
##       Balanced Accuracy : 0.61805
##
##        'Positive' Class : 0
##
```

3. Results

3.1 Model Interpretation

On examining the fitted logistic regression model summary above, we can see that all the predictors are **statistically significant** with p-values far less than required 5%. These results also concur with the findings that predictors such as HUIGDEX, HUIGSPE, and DHH_SEX are not very significant predictors in determining a respondents cognitive health state. When we compare the full model of all available variables *(DHH_SEX + meanAges + isHappy + hearingState + mobilityHelp + visualProb)* with that of a model build on the subset of 4 predictors ($isHappy + mobilityHelp + hearingState + meanAges$) using Anova chisq test, a small p-value ($7.693e-08$) < 0.05 indicated that both models are similar - thus parsimoniously we chose the smaller model. Also, we have checked with various available logistic regression model selection techniques such as best subset, forward, backward and step-wise selection, all of them pointed to the four predictors that was used in the final model build. Therefore the final model that we've selected for predicting **HUIDCOG** is:

$$y_i = \beta_0 + isHappy * \beta_1 + mobilityHelp * \beta_2 + hearingState * \beta_3 + meanAges * \beta_4$$

We can interprete the model in this way: If a respondent reported that (s)he requires mobility help == 1, keeping other predictor unchanged, that can be associated with an estimated increase of (-0.53) units in the log-odds of the respondent being cognitively healthy. We can see from the Table 7 that it is estimated that if the respondent is classified as happy or need no mobility help or has a good hearing state or is younger (selecting any one of these predictor while not changing the others) it generally is an indication of the respondent is cognitively healthy, which makes sense. Now we can not only point out which predictor is associated with diminishing cognitive ability but we can also indicate more relevant statistics which is by how much units they affect the mental state of the patients.

3.2 Model Evaluation

Logistic regression model is used to predict the test data as well as the validation dataset that was released later to the students. We have used various statistical measures to measure the effectiveness of this model such as missclassification rate, sensitivity, specificity and Area under the ROC curve. We have also tried k fold cross validation on the training data set. Below is a graph that shows the ROC curve plot that is derived on the test dataset. The value of Area under the ROC curve we got is ~0.61. Based on various cut-off values, we found a cut-off of 0.5 leads to the best balance of accuracy, sensitivity and specificity. Please refer to the Appendix for further details on this section's derivations.

3.3 Comparisons of Classification Models

This sections shows the various models that we tried along with Logistic Regression model that was ultimately selected. The classification was model was build on 70% of the available data and 30% of the data was used for testing. Different metric that are used to compare the models are discussed in the following:

Misclassification Error: The number of observations that were predicted wrongly by the model. It is the proportion of misclassified observations.

Sensitivity: It is the ability of a model to correctly identify those with diabetic disease. It is observed True positive rate. TP/(TP+FN) where TP is True Positive and FN is False Negative.

Specificity: It is the ability of a model to correctly identify those without diabetic disease. It is observed True negative rate. TN/(TN+FP) where TP is True Negative and FP is False Positive.

Table 8: Model Comparison - Predicting HUIDCOG

method	accuracy	specificity	sensitivity
Logistic Regression	0.7003333	0.7383215	0.4978903
LDA	0.7088333	0.7215715	0.5521064
QDA	0.6938333	0.7420805	0.4805781
Random Forest	0.7070000	0.7181818	0.5461538

Evaluation of best model on the Holdout dataset

Once I finalized the model, I have re-fit the model on the full dataset (without any tran-test split) and then used the holdout dataset to measure the accuracy, sensitivity & specificity which I got is :

- accuracy : 0. 70

- sensitivity : 0.502
- specificity : 0.729

Conclusion and Discussion

After going through all the models, we have picked the logistic regression model with four predictors as our final selected model. However we have only considered linear terms in this model, which can be a drawback in this model. Therefore in the future it will be noteworthy to check how adding interaction and non-linear terms changes the predictive ability of the logistic regression model, or if there are any other family of classification techniques stands out when these new terms are introduced. We can try for even more powerful models such as deep neural networks to see if that helps us to give a better result. Finally all these discussions are based on the dataset that was provided to us. The correctness of this model will change as new data are available to us in the future, so that we might have to tune it later.

Appendix 1 (For Part 1)

This section mainly deals with the code base that was used for the analysis part written for part 1.

Software Version:

- All analysis on this project was done using R Studio - version 1.1.456 – © 2009-2018 RStudio, Inc.
- OS - Mac OS v 10.14.1 (18B75)

Data Loading:

Load all the required packages and some helper functions

```
library(dplyr, quietly = T)
library(caret, quietly = T)
library(MASS, quietly = T)
library(pROC, quietly = T)
library(randomForest, quietly = T)
library(leaps, quietly = T)
library(caret, quietly = T)
library(gam, quietly = T)

getCMMeasurements <- function(cm){
  accuracy = cm$overall[[1]]
  sensitivity = cm$byClass[[1]]
  specificity = cm$byClass[[2]]
  result <- t(as.data.frame(c(accuracy, sensitivity, specificity)))
  colnames(result) <- c('accuracy', 'sensitivity', 'specificity')
  rownames(result) <- NULL
  return(result)
}
```

Read the csv file and get a summary of the data

```
hui <- read.csv("hui.csv")
summary(hui)
```

```
##            DHHGAGE       DHH_SEX                  HUIDCOG
##  55 TO 59 YEARS:3085   FEMALE:11385   COG. ATT. LEVE 1:13949
##  60 TO 64 YEARS:2982   MALE  : 8615   COG. ATT. LEVE 2:  496
##  85 AND OLDER  :2602                  COG. ATT. LEVE 3: 3764
##  65 TO 69 YEARS:2595                  COG. ATT. LEVE 4: 1268
##  70 TO 74 YEARS:1958                  COG. ATT. LEVE 5:  429
##  75 TO 79 YEARS:1928                  COG. ATT. LEVE 6:   71
##  (Other)       :4850                  NOT STATED      :   23
##             HUIGDEX                  HUIDEMO                 HUIGHER
##  LIM. HANDS/F   :  252   EMOT. ATT. LEV.1:14912   NO PROBLEMS    :17335
##  NOT STATED     :   10   EMOT. ATT. LEV.2: 4067   NOT STATED     :  296
##  USE OF HANDS/F.:19738   EMOT. ATT. LEV.3:  749   PROB./CORR.    : 1579
##                          EMOT. ATT. LEV.4:  183   PROB./NOT CORR.:  790
##                          EMOT. ATT. LEV.5:   39
##                          NOT STATED      :   50
```

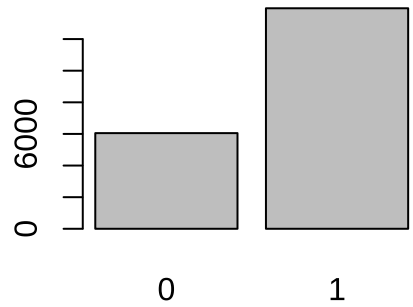


Figure 1: distribution of target variable

```
##
##            HUIGMOB                    HUIGSPE                    HUIGVIS
##  NEED MECH. SUPP: 1580   NO PROBLEMS      :19837   NO PROBLEMS      : 4210
##  NO AID REQUIRED:  322   NOT STATED       :    11   NOT STATED       :  142
##  NO PROBLEMS    :17496   PARTIAL/NOT UND.:   152   VISUAL P. UNCOR.:  658
##  NOT STATED     :   16                             VISUAL PROB. COR:14990
##  REQUIRES HELP  :  586
##
##
```

We are going to filter our target classes into 2 classes, based on the COG. ATT. LEVE value for HUIDCOG column. We say 1 if the respondent is doing alright, and 0 if there's some risk of imperfect cognitive attention level.

```
hui$HUIDCOG = as.factor(ifelse(hui$HUIDCOG=='COG. ATT. LEVE 1', 1, 0))
plot(hui$HUIDCOG)
```

Data Cleaning

We first need to remove all the incomplete responses.

```
# remove NOT STATED
hui <- hui[hui$HUIDCOG!= 'NOT STATED',]
hui$HUIDCOG <- factor(hui$HUIDCOG)
dim(hui)
```

```
## [1] 20000     9
```

Next we are going to go through each of the available independent variable columns to see if we can further group them down. The details of every variable:

- DHHGAGE - This variable indicates the age of the selected respondent. (p29)
- DHH_SEX - sex of the patient
- HUIDCOG - Cognition (Function Code) This variable classifies respondents based on cognitive health status. (p53)

- HUIGDEX - This variable classifies the respondents based on their state of dexterity trouble. (p55)
- HUIDEMO - This variable classifies respondents based on emotional health status. (p55)
- HUIGHER - This variable classifies the respondents based on their hearing state. (p57)
- HUIGMOB - This variable classifies the respondents based on their state of mobility trouble. (p59)
- HUIGSPE - This variable classifies the respondents based on their state of speech trouble. (p60)
- HUIGVIS - This variable classifies the respondents based on their vision state. (p62)

Age: We are replacing each with mean of the age class

```
lookup = data.frame(DHHGAGE=levels(hui$DHHGAGE),
                    meanAges=c(47, 52, 57, 62, 67, 72, 77, 82, 87))
hui = dplyr::select(merge(hui, lookup, by="DHHGAGE"), -DHHGAGE)

rm(lookup)
```

Emotional index, happy/not happy

```
lookup = data.frame(HUIDEMO=levels(hui$HUIDEMO),
                    isHappy=c('Happy', 'Happy', 'Unhappy','Unhappy','Unhappy','Unhappy'))
hui = dplyr::select(merge(hui, lookup, by="HUIDEMO"), -HUIDEMO)
```

Hearing State

```
lookup = data.frame(HUIGHER=levels(hui$HUIGHER),
                    hearingState=c('Good', 'Bad', 'Good','Bad'))
hui = dplyr::select(merge(hui, lookup, by="HUIGHER"), -HUIGHER)
```

Mobility Trouble

```
lookup = data.frame(HUIGMOB=levels(hui$HUIGMOB),
                    mobilityHelp=c(T, F, F, F,T))
hui = dplyr::select(merge(hui, lookup, by="HUIGMOB"), -HUIGMOB)
```

Vision State

```
lookup = data.frame(HUIGVIS=levels(hui$HUIGVIS),
                    visualProb=c(T, T, F, F))
hui = dplyr::select(merge(hui, lookup, by="HUIGVIS"), -HUIGVIS)
rm(lookup)
```

We can also see that both HUIGDEX and HUIGSPE are highly biased towards one particular class

```
table(hui$HUIGDEX)
```

```
##
##    LIM. HANDS/F      NOT STATED USE OF HANDS/F.
##             252              10           19738
```

```
table(hui$HUIGSPE)
```

```
##
##      NO PROBLEMS       NOT STATED PARTIAL/NOT UND.
##            19837               11             152
```

Therefore we remove these columns with high bias from our dataset:

```
hui = dplyr::select(hui, -c(HUIGDEX, HUIGSPE))
```

The final dataset looks like this:

```
summary(hui)
```

```
##    DHH_SEX      HUIDCOG      meanAges        isHappy      hearingState
##  FEMALE:11385   0: 6051   Min.   :47.00   Happy  :18979   Bad : 1086
##  MALE  : 8615   1:13949   1st Qu.:57.00   Unhappy: 1021   Good:18914
##                           Median :67.00
##                           Mean   :66.96
##                           3rd Qu.:77.00
##                           Max.   :87.00
##  mobilityHelp    visualProb
##  Mode :logical   Mode :logical
##  FALSE:17834     FALSE:15648
##  TRUE :2166      TRUE :4352
##  NA's :0         NA's :0
##
##
```

Train Test Split

The dataset is split into 70/30 ratio

```
## 70% of the sample size
smp_size <- floor(0.70 * nrow(hui))

## set the seed to make your partition reproducible
set.seed(19)
train_ind <- sample(seq_len(nrow(hui)), size = smp_size)

hui.train <- hui[train_ind, ]
hui.test <- hui[-train_ind, ]

rm(smp_size)
```

Model Fitting & Analysis

Logistic Regression without any subset selection on the test dataset

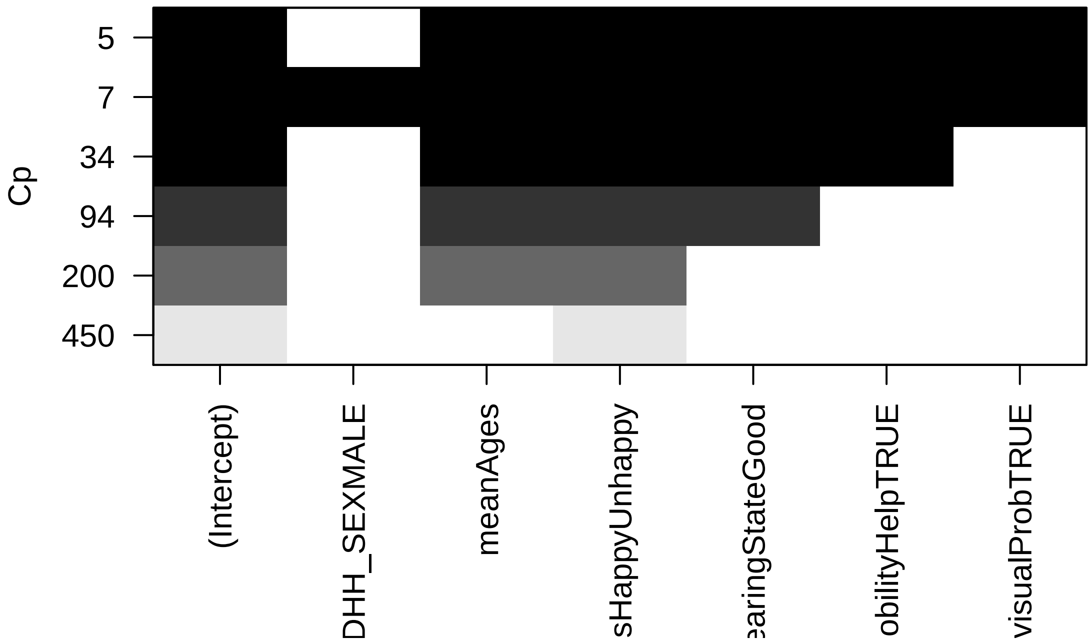
```r
full.model.glm <- glm(HUIDCOG ~ ., data=hui.train, family = binomial())
summary(full.model.glm)
```

```
##
## Call:
## glm(formula = HUIDCOG ~ ., family = binomial(), data = hui.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8777  -1.2533   0.7479   0.8069   2.0037
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.410326   0.145664   9.682  < 2e-16 ***
## DHH_SEXMALE       0.001772   0.038612   0.046    0.963
## meanAges         -0.017437   0.001648 -10.580  < 2e-16 ***
## isHappyUnhappy   -1.326871   0.082034 -16.175  < 2e-16 ***
## hearingStateGood  0.712945   0.079692   8.946  < 2e-16 ***
## mobilityHelpTRUE -0.429631   0.061055  -7.037 1.97e-12 ***
## visualProbTRUE    0.269040   0.047676   5.643 1.67e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 17200  on 13999  degrees of freedom
## Residual deviance: 16439  on 13993  degrees of freedom
## AIC: 16453
##
## Number of Fisher Scoring iterations: 4
```

Other than DHH_SEXMALE, every other predictor looks significant.

Subset Selection using forward/backward and stepwise

```r
cfits.fwd <- regsubsets(HUIDCOG ~ .,data=hui.train, method="forward")
plot(cfits.fwd,scale="Cp")
```




```r
rm(cfits.fwd)
```

Forward subset selection method concurs the finding in the Logistic Regression summary above, DHH_SEXMALE is still something we can get rid of. Also We get the same kind of result using the backward and seqrep method as well, so its not included.

Based on these analysis, we are going with the following reduced subset of independent variables - isHappy, mobilityHelp, hearingState, meanAges & visualProb.

```r
final.model.FUN <- (HUIDCOG ~ isHappy + mobilityHelp + hearingState + meanAges + visualProb)
```

We can now build our first logistic regression model

```r
small.model.glm <- glm(final.model.FUN,data=hui.train, family = binomial())
```

Is this model any better than the full model that we had earlier?

```r
# Anova Test will tell us
anova(full.model.glm, small.model.glm, test ="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: HUIDCOG ~ DHH_SEX + meanAges + isHappy + hearingState + mobilityHelp +
##     visualProb
## Model 2: HUIDCOG ~ isHappy + mobilityHelp + hearingState + meanAges +
##     visualProb
##   Resid. Df Resid. Dev Df   Deviance Pr(>Chi)
## 1     13993      16439
## 2     13994      16439 -1 -0.0021067   0.9634
```

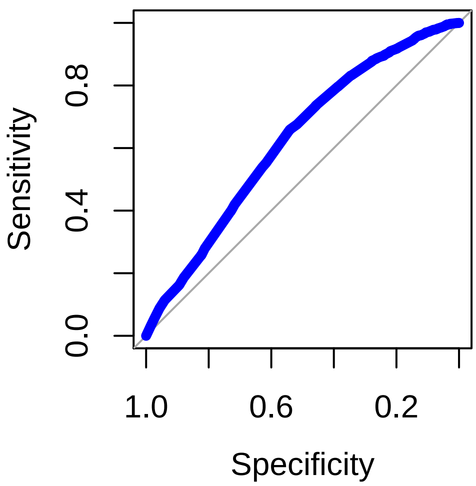
We can accept the null hypothesis that these 2 models are the basically the same. But going by the parsimony, we will choose the lighter model.

So we can now do some prediction and see how good our model is

```
pred <- predict(small.model.glm, hui.test)
pred_class <- ifelse(pred > 0.5, 1, 0)
roc.curve <- roc(hui.test$HUIDCOG, pred, direction="<")
print(roc.curve)
```

```
##
## Call:
## roc.default(response = hui.test$HUIDCOG, predictor = pred, direction = "<")
##
## Data: pred in 1794 controls (hui.test$HUIDCOG 0) < 4206 cases (hui.test$HUIDCOG 1).
## Area under the curve: 0.6226
```

```
plot(roc.curve,col="blue", lwd=5)
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  472 1322
##          1  476 3730
##
##                Accuracy : 0.7003
##                  95% CI : (0.6886, 0.7119)
##     No Information Rate : 0.842
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1734
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.49789
##             Specificity : 0.73832
##          Pos Pred Value : 0.26310
```

```
##          Neg Pred Value : 0.88683
##              Prevalence : 0.15800
##          Detection Rate : 0.07867
##    Detection Prevalence : 0.29900
##       Balanced Accuracy : 0.61811
##
##        'Positive' Class : 0
##
```

Table 9: Logistic Regression Metrics

accuracy	sensitivity	specificity
0.7003333	0.4978903	0.7383215

Based on this we can see that our accuracy for this model is 70.03% while the calculated sensitivity is 49% & the specificity is 73%. Our Area under the ROC curve is 62.26%

Does KFold Cross Validation makes our prediction better? I have written a small function following the lecture notes to implement it. Here we do a 10 fold cross validation.

```r
set.seed(19)
k_fold <- 10
cv.err <- rep(NA,k_fold)
cv.sen <- rep(NA,k_fold)
cv.spec <- rep(NA,k_fold)

#Randomly shuffle the data
hui<-hui[sample(nrow(hui)),]
#Create 10 equally size folds
folds <- cut(seq(1,nrow(hui)),breaks=10, labels=FALSE)
#Perform 10 fold cross validation
for(i in 1:k_fold){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i, arr.ind=TRUE)
  testData <- hui[testIndexes, ]
  trainData <- hui[-testIndexes, ]
  #Use the test and train data partitions however you desire...
  model <- glm(final.model.FUN, data=trainData, family = binomial())
  pred <- predict(model, testData)
  pred_class <- ifelse(pred > 0.5, 1, 0)
  cm <- confusionMatrix(testData$HUIDCOG, pred_class)
  cv.err[i] <- cm$overall[[1]]
  cv.sen[i] = cm$byClass[[1]]
  cv.spec[i] = cm$byClass[[2]]
}

# print(mean(cv.err))
result <- data.frame(mean(cv.err), mean(cv.sen), mean(cv.spec))
colnames(result) <- c('accuracy','sensitivity', 'specificity')
knitr::kable(
  result,
  caption = 'K-Fold Cross Validation Result for Logistic Regression'
)
```

Table 10: K-Fold Cross Validation Result for Logistic Regression

accuracy	sensitivity	specificity
0.6989	0.5042245	0.735798

Due to computational complexity, I have not performed K-Fold cross validation for all of the models discussed here.

Linear Discriminant Analysis:

```
lda.model <- lda(final.model.FUN, data=hui.train)
lda.pred <- predict(lda.model, hui.test, type= 'class')
pred_class <- lda.pred$class
plot(lda.model)
```

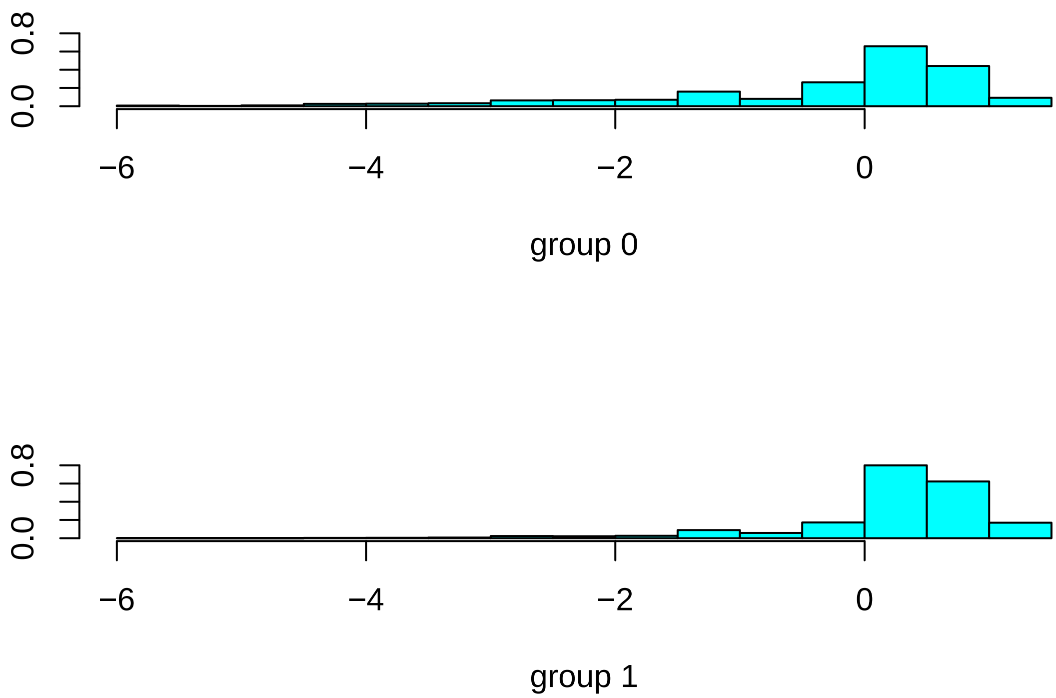


Table 11: LDA Metrics

accuracy	sensitivity	specificity
0.7088333	0.5521064	0.7215715

We get a slightly better accuracy and sensitivity but the specificity dropped compared to logistic regression.

Quadratic Discriminant Analysis

```
qda.model <- qda(final.model.FUN, data=hui.train)
qda.pred <- predict(qda.model, hui.test, type= 'class')
pred_class <- qda.pred$class
```

Table 12: QDA Metrics

accuracy	sensitivity	specificity
0.6938333	0.4805781	0.7420805

The QDA apparently don't promise any improvement over other models.

Random Forest:

```
set.seed(19)
tr <- randomForest::randomForest(final.model.FUN, data=hui.train,
                    ntree = 100, mtry= 5)
pred_class <- predict(tr, hui.test, type= 'class')
```

Table 13: Random Forest Metrics

accuracy	sensitivity	specificity
0.707	0.5461538	0.7181818

So far Random Forest have the best accuracy and sensitivity.

General Additive Model

Generalized additive models (GAMs) extend a standard linear model by allowing non-linear functions of each of the variables, while maintaining additivity. In this problem, upon further analysis, we extended non-linearity to one of the predictor meanAges by adding spline function to it since its non-linear form appears statistically significant.

```
gam.fit <- gam(HUIDCOG ~ isHappy + mobilityHelp + hearingState + s(meanAges,2),
           data=hui.train,family=binomial)
summary(gam.fit)
```

```
##
## Call: gam(formula = HUIDCOG ~ isHappy + mobilityHelp + hearingState +
##     s(meanAges, 2), family = binomial, data = hui.train)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7081 -1.2399  0.7295  0.7890  2.0115
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 17199.69 on 13999 degrees of freedom
## Residual Deviance: 16431.09 on 13994 degrees of freedom
## AIC: 16443.09
##
## Number of Local Scoring Iterations: 5
##
## Anova for Parametric Effects
##                    Df  Sum Sq Mean Sq F value    Pr(>F)
```

```
## isHappy            1   288.9 288.927  289.61 < 2.2e-16 ***
## mobilityHelp       1   142.4 142.390  142.73 < 2.2e-16 ***
## hearingState       1   105.7 105.654  105.91 < 2.2e-16 ***
## s(meanAges, 2)     1   120.7 120.741  121.03 < 2.2e-16 ***
## Residuals      13994 13960.8   0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                Npar Df Npar Chisq    P(Chi)
## (Intercept)
## isHappy
## mobilityHelp
## hearingState
## s(meanAges, 2)       1     41.111 1.438e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

End Of Part 1

Part 2: Predicting HUIDHSI (Regression Analysis)

1 Data

1.1 Data Loading

Again the first step is to load the appropriate dataset into the R Studio environment. The dataset can be found on the project github repository. Once downloaded in to the working directory of the R Studio, we can load the data using read.csv() command. Since the data file is big, I have used the `data.table` library to load the data quickly. It is generally faster than the `read.csv()` that is found in the base model. One difference between the default read.csv() and `data.table::fread()` is that we need to explicitly pass `stringAsFactor` in order to make sure that the data is read properly.

```r
library(data.table, quietly = T)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```r
hs <- fread("HStrain.csv", stringsAsFactors = T)
dim(hs)
```

```
## [1] 10000   591
```

```r
# how many different types of measurements are represented by all these columns?
cn <- colnames(hs)
table(substr(cn,start=1,stop=3))
```

```
##
## ADL ADM ALC CAG CCC CGE CIH CR1 CR2 DHH DPS DS2 EDU FAL GEN GEO HC2 HUI
##   4   4   5  45  31   8  27  34  19   9  33   4   2  15  10   2  19   1
## HUP HWT IAL IN2 LBF LON MED NUR OH3 OWN PA2 RET RPL SDC SLP SLS SMK SPA
##   1   5   6   8  19   4  33  12  27   2  49  32  19   6   1   6  21  24
## SSA TRA
##  25  19
```

We can quickly see that there are 10000 rows while the number of columns are 591. We can also see that a total of 38 diffrent measurements that are represented in the dataset and each group (except 3) further subdivided into subgroups. Therefore we spend a considerable amount of time trying to figure out how can we reduce this number to something that is managable while keeping the overall variance of the data set intact. Upon a careful look we can see that there are some variables that are used for record keeping (ID type) and kept it in the data set for administrative purposes (e.g. group of variables starts with "ADM"), we need to first remove it.

In class we've taught about the 'curse of dimensionality' which states that the more dimensions you work with, the less effective standard computational and statistical techniques become. This has repercussions that need some serious workarounds when machines are dealing with Big Data. In general, I have used the following techniques to reduce the number of predictors from what was given to us to start with

- Step 1: I have gone through the details of the data documentation that was given to us. The document shows that some group of variables are summarized as a single variable. I have chosen those variables since they represent a good balance of the group of variables without loosing a lot of variability. For example, ADLDCLS is an overall summary measurement of Instrumental and Basic Activities of Daily Living for a respondent. Also not all of the data groups that we want to include, has a summary variable. So in that case we have derived Multiple Correspondence Analysis (MCA) (since they are categorical variables) and included the dimensions that are explaining the variables significantly. Please refer appendix for details
- Step 2: Once we have finished filtering the columns manually by going through the dataset and through MCA technique, the next thing we want to do is to run a subset selection technique. We used two subset selection techniques
  - Regression Subset Selection Technique - Forward, Backward, Mixed
  - Lasso Subset Selection - Used different lambda values and then selected the best lambda for which the mean error is the lowest.
  - Finally compared both of these techniques, and since Lasso gave me the best result, and went with the columns that was ultimately filtered out.

After going through these process of predictor reduction, we finally choose 31 columns out of the possible 590 columns.

```
## [1] "ADLDCLSNO FUNC IMPAIR"       "ADLDCLSSEV IMPAIRMENT"
## [3] "ADLDCLSTOTAL IMPAIRMENT"     "CCCF1HAS NO CHRON CON"
## [5] "CR1FRHCREC FORMAL H C"       "CR2DTHCDID NOT REC H C"
## [7] "CR2DFARREG BASIS DLY"        "EDUDR04POST-SEC. GRAD."
## [9] "FALG022"                              "FALG027"
## [11] "FALG02NOT APPLICABLE"       "GENDHDIFAIR"
## [13] "GENDHDIPOOR"                    "GENDMHIFAIR"
## [15] "GENDMHIGOOD"                    "GENDMHIPOOR"
## [17] "HUPDPADPAIN ATT. LEV.2"     "HUPDPADPAIN ATT. LEV.3"
## [19] "HUPDPADPAIN ATT. LEV.4"     "HUPDPADPAIN ATT. LEV.5"
## [21] "IN2GHH$80,000 OR MORE"      "IN2GHH< $20,000"
## [23] "LONDSCR"                          "NURDHNRNOT AT HIGH N R"
## [25] "PA2DSCR"                          "SLSDCLSEXT DISSATISFIED"
## [27] "SLSDCLSEXT SATISFIED"       "SLSDCLSSATISFIED"
## [29] "SLSDCLSSL DISSATISFIED"     "SPAFPARPARTICIPANT"
## [31] "CIH.Dim.4"
```

Methods

Before we could run any model, we first need to split the dataset into train/test. For that I have used 70:30 split. Based on the filtered columns, I have build three regression models. They are: * Linear Regression * Random Forest * Gradient Boosting Model (GBM)

In terms of the error rate, the linear regression model gave me the lowest root mean square error, therefore that is what I chose as my final model in terms of predicting `HUIDHSI`. Here is the summary of the final model.

```
Call:
lm(formula = HUIDHSI ~ ., data = hsred2.train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.94931 -0.03026  0.02223  0.05567  0.45694
```

```
Coefficients:
                         Estimate Std. Error t value
(Intercept)              0.831882   0.001359 612.244
ADLDCLSMOD.IMPAIRMENT   -0.016032   0.001520 -10.545
ADLDCLSNO.FUNC.IMPAIR    0.028790   0.001769  16.271
ADLDCLSSEV.IMPAIRMENT   -0.017453   0.001468 -11.890
ADLDCLSTOTAL.IMPAIRMENT -0.007804   0.001433  -5.446
CCCF1HAS.NO.CHRON.CON    0.005390   0.001410   3.822
CR1FRHCREC.FORMAL.H.C   -0.004662   0.001756  -2.655
CR2DTHCDID.NOT.REC.H.C   0.009019   0.001974   4.568
CR2DFARREG.BASIS.DLY    -0.004080   0.001539  -2.650
EDUDR04POST.SEC..GRAD.   0.004840   0.001419   3.410
FALG022                 -0.003912   0.001501  -2.607
FALG027                 -0.006167   0.001383  -4.458
FALG02NOT.APPLICABLE     0.006580   0.001572   4.186
GENDHDIFAIR             -0.008038   0.001516  -5.302
GENDHDIPOOR             -0.009822   0.001552  -6.329
GENDMHIFAIR             -0.017192   0.001487 -11.563
GENDMHIGOOD             -0.009509   0.001431  -6.646
GENDMHIPOOR             -0.011840   0.001393  -8.498
HUPDPADPAIN.ATT..LEV.2  -0.013663   0.001375  -9.935
HUPDPADPAIN.ATT..LEV.3  -0.037400   0.001400 -26.722
HUPDPADPAIN.ATT..LEV.4  -0.072251   0.001428 -50.605
HUPDPADPAIN.ATT..LEV.5  -0.111778   0.001537 -72.711
IN2GHH.80.000.OR.MORE    0.001835   0.001451   1.264
LONDSCR                 -0.010775   0.001533  -7.029
NURDHNRNOT.AT.HIGH.N.R   0.003714   0.001434   2.590
PA2DSCR                  0.007354   0.001534   4.793
SLSDCLSEXT.DISSATISFIED -0.004990   0.001462  -3.414
SLSDCLSEXT.SATISFIED     0.014593   0.001923   7.587
SLSDCLSSATISFIED         0.011573   0.001942   5.959
SLSDCLSSL.DISSATISFIED  -0.003132   0.001512  -2.071
SPAFPARPARTICIPANT       0.004148   0.001389   2.986
CIH.Dim.4               -0.004077   0.001379  -2.958
                        Pr(>|t|)
(Intercept)              < 2e-16 ***
ADLDCLSMOD.IMPAIRMENT    < 2e-16 ***
ADLDCLSNO.FUNC.IMPAIR    < 2e-16 ***
ADLDCLSSEV.IMPAIRMENT    < 2e-16 ***
ADLDCLSTOTAL.IMPAIRMENT 5.33e-08 ***
CCCF1HAS.NO.CHRON.CON   0.000133 ***
CR1FRHCREC.FORMAL.H.C   0.007940 **
CR2DTHCDID.NOT.REC.H.C  5.00e-06 ***
CR2DFARREG.BASIS.DLY    0.008057 **
EDUDR04POST.SEC..GRAD.  0.000652 ***
FALG022                 0.009163 **
FALG027                 8.39e-06 ***
FALG02NOT.APPLICABLE    2.88e-05 ***
GENDHDIFAIR             1.18e-07 ***
GENDHDIPOOR             2.61e-10 ***
GENDMHIFAIR              < 2e-16 ***
GENDMHIGOOD             3.23e-11 ***
GENDMHIPOOR              < 2e-16 ***
```

```
HUPDPADPAIN.ATT..LEV.2   < 2e-16 ***
HUPDPADPAIN.ATT..LEV.3   < 2e-16 ***
HUPDPADPAIN.ATT..LEV.4   < 2e-16 ***
HUPDPADPAIN.ATT..LEV.5   < 2e-16 ***
IN2GHH.80.000.OR.MORE   0.206098
LONDSCR                 2.27e-12 ***
NURDHNRNOT.AT.HIGH.N.R  0.009624 **
PA2DSCR                 1.68e-06 ***
SLSDCLSEXT.DISSATISFIED 0.000644 ***
SLSDCLSEXT.SATISFIED    3.70e-14 ***
SLSDCLSSATISFIED        2.65e-09 ***
SLSDCLSSL.DISSATISFIED  0.038407 *
SPAFPARPARTICIPANT      0.002838 **
CIH.Dim.4               0.003110 **
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1136 on 6968 degrees of freedom
Multiple R-squared:  0.7241,    Adjusted R-squared:  0.7228
F-statistic: 589.8 on 31 and 6968 DF,  p-value: < 2.2e-16
```

For the final linear regression model that I chose, all of the 31 predictors are significant except the Income column (`IN2GHH.80.000.OR.MORE`). The calculated RMSE is 0.01259476 with Multiple R-Square value of 0.72 means this model explains 72% of the available data and a p-value of p-value: $< 2.2e-16$ which means its a significant model in predicting `PA2DSCR`. Lets take an example of how we can interprete this model and the relationship between the dependent and the independent variable. For example we see the coefficient for `LONDSCR = -0.010775` in the model which means that for one unit increase in the Lonliness scale, keeping all other predictors constant, it contributes to -0.010775 times decrease in HUIDHSI score which means the more lonely the person is, the more unhealthy the person will become. So we can now not only measure qualitative relations but also quantitatively measure the relationship between the dependent and independent variables. We can also compare different predictors and say which predictor is more impacting in determining the `HUIDHSI` score.

Results

In order to evaluate different models that I build, I used the measurement of root mean square error for the purpose since this is a regression problem, where we are predicting a real number rather than a categorical value. The root mean square derivation is frequently used in the world of statistics. Mean Square Error (MSE) is the variance of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. The lower the number, the better the model is. The minimum the RMSE can go is 0. Therefore a value very close to zero is considered a good model.

Here are the three MSE that I have got for different models that are trained with the train set and evaluated on the test set:

Table 14: Mean Square Error For Different Models

Linear Reg	Random Forest	Gradient Boosting Method
0.0254102	0.0141791	0.0384641

Based on the result, we can see that our best model that was able to predict the test set with the least error is *Linear Regression* followed by the model built using Random Forest. We therefore conclude that Linear Regression is the best model along with the selected set of predictors to predict `HUIDHSI` score.

Evaluation of best model on the Holdout dataset

Once I finalized the model, I have re-fit the model again on the full dataset (without any tran-test split) and then used the holdout dataset to measure the MSE which I got is **0.01533271**

Conclusion

After going through all the models, we have picked the linear regression model with four predictors as our final selected model. However we have only considered linear terms in this model, which can be a drawback in this model. We also haven't extensively used K-Fold cross validation here due to time complexity. Therefore in the future it will be noteworthy to check how adding interaction and non-linear terms and doing K-fold CV changes the predictive ability of all these models that I have tried, or if there is any other family of regression techniques stands out when these new terms / new data are introduced.

Apprendix (Part II)

Here is a list of all the libraries that are used for part II. Load all the required Libraries

```
req_libs = c("data.table", 'FactoMineR', 'factoextra', 'dplyr',
         'leaps', 'glmnet', 'randomForest', 'gbm')
lapply(req_libs, require, character.only = TRUE)
```

Load Data Set:

```
hs <- fread("HStrain.csv", stringsAsFactors = T)
```

Explore the dataset.

```
cn <- colnames(hs)
length(unique(substr(cn,start=1,stop=3)))
```

```
## [1] 38
```

```
table(substr(cn,start=1,stop=3))
```

```
##
## ADL ADM ALC CAG CCC CGE CIH CR1 CR2 DHH DPS DS2 EDU FAL GEN GEO HC2 HUI
##   4   4   5  45  31   8  27  34  19   9  33   4   2  15  10   2  19   1
## HUP HWT IAL IN2 LBF LON MED NUR OH3 OWN PA2 RET RPL SDC SLP SLS SMK SPA
##   1   5   6   8  19   4  33  12  27   2  49  32  19   6   1   6  21  24
## SSA TRA
##  25  19
```

Remove unwanted columns. * The variables that start with ADM are to do with administering the survey and are not useful for prediction. For example, ADM_RNO is a sequential record number, ADM_N09 indicates whether the interview was by phone, in-person, etc. These columns do not add any value to the model's predicting abilities, so we will remove these.

```
hsred <- dplyr::select(hs,-starts_with("ADM"))
```

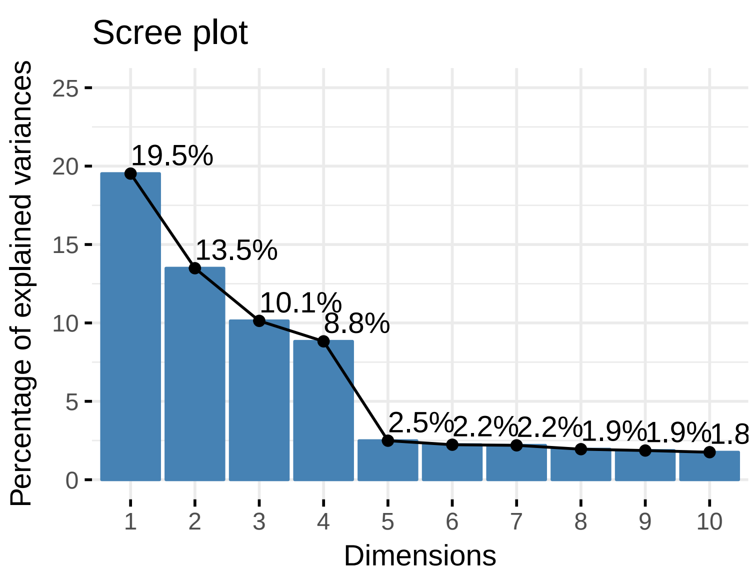
The next step for me was to go through the data documentation and search for predictors that makes sense. I need to admit here that I got a little help from the lecture notes on this. The instructor has provided a fairly good list of predictors which has an overall summary indicator, however I have included couple others that I felt was important as well.

```
hsred <- dplyr::select(hs,
ADLDCLS, # Instrumental and Basic Activities of Daily Living Classification
ALCDTTM, # Type of Drinker
CAGDFAP, #This variable indicates the frequency of assistance provided by the respondent to the main ca
CCCF1, # Has a Chronic Condition
CCCDCPD, #Has Chronic Obstructive Pulmonary Disease
CR1FRHC,# Flag for Receiving Formal Home Care Services
CR2DTHC, # Receipt of Formal or Informal Home Care
CR2DFAR, # Frequency of Assistance Received from the Main Caregiver (for the main source of assistance)
```

```
DPSDSF, # Depression Scale - Probability of Caseness to Respondents
EDUDR04, # Highest Level of Education - Household, 4 Levels
FALG02, # Number of falls - past 12 months - grouped
GENDHDI, # Perceived Health
GENDMHI, # Perceived Mental Health
HC2FCOP, # Flag for Consultation with Health Professional
HWTGBMI, # Body mass index - grouped
IN2GHH, # Total Household Income - All Sources - grouped
LONDSCR, # Three Item Loneliness Scale - Score
MEDF1, # Flag Indicating Medication Use (Past Month)
NURDHNR, # High Nutritional Risk
PA2DSCR, # PASE Score
SLSDCLS, # Satisfaction with Life Scale
SMKDSTY, # Number of Years Since Stopped Smoking Completely
SPAFPAR, # Frequency of Community-Related Activity Participation (participant)
GEOGCMA2, # Metropilitan Area Summary
HUIDHSI # response
)
```

There were some predictors which do not have any sort of summary variable, therefore for those variables I have done a quick MCA() analysis to come up with a custom columns. I have first written a small function, that lets me explore the result of the analysis and then I used the output of the function to select the number of dimensions that I want to include in the final table. Lets explore the group of variables starting with "CIH":

```
result.mca <- MCA(dplyr::select(hs,starts_with("CIH")), graph = F)
fviz_screeplot(result.mca, ylim = c(0,25), addlabels = TRUE)
```

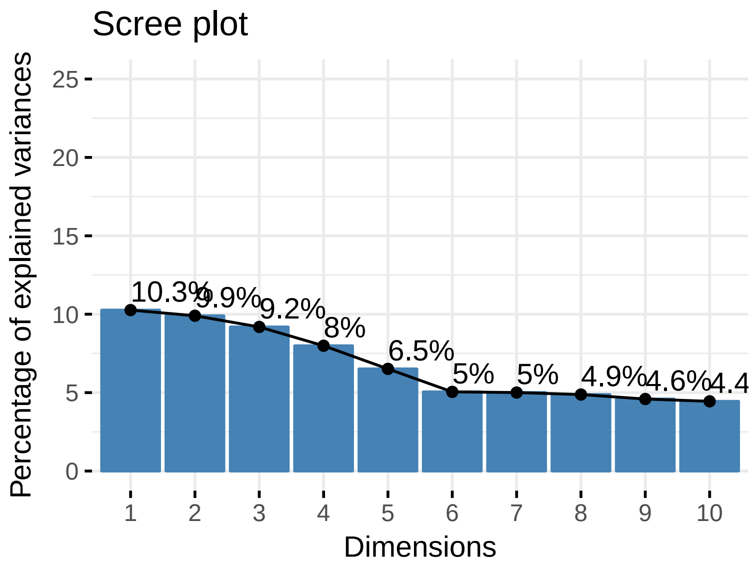



```
# looks like the first 4 Dims exaplain about 50% of the data
CIHPCs <- result.mca$ind$coord[,1:4]
colnames(CIHPCs) <- paste("CIH",colnames(CIHPCs))

# include these new predictors in the final dataset for model building
hsred <- data.frame(hsred,CIHPCs)
```

Also take a look into the variables starting with "DS2":

```
result.mca <- MCA(dplyr::select(hs,starts_with("DS2")), ncp = 10, graph = F)
fviz_screeplot(result.mca, ylim = c(0,25), addlabels = TRUE)
```




```
# looks like the first 7 Dims exaplain about 50% of the data
DS2PCs <- result.mca$ind$coord[,1:7]
colnames(DS2PCs) <- paste("DS2",colnames(DS2PCs))

# include these new predictors in the final dataset for model building
hsred <- data.frame(hsred,DS2PCs)
dim(hsred)
```

```
## [1] 10000    36
```

`hsread` now contains our final set of columns that we have chosen manually. There are 37 columns and 10000 rows. After manual selection, we can now run a subset selection and Lasso analysis to choose the most significant columns out of these 37 columns and proceed in building the model.

Regression Subset Selection

We first need to scale the dataset since some of the techniques that we are going to use requires that, split the data set into train test

```
tem <- model.matrix(HUIDHSI ~ .,data=hsred)[,-1]
X <- as.data.frame(scale(tem))
Y <- hsred$HUIDHSI
rm(tem)

# split test-train
set.seed(19)
n.train <- 7000
train <- sample(1:nrow(hs),replace=FALSE,size=n.train)
```

```
X.train <- X[train,]
Y.train <- Y[train]
X.test <- X[-train,]
Y.test <- Y[-train]
```

```
rr <- regsubsets(X.train,Y.train,nvmax=30, method="forward")
```

```
## Reordering variables and trying again:
```

```
ss <- summary(rr)
pbest <- which.min(ss$bic)
cols<- ss$which[pbest,-1] # don't include intercept
Xred <- as.matrix(X.test[,cols])
pred.test <- cbind(1,Xred) %*% coef(rr, id=pbest)
```

The `regsubsets()` choose 27 columns and came up with a MSE of **0.0349142**. Lets try and see if we can get a better result with Lasso technique that we learned in the class.

Lasso Subset Selection

```
lambdas <- 10^{seq(from=-3,to=5,length=100)} # range of lambdas to try from
alpha = 0.8
cv.lafit <- cv.glmnet(as.matrix(X.train),Y.train,alpha= alpha,lambda=lambdas) # one hot encoding
la.best.lam <- cv.lafit$lambda.1se
ll <- glmnet(as.matrix(X.train),Y.train,alpha= alpha,lambda=la.best.lam)
pred.test <- predict(ll,as.matrix(X.test))
```

Since the MSE we found here 0.0268348 is less than that of the regression subset, we chose the result produced by lasso techniques as our final set of columns. Which are:

\begin{center} **Final Set of Selected Predictors** \end{center}

```
##  [1] "ADLDCLSNO FUNC IMPAIR"   "ADLDCLSSEV IMPAIRMENT"
##  [3] "CCCF1HAS NO CHRON CON"   "CR1FRHCREC FORMAL H C"
##  [5] "CR2DTHCDID NOT REC H C"  "CR2DFARREG BASIS DLY"
##  [7] "DPSDSF"                  "FALG02NOT APPLICABLE"
##  [9] "GENDHDIFAIR"             "GENDHDIGOOD"
## [11] "GENDHDIPOOR"             "GENDMHIFAIR"
## [13] "GENDMHIPOOR"             "LONDSCR"
## [15] "NURDHNRNOT AT HIGH N R"  "PA2DSCR"
## [17] "SLSDCLSEXT DISSATISFIED" "SLSDCLSEXT SATISFIED"
## [19] "SLSDCLSSATISFIED"        "SLSDCLSSL DISSATISFIED"
## [21] "CIH.Dim.4"
```

So we can now finally create our train and test data based on these 32 filtered columns.

```
nonz <- (as.numeric(coef(ll))!=0)[-1] # rm intercept
train.data <- data.frame(HUIDHSI=Y.train,X.train[,nonz])
test.data <- data.frame(HUIDHSI=Y.test,X.test[,nonz])

# remove all the unwanted temporary variables at this stage
rm(hsred, hs)
```

Models:

Linear Regression

```
linear.fit <- lm(HUIDHSI ~ ., data = train.data)
preds <- predict(linear.fit, newdata = test.data)
with(test.data, mean((test.data$HUIDHSI-preds)^2))
```

```
## [1] 0.02541022
```

Random Forest

```
set.seed(19)
bb <- randomForest(X.train,y=Y.train,xtest=X.test,
          ytest=Y.test,ntree=200,
mtry=sqrt(ncol(X.train)),importance=TRUE)
pred.test <- bb$test$predicted
mean((Y.test - pred.test)^2)
```

```
## [1] 0.02662107
```

GBM

```
hs.train <- data.frame(HUIDHSI=Y.train,X.train)
hboost <- gbm(HUIDHSI ~ ., data=hs.train ,n.trees=400,
              interaction.depth=5, distribution="gaussian")
hs.test <- data.frame(HUIDHSI=Y.test,X.test)
pred.test <- predict(hboost,newdata=hs.test, n.trees=200,type="response")
mean((Y.test-pred.test)^2)
```

```
## [1] 0.04187508
```

From the MSE score we can see that our best model is Linear Regression.