**DS6372 – Project 2**

# **Introduction**

The Health and Aging Brain among Latino Elders (HABLE) study is focused on studying the combined impact of depression and inflammation on memory functioning among Mexican-American adults and elders (Johnson). The authors of this paper have elected to leverage data from this study to determine if it is possible to generate computer-aided diagnosis (CAD) models for cognitive impairment in the population studied within the HABLE data set. Cognitive impairment is assessed through a doctor’s diagnosis which requires the following: clinical assessment during an interview of the patient to address concerns or difficulties with life tasks. Evaluation of cognitive abilities using office-based tests, medical conditions, including mental health conditions and laboratory tests, aid in the assessment. CAD models generated from the authors analyses show promise to aid and possibly even improve diagnosis of complex cognitive assessments. Predictive models have shown excellent ability and promise to increase the speed of diagnostic process, reduce diagnostic errors, and improve quantitative evaluation. The authors have been able to obtain the HABLE dataset, as described below, for their analysis. The purpose of this paper is to provide a statistical analysis of the HABLE dataset focused on being able to predict cognitive impairment from socio economic status (SES), patient vitals, reported conditions, and laboratory tests. The analysis has two main objectives with the dataset. The first objective is to build and interpret a logistic regression model. The second objective is to build and compare additional models to improve on prediction performance metrics.

## **Data Description**

The data leveraged for this analysis is provided by one of the authors as part of analysis being performed in the authors role. The original study data set in its entirety was too large and complex for a proof of concept study, consisting of almost 750 individuals each with over 1700 separate variables. To best manage the analysis, the study data was restricted based on domain knowledge and recommendations from a Co-Chair of the HABLE study. The restricted data set we were given contained 741 individuals with 161 features or variables. These 161 features contain information relating to various patient attributes such as: age, gender, race, blood pressure, education, income, height, weight, BMI, blood-work markers and mental cognition. The dataset contains a mix of categorial and continuous features. The categorical features indicate information such as cancer status, education level, race consideration, and gender. The continuous features indicate information such as age, blood pressure, BMI, height, weight and blood-based measurements. The features were captured by both an interview process as well as through standard medical result capturing. Thus, some of the values are self-reported by participants in the study.

## **Exploratory Data Analysis (EDA)**

Possible bias stated up front: participants drawn to the HABLE study in the interest of having their memory assessed by a clinician. The first step in the EDA was to determine which features, if any, could be removed from the data set due to multicollinearity and or relevance/value for the analysis. An initial statistical summary analysis (Table 1) was used to look for data quality issues including missing values. After removing the factors above the data set contains 117 factors across ~400 observations. The next analysis was to determine correlation across factors to look for further simplification. Correlation plots were used to determine which items were highly correlated. For example, factors in the dataset such as height and weight highly correlate with the BMI factors. To reduce correlation in the data the authors reduced the data to uncorrelated factors. Using the example above BMI was kept as a factor instead of height and weight. Figures 1 and 2 provide correlation plots of the pre and post data cleansing. A few factors remained showing mild collinearity in post cleansing, these were retained due the importance in model and represent different measurements. Example LM1\_AB\_Score and LM2\_AB\_Score each measuring a different cognitive function.

In order to test the model accuracy, the authors also elected to remove doctor diagnosed factors, except for cognitive impairment, and maintained only the self-reported factors (smoking, alcohol). Since the goal of the model is to be able to predict cognitive impairment the authors felt that the model would hold more validity if the model were able to predict the cognitive impairment from SES, cognitive office tests, self-reported factors and blood test results.

In order to predict cognitive impairment using the dataset, the authors also elected to collapse the cdx\_cog feature from a 9 factor feature to a binary feature. The various factors were easily grouped into a binary configuration of cognitive impairment with value “1” or no cognitive impairment with value “0”. This imputation of variable allowed for additional flexibility in analysis with Cognitive Impairment; which encompassed follow conditions: Subjective Memory Complaint (SMC), MCI, Dementia, Other Cognition, Abnormal Cognition. Additional visualization was done, we have displayed some of the more interesting relationships in Table 2 as well as a nice break down of the response variable in relation to several predictors. After completing the EDA, the resulting data set was reduced from the initial 161 features down to 76 features.

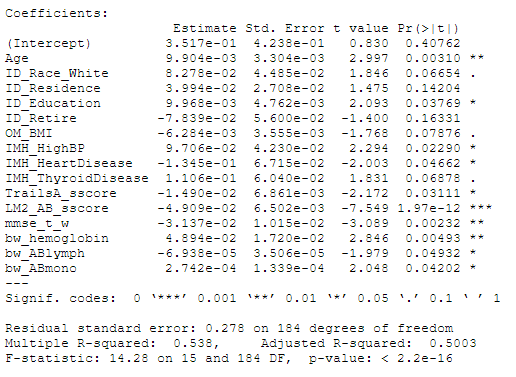
## **Analysis Question 1**

## Problem Statement

The goal of analysis question 1 is to perform a logistic regression analysis on the dataset and develop a statistically and practically significant model. The interpretation of the regression coefficients including hypothesis testing and confidence intervals are provided for the reader to provide model interpretation

## Model Selection

In order to build the initial logistic regression model, the data was split into a 50%/50% test-train split. Initial model building started with a stepwise selection approach using all predictors available in the data set. This process resulted in a model with an adjusted R-squared of 0.5003. The model built from the stepwise approach is as follows:



The stepwise model was an initial attempt to identify significant predictors. Given the extraneous predictors returned in the stepwise selection model (such as: ID\_Retire – from a practical perspective, retirement status does not necessarily line up well with our understanding of a primary source of cognitive impairment), we felt another approach would be useful as a comparative.

We selected LASSO as the second approach to determine an optimal number of variables. LASSO recommended between 5 and 15 factors for the model and the coefficients produced by the LASSO procedure are displayed below (denoted by non “.” value coefficients):

Predictor Coefficient

Intercept 4.58e-01

Age 8.09e-03

ID\_gender -1.11e-02

ID\_USlive 2.01e-04

ID\_Education 6.11e-06

OM\_BMI -2.81e-03

IMH\_HighBP 2.04e-02

TrailsA\_sscore -1.44e-02

LM1\_AB\_sscore -4.72e-03

LM2\_AB\_sscore -1.12e-02

bw\_hemoglobin 7.73e-03

bw\_ABneutro 2.03e-06

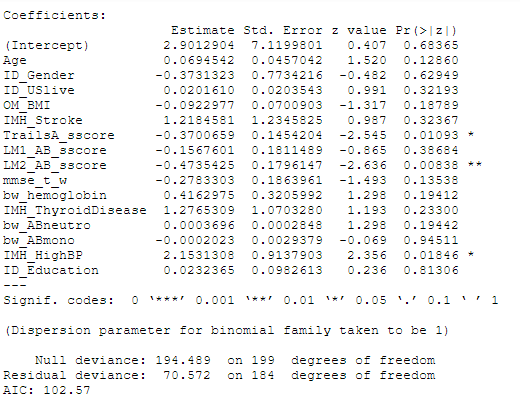
bw\_ABmonmo 9.19e-05

While the number of predictors recommended lined up with the results displayed in the stepwise model, not all predictors matched between the two approaches. For example, ID\_Race\_White was chosen by the stepwise approach but did not appear within the LASSO approach. Practically speaking, this data set was primarily constructed from respondents of Hispanic descent, so we opted to remove ID\_Race\_White completely.

Conversely, IMH\_Thyroid\_Disease was chosen by the stepwise approach and not the LASSO approach. From our experience however, we know that individuals with thyroid conditions, particularly hypothyroid conditions, do suffer from what’s commonly referred to as “brain fog” especially if they are unmedicated or undermedicated (Samuels). Therefore, we elected to carry forward this predictor into the final logistic regression model.

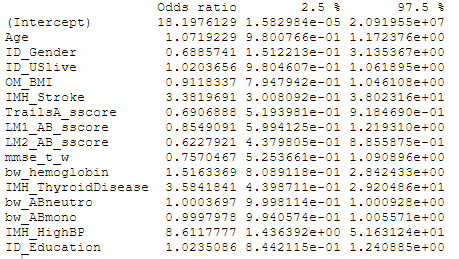
IMH\_Stroke also has a significant impact on cognitive function, especially if the stroke is severe enough. Interestingly, neither the stepwise nor the LASSO approaches selected this predictor for inclusion in their respective models. We elected to include this predictor in the logistic regression model as well based on our experience.

Combining the insights drawn from the stepwise and LASSO models, along with our practically significant analysis, we constructed a logistic regression model that identified predictors we feel would most likely would impact cognitive function. The model resulted in an AIC of 101.39 and is displayed below:



## Parameter Interpretation

Using the model above we can interpret the parameters using odds ratios and confidence intervals (CI’s) at the 95% level to allow for model interpretation. The odds ratios and CI’s are:



Examples of the odds ratio interpretation are as follows:

* The odds of an individual with high blood pressure (IMH\_HighBP) results in a 8.61% increase in diagnosis of cognitive impairment.
* The odds of an individual with thyroid disease (IMH\_ThyroidDisease) results in a 3.58% increase in diagnosis of cognitive impairment.
* The odds of an individual who has experienced a stroke (IMH\_Stroke) result in a 3.38% increase in diagnosis of cognitive impairment.
* Interestingly the number of years lived in the US results in a 1.02 increase in diagnosis of cognitive impairment. The HABLE study is 2/3 Race\_Hispanic with the reminder mainly Race\_White. This result suggests the years spent in the US could influence the onset of cognitive impairment but requires further investigation.

## Conclusion

In conclusion, the model developed for determining cognitive impairment was built leveraging multiple methods – including multiple linear regression (leveraging stepwise automated variable selection) and LASSO, with an unsophisticated train/test data split (50/50) and applying our experience to determine practically significant predictors. Using these approaches, we constructed a logistic regression model. From a statistical and practical perspective, we believe this model is effective in determining if an individual would be cognitively impaired.

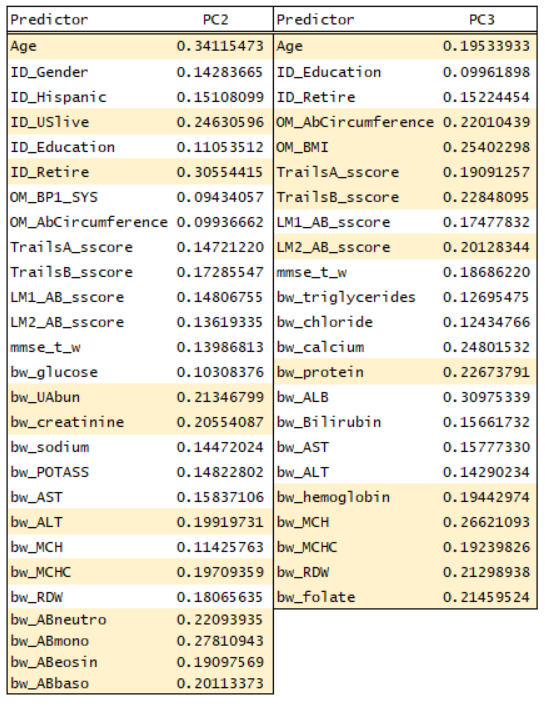
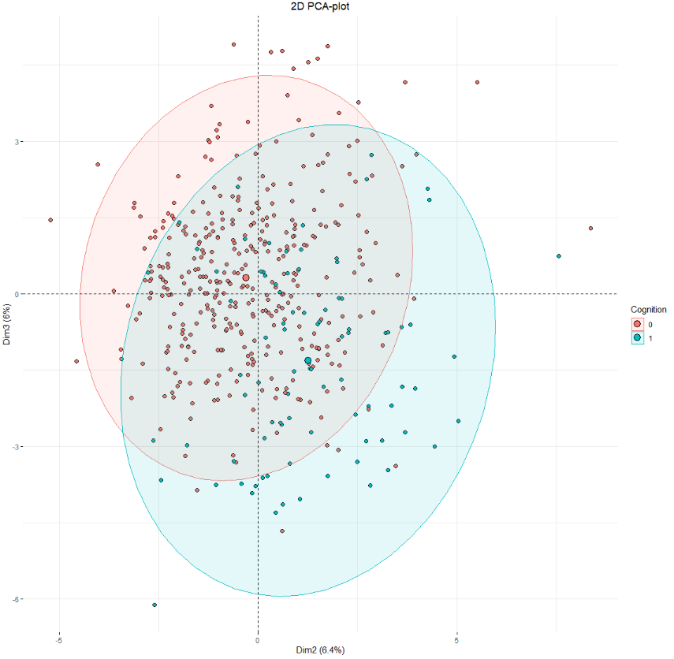
## **Analysis Question 2**

## Problem Statement

The goal of analysis question 2 is to compare predictive performance of models. The authors will leverage the model developed in analysis question 1 as a baseline for predictive performance. The baseline model will be built upon and predictive performance metrics across the various models compared to determine if predictive performance can be improved.

## Main Analysis

**PCA:** Scree Plot (Figure 4) estimates approximately that 24 PCs has an Eigenvalue >1 which account for about 80% variance (Figure 5). Running a PCA analysis is one method to reduce large numbers of variables. PCA plot of PC2 vs PC3 shows separation unfortunately it is not perfectly clean and does have some overlap. We felt the PCA analysis was functioning well enough to aid in proof of concept that a difference in cognition could be detected.



Further investigation of principal component loadings helped to identify high leverage predictors for to test in other models. An arbitrary (abs) cut-off level was used to identify the PCs with higher loadings. PC2 returned 27 predictors while PC3 only returned 23. Intuitively the loadings were further reduced. The final reduction returned 11 predictors for PC2 and 12 predictors for PC3. PCA analysis does not contain a response variable so the predictions cannot be made directly. We used the information from PCA to assess other variable selections techniques and found similarity. Interestingly each PC reported does show consistency with other statistical tests. Both PCs have identified Age, US years lived, Education, Cognitive Scores, BMI as highly explanatory. PC2’s higher loadings showed indication of metabolic dis-function (liver and kidney biomarkers) and inflammatory response (CBC levels), while PC3’s higher loadings indicate folate or B12 deficiencies and possibly Vascular Disease.

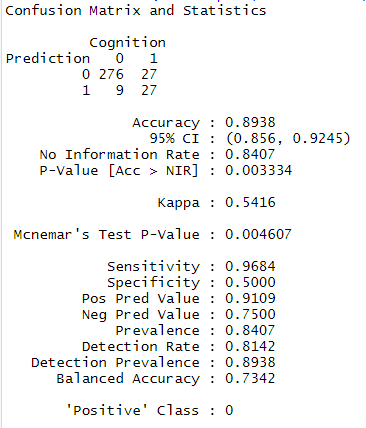
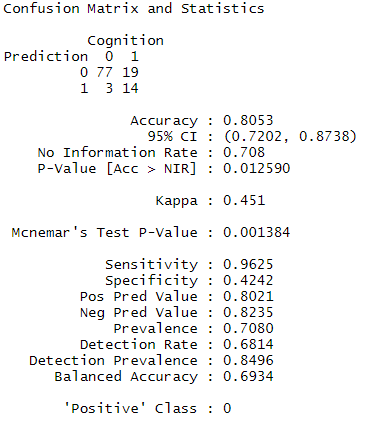
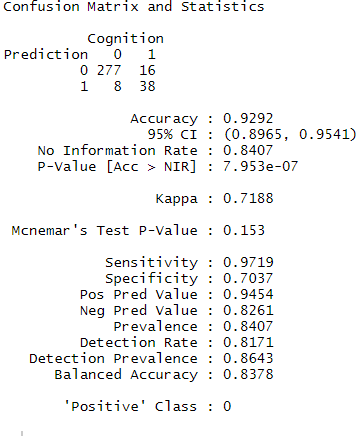
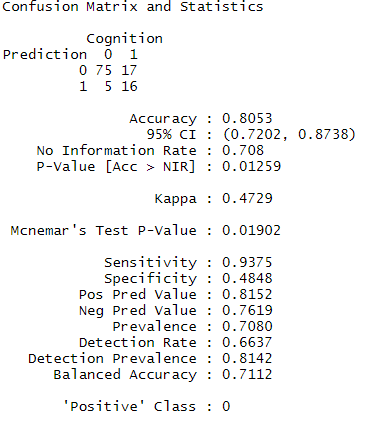
**LDA/QDA:** Modeling was conducted with the same predictors identified from Logistic Regression analysis in Analysis Question 1, minus the categorical predictors. Using the same predictors identified in Analysis Question 1 as a starting point allowed the LDA and QDA models to be compared across different methods. The results from the LDA and QDA call (Table 4) made on a randomized train/test split of (75%, 25%), respectively from the clean data set. LDA and QDA analysis showed similar results with LDA performing just slightly better than QDA.

LDA model performed had 80% Accuracy with 0.9375 Sensitivity and 0.4848 Specificity on test set.

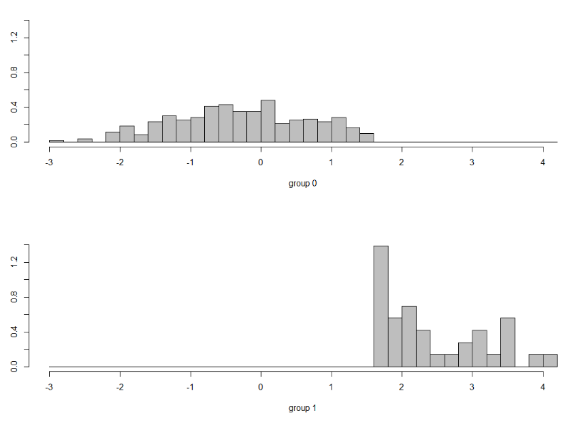
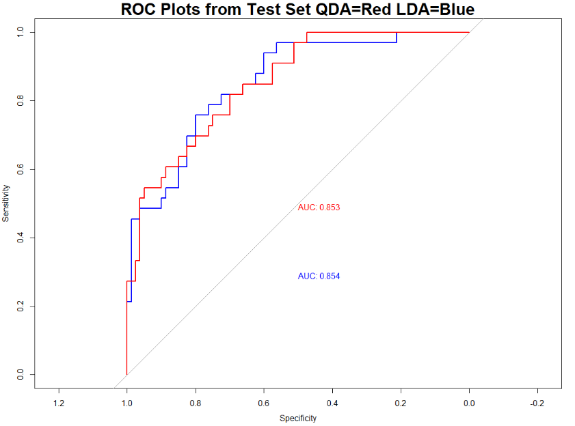
QDA model performed had 80% Accuracy with 0.9625 Sensitivity and 0.4244 Specificity on test set.

Both models suffer with low specificity, possibly mis-classifying some individuals as positive for cognitive impairment.

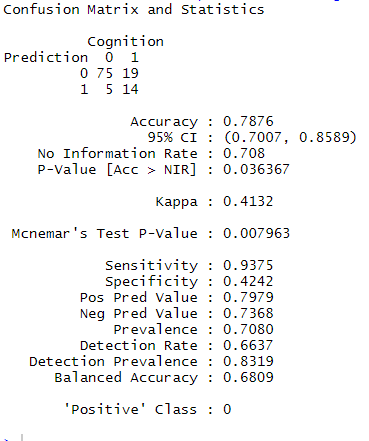
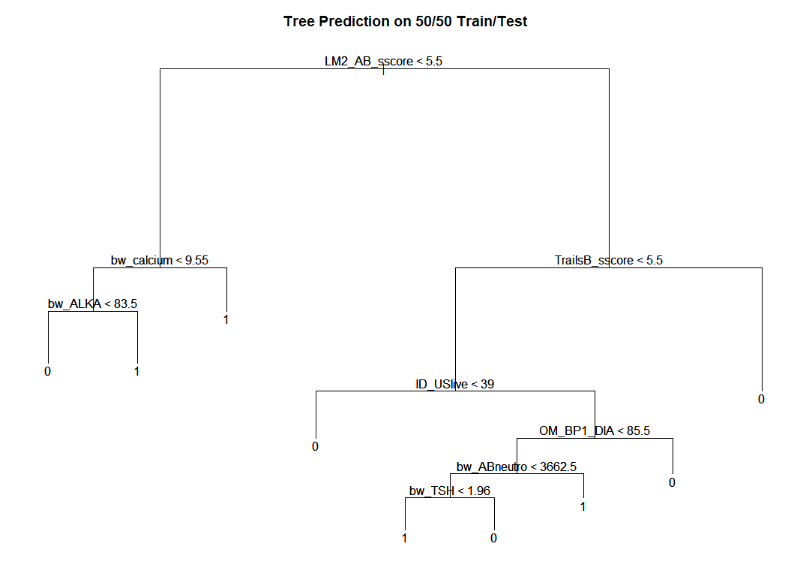
Confusion Matrix Results LDA Train/Test Set Confusion Matrix Results QDA Train/Test Set

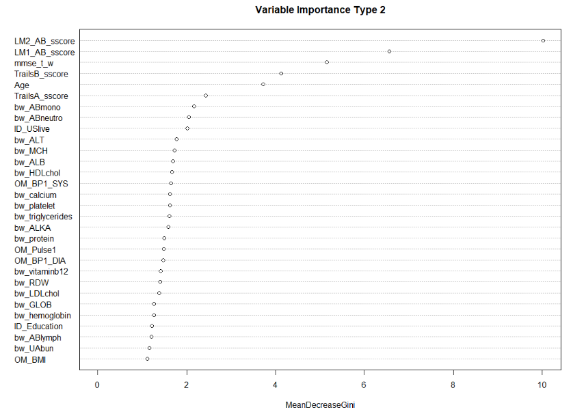
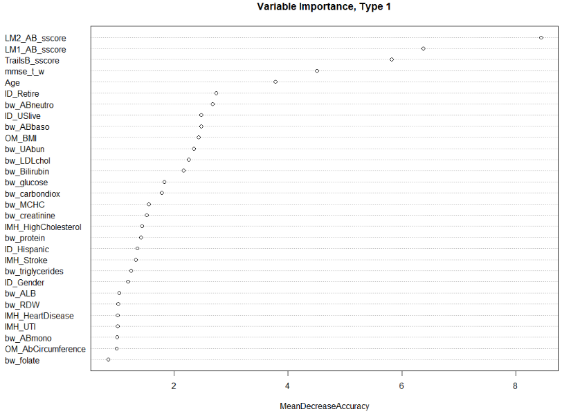
LDA Classification histogram (Left) has the X-axis defined by the co-efficient of linear discriminant for LDA model. The two groups “0” for normal and “1” for impaired, display nice separation. The ROC plots for both models (Right) show the same performance AUC=0.85. We feel that the low specificity could be due to over-fitting when building each model using the Logistic Regression Call. Further investigation into this should be carried with additional predictors.

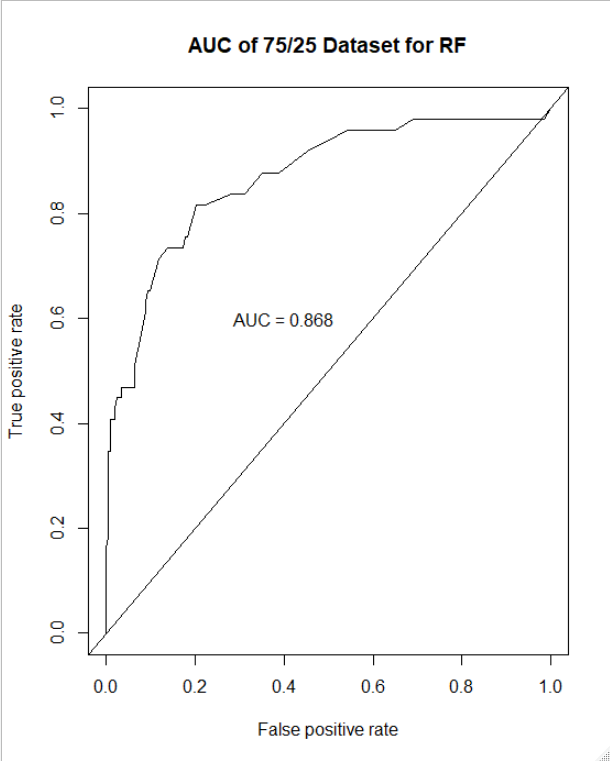
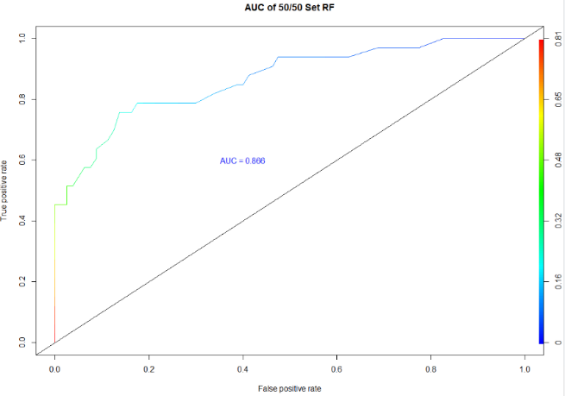
 

**Trees:** Analysis was performed on the 50/50 train/test split. The following predictors populated and produced the corresponding Confusion Matrix. The right side of the tree is very consistent to other analysis methods. The left side shows two predictors Calcium and ALKA possibly showing a linkage to liver disease. Accuracy of 78.8% and Sensitivity 0.93 and Specificity 0.4242. Specificity is low and will need to be addressed.



**Random Forest:** Analysis was also performed on both the randomized train/test split of (75%, 25%) and the 50/50 train/test split. The 50/50 train/test data set produced an AUC = 0.866. The randomized data set produced an AUC = 0.868. These results continue to reinforce the variable selections described above. Variable importance plots continue to show the importance of age, education, USLive, and cognitive screening scores (i.e., various tests that were designed to test cognitive function) as key variables in this analysis. The variable importance plots show results based on both mean decrease in accuracy and mean decrease in GINI. The higher either of these values are the more important the variable is to the model.



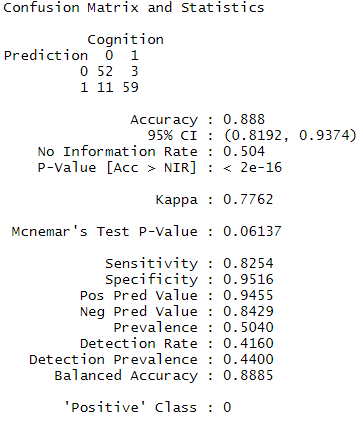
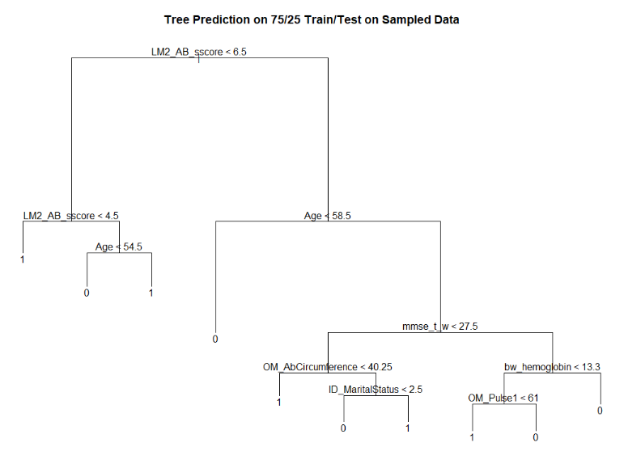


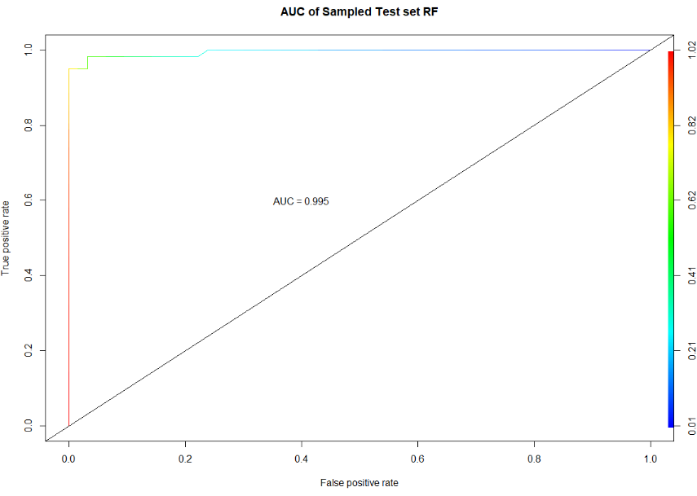
## Conclusion

The possible bias outlined in this study should not detract from the potential seen from our analysis and predictive models. Our results warrant further investigation with additional datasets or possibly run in conjunction with a physician’s clinical diagnosis to clearly demonstrate validity. A side by side comparison in a real-world setting would allow for better tuning of the predictive models and gauge the effectiveness of the claims within. An expanded sample open to additional races would also prove highly valuable in both validating and further refining these models. The HABLE data set was able to provide excellent insight into which predictors are most influential. Each statistical method (Logistic Regression, PCA, LDA, QDA, Trees and Random Forest) returned a similar set of predictors showing stability across the different methods and adding strength to the predictive claims.

It is the authors’ belief that the disproportion of cognitive impairment to normal seen within the data set may cause some of the specificity issues. The “Truth” was 365 normal and 87 impaired. After train/test split this left only small percentage of cognitively impaired individuals in either set. A data sampling approach was takento possibly address the proportion issues. Using an under and over sampling method adjusted both normal and impaired observations to a count of 250 each. We then re-analyzed the sampled data with 75/25 train test split. With all models held constant, the sampled data shows a large improvement.

Using Trees, we observed the accuracy change to 88.8%, sensitivity change to 0.8254, and specificity increase to 0.9516. There is slight shift seen in the selected predictors as well. Cognitive Scores and Age are still the first major branches, with AbCircumference, Hemoglobin, MaritalStatus, OMPulse finishing off the tree. Using Random Forest on sampled data the AUC=0.995. The sampling approach should be considered cautiously since data was manipulated; however, this approach does provide some insight on how the models may perform with more observations.





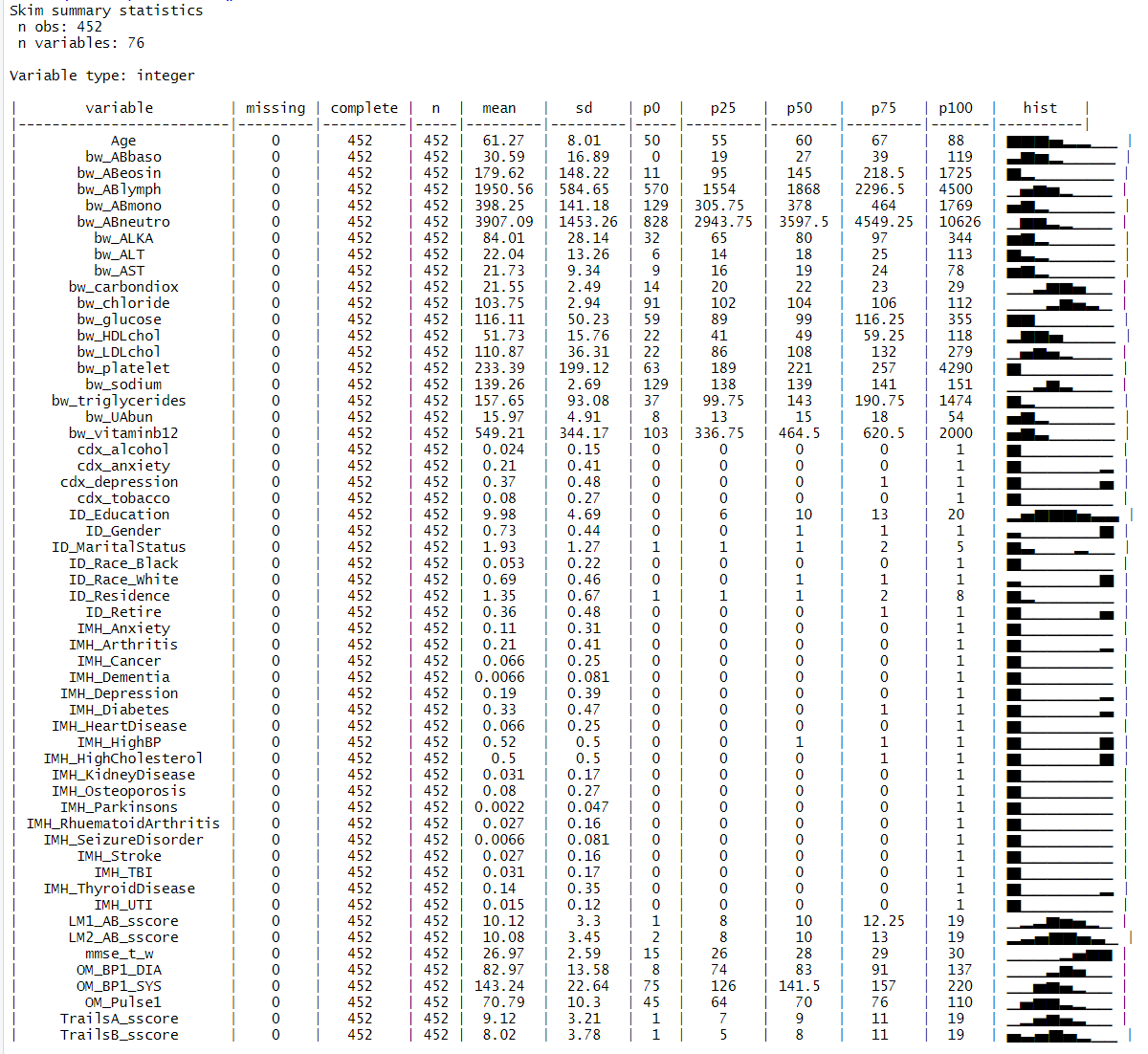
## **Appendix**

Upon review it was initially determined that the following factors or factor groups would be removed from the analysis:

* MedID – removed as it is a record identifier
* Race Identifiers – categorical factors that are summarized in another way
* CM\_notes – removed as it is a notes field
* IMH\_age variables – removed as they contained many missing values
* APOE variables – removed for initial model
* eGRF variables – removed as they contained many missing values
* cdx\_mci – remove doctor diagnosis
* Income – removed for initial model

## List of Tables

Table 1 – Summary Statistics



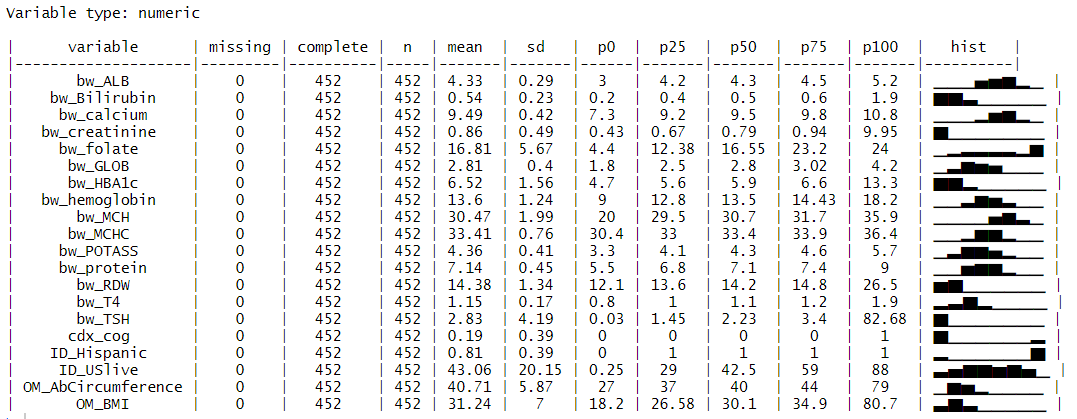


Table 2-EDA Data Visualization: Analysis plotting of predictors to gain insight, cdx\_cog was left un-bucketed to visually show the trends that supported making the predictor binary.



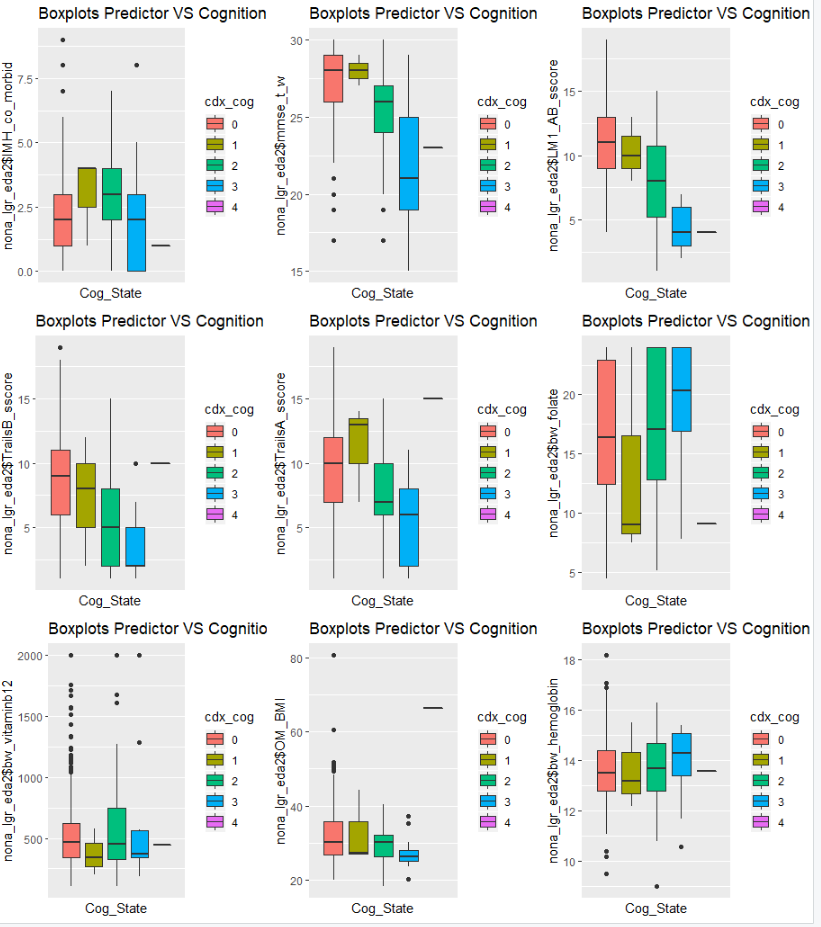


Table 3 – Initial MLR Output

Call:

lm(formula = cdx\_cog ~ ., data = train)

Residuals:

Min 1Q Median 3Q Max

-0.74889 -0.17532 -0.04557 0.13834 0.69653

Coefficients: (2 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.223e+00 2.693e+00 -0.454 0.650579

Age 8.932e-03 5.724e-03 1.560 0.121189

ID\_Gender -7.238e-02 8.016e-02 -0.903 0.368269

ID\_MaritalStatus 4.106e-03 2.368e-02 0.173 0.862606

ID\_Race\_White 1.283e-01 8.102e-02 1.583 0.115840

ID\_Race\_Black -4.493e-02 1.617e-01 -0.278 0.781552

ID\_Hispanic -3.677e-02 1.155e-01 -0.318 0.750728

ID\_Residence 4.919e-02 4.353e-02 1.130 0.260659

ID\_USlive 1.143e-03 2.406e-03 0.475 0.635569

ID\_Education 9.027e-03 8.215e-03 1.099 0.273970

ID\_Retire -9.638e-02 7.379e-02 -1.306 0.193859

OM\_Pulse1 -1.435e-03 2.853e-03 -0.503 0.615933

OM\_BP1\_SYS 3.345e-04 1.916e-03 0.175 0.861716

OM\_BP1\_DIA -6.117e-04 2.723e-03 -0.225 0.822578

OM\_AbCircumference 1.240e-03 9.275e-03 0.134 0.893855

OM\_BMI -1.129e-02 9.273e-03 -1.218 0.225589

IMH\_Cancer -9.270e-02 9.812e-02 -0.945 0.346580

IMH\_Diabetes -3.066e-03 8.501e-02 -0.036 0.971288

IMH\_HighBP 9.242e-02 6.623e-02 1.395 0.165383

IMH\_HighCholesterol 6.214e-02 6.982e-02 0.890 0.375197

IMH\_Dementia -2.540e-01 3.270e-01 -0.777 0.438743

IMH\_HeartDisease -1.562e-01 9.683e-02 -1.613 0.109169

IMH\_ThyroidDisease 1.099e-01 8.706e-02 1.263 0.209016

IMH\_KidneyDisease 3.083e-02 1.465e-01 0.210 0.833635

IMH\_Stroke 1.226e-01 1.576e-01 0.778 0.438231

IMH\_Depression 5.970e-02 9.048e-02 0.660 0.510587

IMH\_Anxiety 2.329e-02 1.139e-01 0.205 0.838286

IMH\_Osteoporosis -6.202e-02 1.130e-01 -0.549 0.584019

IMH\_Arthritis 1.225e-02 6.952e-02 0.176 0.860432

IMH\_RhuematoidArthritis 1.474e-01 2.270e-01 0.650 0.517180

IMH\_Parkinsons NA NA NA NA

IMH\_UTI 2.579e-02 1.874e-01 0.138 0.890756

IMH\_SeizureDisorder 1.277e-02 2.241e-01 0.057 0.954638

IMH\_TBI -1.156e-02 1.346e-01 -0.086 0.931654

TrailsA\_sscore -1.771e-02 1.008e-02 -1.757 0.081380 .

TrailsB\_sscore 7.287e-03 9.030e-03 0.807 0.421212

LM1\_AB\_sscore 1.331e-05 1.488e-02 0.001 0.999288

LM2\_AB\_sscore -5.677e-02 1.530e-02 -3.711 0.000309 \*\*\*

mmse\_t\_w -2.779e-02 1.431e-02 -1.943 0.054256 .

bw\_HDLchol 2.681e-03 2.322e-03 1.155 0.250365

bw\_triglycerides 5.186e-04 4.865e-04 1.066 0.288421

bw\_LDLchol -3.292e-04 7.868e-04 -0.418 0.676337

bw\_glucose -4.695e-05 1.160e-03 -0.040 0.967774

bw\_UAbun 3.684e-03 7.428e-03 0.496 0.620808

bw\_creatinine -2.616e-02 6.107e-02 -0.428 0.669156

bw\_sodium 7.508e-03 1.798e-02 0.418 0.676982

bw\_POTASS -5.306e-02 6.552e-02 -0.810 0.419545

bw\_chloride 5.837e-03 1.658e-02 0.352 0.725320

bw\_carbondiox 6.772e-03 1.337e-02 0.507 0.613300

bw\_calcium -7.193e-02 7.973e-02 -0.902 0.368679

bw\_protein 1.012e-01 8.766e-02 1.154 0.250568

bw\_ALB 3.224e-03 1.414e-01 0.023 0.981846

bw\_GLOB NA NA NA NA

bw\_Bilirubin 4.595e-02 1.503e-01 0.306 0.760370

bw\_ALKA -5.487e-04 1.117e-03 -0.491 0.624222

bw\_AST -3.586e-03 7.008e-03 -0.512 0.609744

bw\_ALT 3.507e-03 5.422e-03 0.647 0.518877

bw\_HBA1c 1.564e-02 3.371e-02 0.464 0.643372

bw\_TSH -1.022e-02 1.578e-02 -0.648 0.518275

bw\_T4 -2.197e-02 1.713e-01 -0.128 0.898164

bw\_hemoglobin 4.048e-02 3.369e-02 1.202 0.231770

bw\_MCH -1.547e-02 2.084e-02 -0.743 0.459139

bw\_MCHC 8.967e-03 5.081e-02 0.176 0.860206

bw\_RDW 1.065e-02 2.497e-02 0.427 0.670286

bw\_platelet 1.663e-04 6.380e-04 0.261 0.794825

bw\_ABneutro 1.465e-05 2.429e-05 0.603 0.547552

bw\_ABlymph -6.523e-05 5.119e-05 -1.274 0.204930

bw\_ABmono 1.545e-04 2.223e-04 0.695 0.488361

bw\_ABeosin 5.188e-05 1.724e-04 0.301 0.763937

bw\_ABbaso 2.734e-04 2.080e-03 0.131 0.895611

bw\_vitaminb12 1.200e-04 9.544e-05 1.257 0.210951

bw\_folate -5.130e-03 5.756e-03 -0.891 0.374533

cdx\_depression -7.629e-02 7.969e-02 -0.957 0.340176

cdx\_anxiety -2.645e-02 9.125e-02 -0.290 0.772403

cdx\_alcohol 8.432e-02 1.436e-01 0.587 0.558234

cdx\_tobacco -4.667e-02 1.006e-01 -0.464 0.643400

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3135 on 126 degrees of freedom

Multiple R-squared: 0.5976, Adjusted R-squared: 0.3645

F-statistic: 2.564 on 73 and 126 DF, p-value: 1.72e-06

Table 4 – LASSO Output

76 x 1 sparse Matrix of class "dgCMatrix"

1

(Intercept) 4.585587e-01

Age 8.092761e-03

ID\_Gender -1.114796e-02

ID\_MaritalStatus .

ID\_Race\_White .

ID\_Race\_Black .

ID\_Race\_Other .

ID\_Residence .

ID\_USlive 2.006045e-04

ID\_Education 6.107364e-06

ID\_Retire .

OM\_Pulse1 .

OM\_BP1\_SYS .

OM\_BP1\_DIA .

OM\_AbCircumference .

OM\_BMI -2.806414e-03

IMH\_Cancer .

IMH\_Diabetes .

IMH\_HighBP 2.040117e-02

IMH\_HighCholesterol .

IMH\_Dementia .

IMH\_HeartDisease .

IMH\_ThyroidDisease .

IMH\_KidneyDisease .

IMH\_Stroke .

IMH\_Depression .

IMH\_Anxiety .

IMH\_Osteoporosis .

IMH\_Arthritis .

IMH\_RhuematoidArthritis .

IMH\_Parkinsons .

IMH\_UTI .

IMH\_SeizureDisorder .

IMH\_TBI .

TrailsA\_sscore -1.439168e-02

TrailsB\_sscore .

LM1\_AB\_sscore -4.718397e-03

LM2\_AB\_sscore -3.644274e-02

mmse\_t\_w -1.115850e-02

bw\_HDLchol .

bw\_triglycerides .

bw\_LDLchol .

bw\_glucose .

bw\_UAbun .

bw\_creatinine .

bw\_sodium .

bw\_POTASS .

bw\_chloride .

bw\_carbondiox .

bw\_calcium .

bw\_protein .

bw\_ALB .

bw\_GLOB .

bw\_Bilirubin .

bw\_ALKA .

bw\_AST .

bw\_ALT .

bw\_HBA1c .

bw\_TSH .

bw\_T4 .

bw\_hemoglobin 7.729703e-03

bw\_MCH .

bw\_MCHC .

bw\_RDW .

bw\_platelet .

bw\_ABneutro 2.033515e-06

bw\_ABlymph .

bw\_ABmono 9.195361e-05

bw\_ABeosin .

bw\_ABbaso .

bw\_vitaminb12 .

bw\_folate .

cdx\_depression .

cdx\_anxiety .

cdx\_alcohol .

cdx\_tobacco .

Table 5-LDA/QDA Call

lda(cdx\_cog\_bucket ~ 6.507920e-02\*Age + -8.264541e-03\*OM\_BMI + -3.912401e-02\*TrailsA\_sscore + 4.938666e-02\*LM1\_AB\_sscore + -2.707941e-01\*LM2\_AB\_sscore + -1.469278e-01\*mmse\_t\_w + 1.329650e-01\*bw\_hemoglobin + 1.181958e-05\*bw\_ABneutro + 1.589155e-03\*bw\_ABmono + -6.490268e-05\*bw\_ABlymph, data = train)

## List of Figures

Figure 1 – Pre-Cleaning Correlation Plots

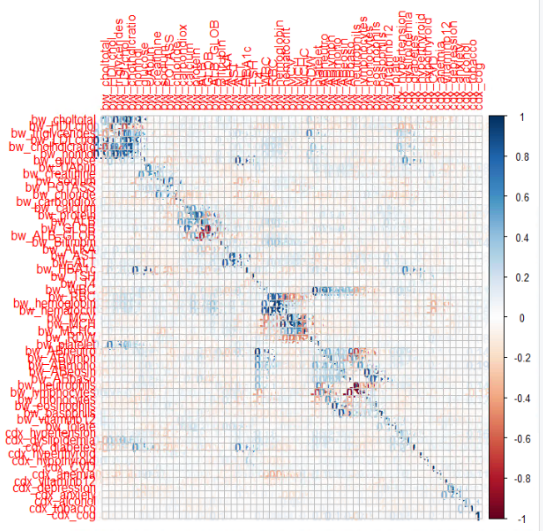
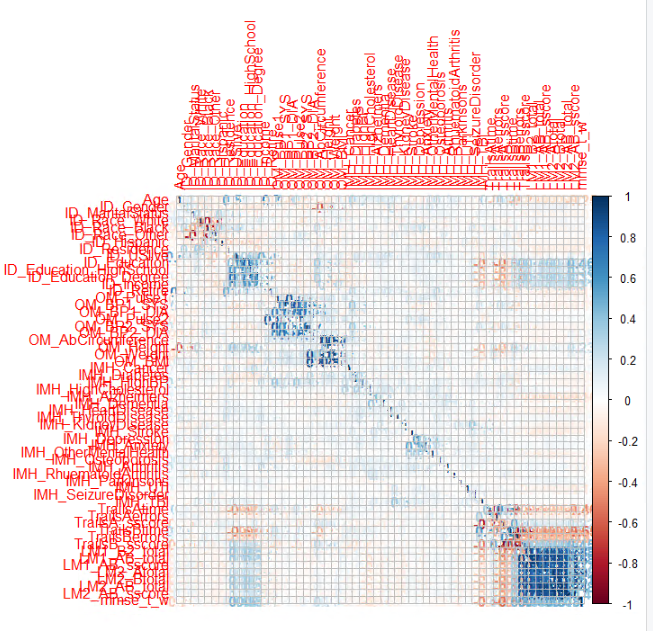


Figure 2 – Post Cleaning Correlation Plots

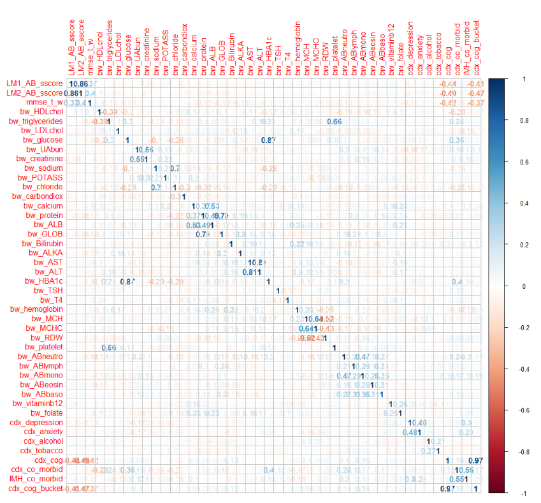
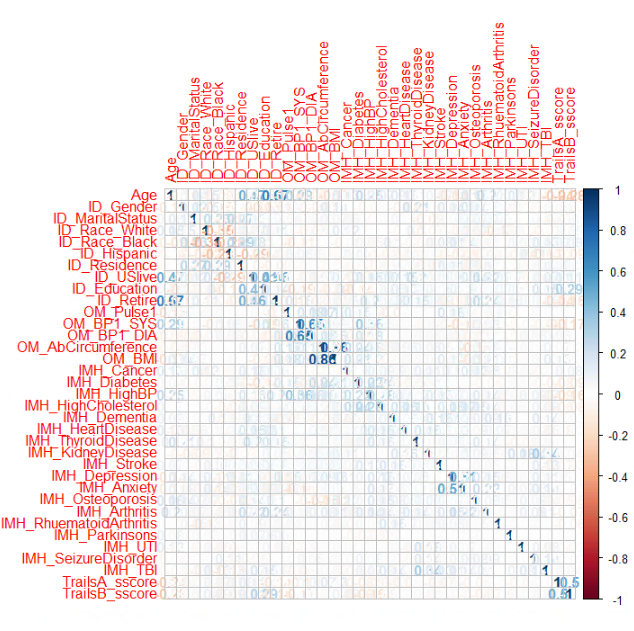


Figure 3 – LASSO Results

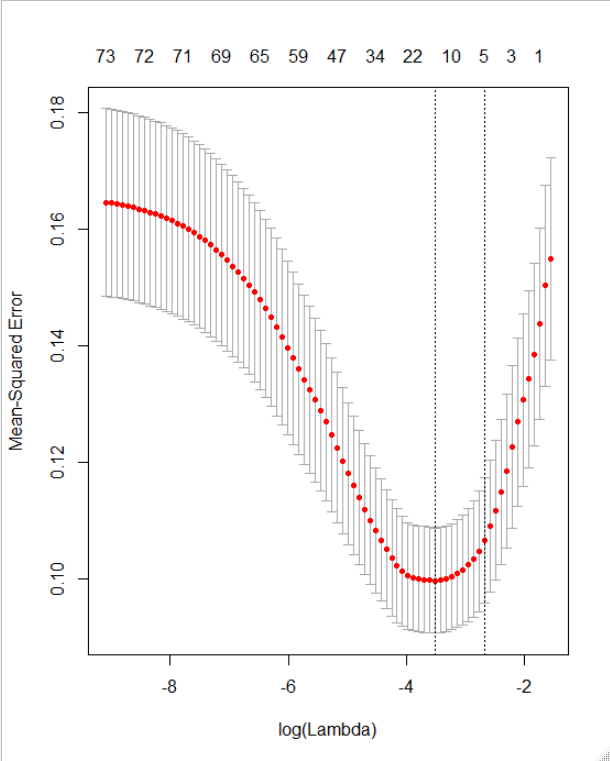
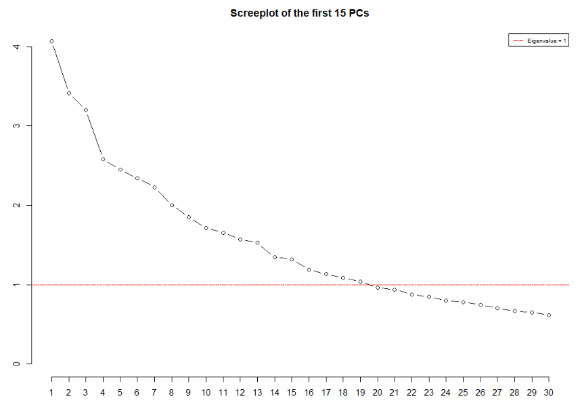
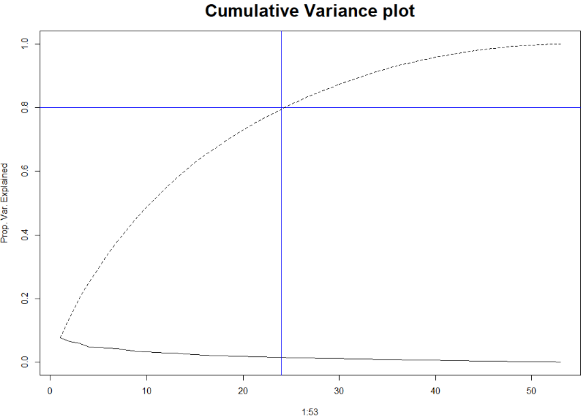


Figure 4-PCA Scree Plot Figure 5-PCA Cumulative Variance Plot

## References

Johnson, Leigh A. et al. "Depression, Inflammation, And Memory Loss Among Mexican Americans: Analysis Of The HABLE Cohort". *International Psychogeriatrics*, vol 29, no. 10, 2017, pp. 1693-1699. *Cambridge University Press (CUP)*, doi:10.1017/s1041610217001016.

Samuels, Mary H. “Psychiatric and Cognitive Manifestations of Hypothyroidism.” *Current Opinion in*

*Endocrinology, Diabetes, and Obesity*, U.S. National Library of Medicine, Oct. 2014

www.ncbi.nlm.nih.gov/pmc/articles/PMC4264616/.

## R-Code for Analysis

Code is also available at: https://github.com/urbandj/STATS\_6372\_Project2.git

# loadings

library(skimr)

library(dplyr)

library(RColorBrewer)

library(glmnet)

library(gplots)

library(ggplot2)

library(leaps)

library(MASS)

library(ggfortify)

library(grid)

library(gridExtra)

library(tidyverse) # data manipulation

library(cluster) # clustering algorithms

library(factoextra) # clustering algorithms & visualization

library(rgl) ##For mac users you may need to download Xquartz before the 3d plots will run.

library(tree)

library(ISLR)

library(randomForest)

# Load data

setwd("F:/SMU/DS6372/Project 2/STATS\_6372\_Project2/EDA")

setwd("C:/Users/daj0079/Desktop/SMU\_2nd/Stats\_Project\_2")

medData1 <- read.csv(file="data\_cut.csv", header=TRUE, sep=',', na.strings=c("", "NA"))

## added co\_morbid variable which is sum of all morbidites cdx class variable

medData1$cdx\_co\_morbid<-rowSums(medData1[,144:151])

minus\_cols<-c(1,7:15,16,17,22:24,29:34,36:38,41,43,45,47,48,49,51,53,55,57,59,61,63,64,65,66,68,70,72,74,76,78,80:82,84,85,87,88,90:92,95,99,100,104,105,114,122,123,125,126,136:140,143:151,157:161)

medData1[,-minus\_cols]->medData

## added co\_morbid variable which is sum of all morbidites IMH class

medData$IMH\_co\_morbid<-rowSums(medData[,16:32] )

## recode of Hispanic level into NO=0 and Yes=1

medData<- medData %>% mutate(ID\_Hispanic=dplyr::recode(ID\_Hispanic,

`1`=0,

`2`=1,

`3`=1,

`4`=1,

`5`=1))

# recode cognitive variable

######

#levels of lgr\_eda2$cdx\_cog

######

#0=normal\*\*\*\*\*

#1=Subjective Memory Complaint(SMC)\*\*\*\*

#2=MCI\*\*\*\*

#3=Dementia\*\*\*\*

#4=Other Cognition

#5=Abnormal Cognition\*\*\*\*

#7=Dont Know

#8=Refuse to answer

#9=Not Appicable

medData$cdx\_cog\_bucket<-ifelse(medData$cdx\_cog=='0', 0, ifelse(medData$cdx\_cog=='1', 1,

ifelse(medData$cdx\_cog=='2', 1,

ifelse(medData$cdx\_cog=='3', 1,

ifelse(medData$cdx\_cog=='4', 1,

ifelse(medData$cdx\_cog=='5', 0,0))))))

# replace all NA values with 0; should we get rid of NAs instead?

medData <- na.omit(medData)

#################################################################

# EDA and DATA VIS see appendix for results

#################################################################

##SKIM view missing, type and spread of data.

skim(medData)

##GGPAIRS plots ~20 per plot

scol1<- c(1:20)

medData[,scol1]->eda1

mcols1<-c(8)

eda1[,-mcols1]->eda1

scol2<- c(21:40)

medData[,scol2]->eda2

scol3<- c(41:79)

medData[,scol3]->eda3

ggpairs(eda1)

ggpairs(eda2)

ggpairs(eda3)

#CORRELATION PLOTS WITH COLINEARITY

##omited NA, we are LEFT WITH ~400 obs and 79 variables for corr plot.

##clearly there are some highly correlated variables. We need to simplfy

##since some the correlated are basically represented are more than on variable

corr\_data<-c(1,7:15,17,32:34,38,41,43,45,47,49,51,53,55,57,59,61,63,65,66,68,70,72,74,76,78,80,104,105,143,157:161)

medData1[,-corr\_data]->corr\_eda

na.omit(corr\_eda)->corr\_eda

precorr1<-c(1:58)

precorr2<-c(59:118)

corr\_eda[,precorr1]->cp11

corr\_eda[,precorr2]->cp12

corr\_object1 = cor(cp11)

corr\_object2= cor(cp12)

##plot was too big needed to break into smaller chuncks

corrplot(corr\_object1, method = "number")#use this in write-up

corrplot(corr\_object2, method = "number")#use this in write-up

#####################################################

##cleaned Corr plots after removing colinearity

medData->corr\_noNA

corr1<-c(1:35)

corr2<-c(36:79)

corr\_noNA[,corr1]->cp1

corr\_noNA[,corr2]->cp2

corr\_object1 = cor(cp1)

corr\_object2= cor(cp2)

##plot was too big needed to break into smaller chuncks

corrplot(corr\_object1, method = "number")#use this in write-up

corrplot(corr\_object2, method = "number")#use this in write-up

##################################################################

#Question 1 LGR Model

##################################################################

#Remove extra columns needed for other analysis.

lgr\_cols<-c(76:78)

medData[,-lgr\_cols]->reduced

# test / train data set split 50/50

train <- reduced[1:200,]

test <- reduced[201:452,]

full.fit <- lm(formula = cdx\_cog\_bucket ~ ., data = train)

# full fit (MLR) before converting response to factor - R2 0.3932

train$cdx\_cog\_bucket<-as.numeric(train$cdx\_cog\_bucket)

full.model <- lm(cdx\_cog\_bucket~., data = train)

summary(full.model)

# stepwise selection - LM

step(lm(cdx\_cog\_bucket~., data = train),direction="both")

# stepwise recommended model - R2 = 0.5003

model.stepwise <- lm(formula = cdx\_cog\_bucket ~ Age + ID\_Race\_White + ID\_Residence + ID\_Education +

ID\_Retire + OM\_BMI + IMH\_HighBP + IMH\_HeartDisease + IMH\_ThyroidDisease +

TrailsA\_sscore + LM2\_AB\_sscore + mmse\_t\_w + bw\_hemoglobin +

bw\_ABlymph + bw\_ABmono, data = train)

summary(model.stepwise)

# LASSO call

x=model.matrix(cdx\_cog\_bucket~.,train)[,-1]

y=(train$cdx\_cog\_bucket)

xtest<-model.matrix(cdx\_cog\_bucket~.,test)[,-1]

ytest<-test$cdx\_cog\_bucket

# Plot LASSO model

grid=10^seq(10,-2, length =608)

lasso.mod=glmnet(x,y,alpha=1, lambda =grid)

par(mfrow=c(1,1))

cv.out=cv.glmnet(x,y,alpha=1) #alpha=1 performs LASSO

plot(cv.out)

bestlambda<-cv.out$lambda.min #Optimal penalty parameter. You can make this call visually.

lasso.pred=predict (lasso.mod ,s=bestlambda ,newx=xtest)

# identify they variables selected by LASSO

coef(lasso.mod,s=bestlambda)

# convert dependent variable to factor

train[, 'cdx\_cog\_bucket'] <- as.factor(train[, 'cdx\_cog\_bucket'])

test[, 'cdx\_cog\_bucket'] <- as.factor(test[, 'cdx\_cog\_bucket'])

# manually built model using practical knowledge and LASSO and Stepwise - AIC 102.57

model.manual <- glm(cdx\_cog\_bucket~ Age + ID\_Gender + ID\_USlive + OM\_BMI + IMH\_Stroke + TrailsA\_sscore + LM1\_AB\_sscore + LM2\_AB\_sscore

+ mmse\_t\_w + bw\_hemoglobin + IMH\_ThyroidDisease + bw\_ABneutro + bw\_ABmono + IMH\_HighBP + ID\_Education

,family=binomial(link='logit'),data=train)

summary(model.manual)

# Using the summary coefficients we can generate CI for each one in the table and get odds ratios - manual model

exp(cbind("Odds ratio" = coef(model.manual), confint.default(model.manual, level = 0.95)))

#################################################################

###DATA SAMPLING to normalize th counts of disease and normal states, Over/ Under /Both

################################################################

sum(is.na(medData))

set.seed(1975)

data\_over\_sam <- ovun.sample(cdx\_cog\_bucket ~., data = medData, method = "over",N = 730)$data

data\_under\_sam <- ovun.sample(cdx\_cog\_bucket ~., data = medData, method = "under",N =176)$data

data\_both\_sam <- ovun.sample(cdx\_cog\_bucket ~., data = medData, method = "both",N =500)$data

### simple tables that returns count and prop of cdx\_cog and cdx\_cog\_bucket to keep in mind during anaylsis

cog\_table<-table( medData$cdx\_cog,medData$cdx\_cog)

cog\_table

prop.table(cog\_table) # row percentages

bucket\_table<-table( medData$cdx\_cog\_bucket, medData$cdx\_cog\_bucket)

bucket\_table

prop.table(bucket\_table) # row percentage

reduced\_table<--table( reduced$cdx\_cog\_bucket, reduced$cdx\_cog\_bucket)

bucket\_table

prop.table(bucket\_table) # row percentage

table(data\_over\_sam$cdx\_cog\_bucket)

table(data\_under\_sam$cdx\_cog\_bucket)

table(data\_both\_sam$cdx\_cog\_bucket)

#####################################################################

##PCA

#####################################################################

#removing cat preds and synthetic vars

k\_cols<-c(16:33,72:76,77:79)

medData[,-k\_cols]->pca\_set

#pca with all predictors only continous variables

pc\_cont\_result<-prcomp(pca\_set,scale.=TRUE)

pc\_cont\_scores<-pc\_cont\_result$x

pc\_cont\_result$x

pc\_cont\_scores<-data.frame(pc\_cont\_scores)

options("max.print" = 10000)

pca<-pc\_cont\_result$x

as.data.frame(pca)->pca

pc\_cont\_result$rotation

#Scree plot

screeplot(pc\_cont\_result, type= "l", npcs = 30, main = "Screeplot of the first 15 PCs", ylim=c(0,4))

abline(h = 1, col="red", lty=6)

legend("topright", legend=c("Eigenvalue = 1"),

col=c("red"), lty=5, cex=0.6)

#Cumulative Variance plo

eigenvals<-(pc\_cont\_result$sdev)^2

plot(1:53,eigenvals/sum(eigenvals),type="l",main="Cumulative Variance plot",ylab="Prop. Var. Explained",ylim=c(0,1),cex.main=2)

cumulative.prop<-cumsum(eigenvals/sum(eigenvals))

lines(1:53,cumulative.prop,lty=2,,abline(v=24,h=.80 ,col="blue"))

#2d scatter plot of groups

library("factoextra")

fviz\_pca\_ind(pc\_cont\_result, geom.ind = "point", pointshape = 21,

pointsize = 2,

fill.ind = factor(medData$cdx\_cog\_bucket),

col.ind = "black",

palette = "wes\_palette",

addEllipses = TRUE,

label = "var",

col.var = "black",

repel = TRUE,axes = c(2,3),

legend.title = "Cognition") +

ggtitle("2D PCA-plot") +

theme(plot.title = element\_text(hjust = 0.5))

##Estimate the predictors for the highest loading values and compare this is the write to other selection methods

pc\_cont\_result$rotation->pca\_results

(as.data.frame(pca\_results))->pca\_results

pca\_results <- cbind(Row.Names = rownames(pca\_results), pca\_results)

pca\_results

#takes abs of loading for quick comparaison

abs\_pca\_results<-abs(pca\_results[,2:53])

##looks at loadings with high levelof contribution

pc2\_vars <- subset(abs\_pca\_results, PC2 >0.09)

pc2\_vars[,c(2,3)]

pc3\_vars <- subset(abs\_pca\_results, PC3 >0.09)

pc3\_vars[,c(2,3)]

#################################################################

#LDA QDA ANALYSIS

####################################################################

##remove the catogorical variable for LDA QDA

qda\_split<-c(2:7,16:33,72:76)

medData[,-qda\_split]->qda\_vars

set.seed(1234)

trainobs=sample(seq(1,dim(qda\_vars)[1]),round(.75\*dim(qda\_vars)[1]),replace=FALSE)

LDA\_train=qda\_vars[trainobs,]

LDA\_test=qda\_vars[-trainobs,]

##LDA prediction with 75/25 train test and ROC curve

lda<-lda(cdx\_cog\_bucket~ Age + OM\_BMI + ID\_USlive+ TrailsA\_sscore + LM1\_AB\_sscore + LM2\_AB\_sscore

+ mmse\_t\_w + bw\_hemoglobin + bw\_ABneutro + bw\_ABmono +ID\_Education

, data = LDA\_train)

lda

#Run lda prediction on train set

lda.pred<-predict(lda, LDA\_train)

matrix\_pred<-lda.pred$class

bucket\_train<-as.factor(LDA\_train$cdx\_cog\_bucket)

table(Predicted=lda.pred$class,Cognition=LDA\_train$cdx\_cog\_bucket)

#Print results

confusionMatrix(matrix\_pred,bucket\_train, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

# plot out the histogram for eac predicted groups

ldahist(lda.pred$x[,1], g= lda.pred$class,col = 8)

#Run lda on test set

test.lda <- predict(lda, newdata=LDA\_test)

matrix\_test<-test.lda$class

table(Predicted=test.lda$class, Cognition=LDA\_test$cdx\_cog\_bucket)

bucket\_test<-as.factor(LDA\_test$cdx\_cog\_bucket)

#plots the results from the test set

plot(test.lda$x[,1], test.lda$class, col=LDA\_test$cdx\_cog\_bucket+10)

confusionMatrix(matrix\_test, bucket\_test, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#plot ROC of LDA test results

pred <- prediction(test.lda$posterior[,2], LDA\_test$cdx\_cog\_bucket)

perf <- performance(pred,"tpr","fpr")

plot(perf,colorize=TRUE)

auc.train <- performance(pred, measure = "auc")

auc.train <- [auc.train@y.values](mailto:auc.train@y.values)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

##QDA predictions with 75/25 train test and ROC curve

qda<-qda(cdx\_cog\_bucket~ Age + OM\_BMI + ID\_USlive+ TrailsA\_sscore + LM1\_AB\_sscore + LM2\_AB\_sscore

+ mmse\_t\_w + bw\_hemoglobin + bw\_ABneutro + bw\_ABmono +ID\_Education

, data = LDA\_train)

qda

#Use qda to on train set

qda.pred<-predict(qda, LDA\_train)

qda\_class<-qda.pred$class

#print out training results

table(Predicted=qda.pred$class,Cognition=LDA\_train$cdx\_cog\_bucket)

confusionMatrix(qda\_class ,bucket\_train, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#run qda on test set

test.qda <- predict(qda, newdata=LDA\_test)

#print out QDA test results

qda\_matrix\_test<-test.qda$class

table(Predicted=test.qda$class, Cognition=LDA\_test$cdx\_cog\_bucket)

plot(test.qda$posterior[,2], test.qda$class, col=bucket\_test)

confusionMatrix(qda\_matrix\_test,bucket\_test, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#plot ROC of QDA test results

par(mfrow=c(1,1))

pred2 <- prediction(test.qda$posterior[,2], LDA\_test$cdx\_cog\_bucket)

perf2 <- performance(pred2,"tpr","fpr")

plot(perf2,colorize=TRUE)

auc.train2 <- performance(pred2, measure = "auc")

auc.train2 <- [auc.train2@y.values](mailto:auc.train2@y.values)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train2[[1]],3), sep = ""))

#combined Roc plots on test data for LDA and QDA no real difference is seen

library(pROC)

plot(roc(LDA\_test$cdx\_cog\_bucket,test.lda$posterior[,2]), print.auc = TRUE, col = "blue", main='ROC Plots from Test Set QDA=Red LDA=Blue',

print.auc.y = 0.3, cex.main=2)

plot(roc(LDA\_test$cdx\_cog\_bucket, test.qda$posterior[,2]), print.auc = TRUE, col = "red", add=TRUE)

#######################################################################################################################

## TREE CALCULATIONS

#######################################################################################################################

#Run with 50/50 train test split

#######################################################################################################################

summary(train$cdx\_cog\_bucket)

#Need to convert to factor for trees to work correctly

train$cdx\_cog\_bucket = factor(train$cdx\_cog\_bucket)

test$cdx\_cog\_bucket = factor(test$cdx\_cog\_bucket)

summary(train$cdx\_cog\_bucket)

#Run regular trees

tree.medical=tree(cdx\_cog\_bucket~.,train)

plot(tree.medical)

text(tree.medical, pretty = 0)

tree.pred=predict(tree.medical,test,type="class")

table(tree.pred,test$cdx\_cog\_bucket)

#Prune the trees

set.seed(3)

par(mfrow=c(1,1))

cv.medical=cv.tree(tree.medical,FUN=prune.misclass)

names(cv.medical)

plot(cv.medical)

#Fit the pruned tree and visualize

prune.medical=prune.misclass(tree.medical,best=9)

plot(prune.medical)

text(prune.medical,pretty=0)

title("Tree Prediction on 50/50 Train/Test")

tree.pred=predict(prune.medical,test,type="class")

table(tree.pred,test$cdx\_cog\_bucket)

confusionMatrix(tree.pred,test$cdx\_cog\_bucket, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#We can verify the overfitting idea by saying to split the tree

#very deep

prune.medical2=prune.misclass(tree.medical,best=7)

plot(prune.medical2)

text(prune.medical2,pretty=0)

tree.pred=predict(prune.medical2,test,type="class")

table(tree.pred,test$cdx\_cog\_bucket)

confusionMatrix(tree.pred,test$cdx\_cog\_bucket, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#For ROC curves on a single decision tree you need predicted probabilities to

#use the previous R scripts. Lets just use the example here with the last run.

tree.pred=predict(prune.medical,test,type="vector")

head(tree.pred)

pred <- prediction(tree.pred[,2], test$cdx\_cog\_bucket)

roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")

#Note in the following code the term "train" means nothing here.

#I'm just rinsing and repeating code the produces the curve.

auc.train <- performance(pred, measure = "auc")

auc.train <- [auc.train@y.values](mailto:auc.train@y.values)

plot(roc.perf,main="AUC of Test set of a Single Tree")

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

#Execute the random forest

rf.model = randomForest(cdx\_cog\_bucket~.,train,subsets=train,importance=T,ntree=100)

fit.pred<-predict(rf.model,newdata=test,type="response")

table(fit.pred,test$cdx\_cog\_bucket)

confusionMatrix(fit.pred,test$cdx\_cog\_bucket, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#Run ROC curves for RandomForest Model

rf.pred<-predict(rf.model,newdata=test,type="prob")

pred <- prediction(rf.pred[,2], test$cdx\_cog\_bucket)

roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")

#Note in the following code the term "train" means nothing here.

#I'm just rinsing and repeating code the produces the curve.

auc.train <- performance(pred, measure = "auc")

auc.train <- [auc.train@y.values](mailto:auc.train@y.values)

plot(roc.perf,main="AUC of 50/50 Set RF",colorize=TRUE)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""),col="blue")

#Variable Importance Plots

varImpPlot (rf.model,type=1,main="Variable Importance")

varImpPlot (rf.model,type=2,main="Variable Importance")

#Run with 75/25 train test split

#######################################################################################################################

set.seed(1234)

trainobs=sample(seq(1,dim(reduced)[1]),round(.75\*dim(reduced)[1]),replace=FALSE)

train\_r=reduced[trainobs,]

test\_r=reduced[-trainobs,]

summary(train\_r$cdx\_cog\_bucket)

#Need to convert to factor for trees to work correctly

train\_r$cdx\_cog\_bucket = factor(train\_r$cdx\_cog\_bucket)

test\_r$cdx\_cog\_bucket = factor(test\_r$cdx\_cog\_bucket)

summary(train\_r$cdx\_cog\_bucket)

#Run regular trees

tree.medical=tree(cdx\_cog\_bucket~.,train\_r)

plot(tree.medical)

text(tree.medical, pretty = 0)

tree.pred=predict(tree.medical,test\_r,type="class")

table(tree.pred,test\_r$cdx\_cog\_bucket)

#Prune the trees

set.seed(3)

par(mfrow=c(1,1))

cv.medical=cv.tree(tree.medical,FUN=prune.misclass)

names(cv.medical)

plot(cv.medical)

#Fit the pruned tree and visualize

prune.medical=prune.misclass(tree.medical,best=9)

plot(prune.medical)

text(prune.medical,pretty=0)

tree.pred=predict(prune.medical,test\_r,type="class")

table(tree.pred,test\_r$cdx\_cog\_bucket)

#We can verify the overfitting idea by saying to split the tree

#very deep

prune.medical2=prune.misclass(tree.medical,best=7)

plot(prune.medical2)

text(prune.medical2,pretty=0)

tree.pred=predict(prune.medical2,test\_r,type="class")

table(tree.pred,test\_r$cdx\_cog\_bucket)

confusionMatrix(tree.pred,test\_r$cdx\_cog\_bucket, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#For ROC curves on a single decision tree you need predicted probabilities to

#use the previous R scripts. Lets just use the example here with the last run.

tree.pred=predict(prune.medical,test\_r,type="vector")

head(tree.pred)

pred <- prediction(tree.pred[,2], test\_r$cdx\_cog\_bucket)

roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")

#Note in the following code the term "train" means nothing here.

#I'm just rinsing and repeating code the produces the curve.

auc.train <- performance(pred, measure = "auc")

auc.train <- [auc.train@y.values](mailto:auc.train@y.values)

plot(roc.perf,main="AUC of Test set of a Single Tree")

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

#Execute the random forest

rf.model = randomForest(cdx\_cog\_bucket~.,train\_r,subsets=train,importance=T,ntree=100)

fit.pred<-predict(rf.model,newdata=test\_r,type="response")

table(fit.pred,test\_t$cdx\_cog\_bucket)

#Run ROC curves for RandomForest Model

rf.pred<-predict(rf.model,newdata=test\_r,type="prob")

pred <- prediction(rf.pred[,2], test\_r$cdx\_cog\_bucket)

roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")

#Note in the following code the term "train" means nothing here.

#I'm just rinsing and repeating code the produces the curve.

auc.train <- performance(pred, measure = "auc")

auc.train <- [auc.train@y.values](mailto:auc.train@y.values)

plot(roc.perf,main="AUC of Test set RF")

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

#Variable Importance Plots

varImpPlot (rf.model,type=1,main="Variable Importance, Type 1")

varImpPlot (rf.model,type=2,main="Variable Importance Type 2")

####################################################

#Tress on both way sampled data with 75/25 split

tree\_split<-c(76)

data\_both\_sam[,-tree\_split]->data\_both\_sam

set.seed(1234)

trainobs=sample(seq(1,dim(data\_both\_sam)[1]),round(.75\*dim(data\_both\_sam)[1]),replace=FALSE)

Both\_train=data\_both\_sam[trainobs,]

Both\_test=data\_both\_sam[-trainobs,]

#Need to convert to factor for trees to work correctly

Both\_train$cdx\_cog\_bucket = factor(Both\_train$cdx\_cog\_bucket)

Both\_test$cdx\_cog\_bucket = factor(Both\_test$cdx\_cog\_bucket)

summary(Both\_train$cdx\_cog\_bucket)

#Run regular trees

tree.medical=tree(cdx\_cog\_bucket~.,Both\_train)

plot(tree.medical)

text(tree.medical, pretty = 0)

tree.pred<-predict(tree.medical,Both\_test,type="class")

table(tree.pred,Both\_test$cdx\_cog\_bucket)

#Prune the trees

set.seed(3)

par(mfrow=c(1,1))

cv.medical=cv.tree(tree.medical,FUN=prune.misclass)

names(cv.medical)

plot(cv.medical)

#Fit the pruned tree and visualize

prune.medical=prune.misclass(tree.medical,best=9)

plot(prune.medical)

text(prune.medical,pretty=0)

title("Tree Prediction on 75/25 Train/Test on Sampled Data")

tree.pred=predict(prune.medical,Both\_test,type="class")

table(tree.pred,Both\_test$cdx\_cog\_bucket)

confusionMatrix(tree.pred,Both\_test$cdx\_cog\_bucket, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#We can verify the overfitting idea by saying to split the tree

#very deep

prune.medical2=prune.misclass(tree.medical,best=7)

plot(prune.medical2)

text(prune.medical2,pretty=0)

tree.pred=predict(prune.medical2,Both\_test,type="class")

table(tree.pred,Both\_test$cdx\_cog\_bucket)

confusionMatrix(tree.pred,Both\_test$cdx\_cog\_bucket, dnn = c("Prediction", "Cognition"), mode = "sens\_spec")

#For ROC curves on a single decision tree you need predicted probabilities to

#use the previous R scripts. Lets just use the example here with the last run.

tree.pred=predict(prune.medical,Both\_test,type="vector")

head(tree.pred)

pred <- prediction(tree.pred[,2], Both\_test$cdx\_cog\_bucket)

roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")

#Note in the following code the term "train" means nothing here.

#I'm just rinsing and repeating code the produces the curve.

auc.train <- performance(pred, measure = "auc")

auc.train <- [auc.train@y.values](mailto:auc.train@y.values)

plot(roc.perf,main="AUC of Sampled Test set of a Single Tree")

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

#Execute the random forest

srf.model = randomForest(cdx\_cog\_bucket~.,Both\_train,subsets=Both\_train,importance=T,ntree=100)

sfit.pred<-predict(srf.model,newdata=Both\_test,type="response")

table(sfit.pred,Both\_test$cdx\_cog\_bucket)

#Run ROC curves for RandomForest Model

srf.pred<-predict(srf.model,newdata=Both\_test,type="prob")

spred <- prediction(srf.pred[,2], Both\_test$cdx\_cog\_bucket)

roc.perf = performance(spred, measure = "tpr", x.measure = "fpr")

#Note in the following code the term "train" means nothing here.

#I'm just rinsing and repeating code the produces the curve.

auc.train <- performance(spred, measure = "auc")

auc.train <- [auc.train@y.values](mailto:auc.train@y.values)

plot(roc.perf,main="AUC of Sampled Test set RF", colorize=TRUE)

abline(a=0, b= 1)

text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

#Variable Importance Plots

varImpPlot (srf.model,type=1,main="Variable Importance, Type 1")

varImpPlot (srf.model,type=2,main="Variable Importance Type 2")

###############################################

##K means clustering

####################################

#removed explanitory vaiables

k\_cols<-c(72:75,76,79)

medData[,-k\_cols]->km\_set

#Normalized the scale for kmeans anaylsis

km<-scale(km\_set)

#distance plot of scale df

distance <- get\_dist(km)

dev.off()

fviz\_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))

#Kmeans model choose centers and nstart

k2 <- kmeans(km, centers =2, nstart =25)

#plot kmeans model

fviz\_cluster(k2, data = km)

k2

#indentify the optimal # of cluster

set.seed(123)

fviz\_nbclust(km, kmeans, method = "wss")

#tuned kmeans model

set.seed(123)

final <- kmeans(km, 2, nstart = 50)

print(final)