Derivation of Risk Factor Prediction Models for Major Depressive Disorders.

Owen R. Evans
Final Project Outline
DSE6111G- Predictive Modeling

Executive Summary

Major depressive disorder constitutes one of the most common mental disorders in the United States, affecting millions of individuals. In fact, the World Health Organization (WHO) has deemed depression the leading cause of disability around the world. Depression is a mood disorder that results in lack of energy, poor sleep, anger/irritability, substance abuse and self-harm. Those suffering from depression are also often incapable of achieving high levels of productivity. It is thus imperative that depression sufferers get the medical and psychiatric help necessary to combat these mental health disorders. Unfortunately, those suffering from depression most often suffer in silence. They are typically unwilling and incapable of seeking help or are just unaware that they are suffering from this disorder. Additionally, there is no obvious quantitative medical test, such as a blood test, that can diagnose this disorder. Quite often, the diagnosis is solely dependent on self-reporting, which typically occurs when disorder is the most severe.

There thus exists a need to provide an alternative means to identify individuals that are most at risk for depression. In particular, it is of particular interest to utilize statistical learning techniques to determine the relationship between the demographic, psychographic and medical of individuals and the diagnosis of depressive disorder. For instance, if an individual (or acquaintances of said individual) recognized that they possessed a risk factors that were correlated with a depression diagnosis, they may be more motivated to seek help. The first section of this statistical modeling project focused mainly on exploring common classification algorithms (logistic regression, discriminant analysis, decision trees) as a means to predict depression diagnosis. Data was aggregated from Behavioral Risk Factor Surveillance System (BRFSS) to provide a clean data set free from outliers and missing values and comprised of ~128,000 observations/respondents and 21 dependent variables. Initial attempts to derive sound models capable of accurate prediction for the positive class (i.e., depression diagnosis) were plagued by low sensitivity, attributed mainly to the use a highly imbalanced data set. The utilization of bagged decision trees and a rebalancing of the data set via an under sampling of the majority class, resulted in significant model improvement. Models based upon random forest algorithms had a sensitivity of 65%, a value that is relatively good for attempting to model complex human behavior. These models indicated that risk factors such as lack of sleep, overworking, gender and marital status were associated with a depression diagnosis.

The second part of this statistical modeling project approached the prediction of depressive disorders from a more quantitative angle. Specifically, approaches to model the severity of depression via regression approaches were explored. In this case, the numerical dependent variable of interest was a depression severity score calculated from the PHQ-09 module on 2017 National Health and Nutrition Examination Survey (NHANES). The NHANES also contains a wealth of demographic, laboratory (i.e., blood tests), medical, dietary and psychographic data. The major goal of this exercise was to develop a model or models that would quantitatively predict the severity of depression based upon certain risk factors. Multivariate linear regression models using predictors as determined via best subset selection resulted in a model with a fairly modest mean squared error, but still substantially better than the null model. Key variables that seem to affect the severity of a depressive disorder were identified and the models were further optimized via regularization. It was observed that all linear regression models were largely affected by right

handed skewness of the dependent variable and preponderance of zero values. Elimination of observations with zero values for depression scores seemingly improved model accuracy and resulted in a random forest regression model with acceptable prediction accuracy (R2 of 0.65).

Lastly, the final project initially endeavored to perform principal components regression and partial least squares on the same NHANES derived dataset. Unfortunately, the PCR or PLS models obtained were no better than the null model. As such, the project focused on exploring the utility of PCR and PLS to predict body mass index based upon dietary intake. For this exercise, the modules in the 2014 version of the NHANES were used to derive a clean and aggerated dataset of dietary intake for over 8000 individuals/observations and 65 dependent variables. This effort concluded that PLS was the most appropriate statistical learning approach, affording a model with a modest prediction accuracy.

Data and Approach

The data used for this project come from two distinct and publicly available resources. The derivation of classification models used to identify individuals who have received a depression diagnosis were compiled from The Behavioral Risk Factor Surveillance System (BRFSS). The BRFSS is a comprehensive nationwide telephone survey run by CDC and designed to collect key information such general health, physical activity levels, tobacco, drug or alcohol use, and health care access. The proceeding classification models were mainly designed to provide reliable predictions for a positive depression diagnosis, captured in the BRFSS survey under the variable name, *ADDEPEV3*. Specifically, a total of 445,985 respondents provided an answer to the following inquiry: *Have you had a depressive disorder?* The response to this inquiry served as the binary response variable on which to base a series of classification models. A variety of independent (or predictor) variables were chosen based mainly upon domain knowledge and focused on the following individual characteristics: *general health, dietary intake, weight, gender, age, marital status, income, alcohol/drug abuse and level of education.* Further details of the independent variables chosen are provide in the appendix. Individuals providing a response of "refused" or "don't know" were removed from the dataset. In most cases, blank or missing values were replaced with median or mode values for continuous and categorical variables, respectively. The majority of classification models in this project were based upon a clean and aggregated dataset with 272,946 individuals and a total 21 separate continuous and categorical predictors. For some classification models, it was necessary to balance the dataset as the response rate for the minority class were fairly small (~17%). This re-balancing was accomplished by randomly under-sampling the majority class to achieve parity for the class response rate. The details of the dataset pre-processing and links to the data and relevant codebooks are provided in the appendix.

Regression models aimed at providing reliable predictions for the severity of depression were based upon an aggregated dataset derived principally from the 2017 National Health and Nutrition Examination Survey (NHANES). Similar to the BRFSS, the NHANES is a nationwide survey designed to assess the health and nutritional status for adults and children living in the United States. It is a rich compilation of demographic, laboratory (i.e., blood tests), medical, dietary and psychographic data. This project focused mainly on deriving specific risk factors that could reliably and accurately predict the severity of depression in individuals. The specific metric used to determine depression severity was engineered via the summation of individual responses to PHQ-09, a patient health questionnaire that provides a continuous score that has been shown to be correlated to the severity of a depressive disorder [i]. Independent variables were chosen based upon domain knowledge and were focused mainly on demographics, psychographics, general health status and medical laboratory data. The majority of the regression models studied in this project were based upon a clean and aggregated data set with 3800 individuals and 24 independent variables.

Classification Models - Detailed Findings

Introduction. One of the primary goals of this effort was to explore the utility and capability of common classification models to identify the specific risk factors associated with diagnosis of depression. To accomplish this goal, models such logistic regression, linear and quadratic discriminant analysis, decision trees (normal, boosted, bagged and random forest) were all explored in detail. Each of these models ultimately differs in complexity, flexibility and interpretability and it is thus expected that an optimal model type for the dataset in question will be derived and validated. All of the classifiers explored in this effort utilized the binary response variable (*ADDPEV3*), which is a Yes/No (1,2) response to a question concerning the diagnosis of depression. Predictor variables, listed in detail in Table 1, were a mix of categorical and continuous metrics.

Table 1. Variables derived from the BRFSS and chosen for classification modeling

Variable Name	Description	Type	Variable Name	Description	Type
ADDPEV3	Depression Diagnosis	Categorical	INCOME2	Annual Family Income	Categorical
SEXVAR	Gender	Categorical	WEIGHT2	Body Weight	Continuous
GENHLTH	Health Status	Categorical	EXERANY2	Times per Week, Exercise	Continuous
MENTHLTH	Days per Month with Poor Mental Health	Continuous	ACEDEPRS	Lived w/Someone with Depression	Categorical
HLTHPLN1	Health Care Coverage	Categorical	ACEDRINK	Lived w/Someone with Alcoholism	Categorical
BPHIGH4	Diagnosed with High Blood Pressure	Categorical	MARIJAN1	Days per Month Used Marijuana	Continuous
TOLDHI2	Diagnosed with High Cholesterol	Categorical	BMI5CAT	Body Mass Index	Categorical
MARITAL	Marital Status	Categorical	DRNKWK1	# Alcoholic Drinks/Week	Continuous
EDUCA	Level of Education	Categorical	PA2VIGM	Minutes of Vigorous Activity per Week	Continuous
EMPLOY1	Employment Status	Categorical	FRUTDA2	# Fruits Eaten per Week	Continuous
VEGESU1	Number of Vegetables per Week	Continuous	GRENDA1	# Dark Green Vegetables per Day	Continuous

Logistic Regression. Logistic regression is a common choice of statistic models for the classification of binary variables due mainly to its simplicity, efficiency and interpretability. For this reason, we initially attempted to derive a logistic model capable of predicting a depression diagnosis based upon the risk factors/predictors shown in Table 1. For exploratory purposes, a logistic model was first derived on the entirety of the data set with the main goal of identifying those independent variables that were statistically significant. This effort identified three independent variables (BPHIGH4, INCOME2, FRUTDA2) with p-values in excess of 0.05. These variables were removed from further modeling in order to improve prediction accuracy.

A logistic model was then built utilizing a subset of the original data and split into training/testing groups (50/50). The model was initially derived on the training data and validated with the test data in order to probe for overfitting, which can happen with high dimensional data sets and logistic regression. With many features, the algorithm will attempt to derive a highly complex decision boundary that may not generalize well on the test data or future data. In this specific case, the misclassification rate (proportion of misclassified observations) for the final logistic regression model was 16.5%. At first glance, this appears to suggest a model with modest accuracy. However, this value is only modestly lower than the 19.6% observed for the null model (Class=No).

Further inspection of the confusion matrix for this logistic model also indicates a relatively poor sensitivity of 31.4%. Sensitivity is a proportion of the positive class (i.e. those with a positive

depression diagnosis) that have been correctly predicted. As such, the utility of this model (or any model) should not only be judged on accuracy and specificity, but on its ability to accurately predict the class of interest.

Table 2. Confusion Matrix - Logistic Regression - Test Set

		Reference	
		Yes	No
Prediction	Yes	8387	4264
	No	18357	105467

The combination of high specificity and low sensitivity is the hallmark of a dataset that is significantly imbalanced. The class of interest in this case only comprises 19.6% of the original dataset. As such, the algorithm will be largely biased to learning the patterns and intricacies of the majority class and will thus have limited predictive power for the minority class. While this may be case for this logistic model, the misclassification error rate is still better than the null model. There does exist a series of options (weighting, under sampling, oversampling) that can mitigate the deleterious effects of imbalanced data, but it was deemed informative to see if any of the other classifiers can handle the imbalance any better.

One of the key benefits of the logistic regression algorithm is its interpretability. Similar to a standard multivariate linear regression, a logistic regression will afford a series of coefficients that provide further information. Because logistic regression aims to derive coefficients to fit a logit link function, the coefficients derived provide information on the change in the log of the odds ratio for the positive class. For instance, we can provide some interpretation for the preceding logistic model for the coefficients with highest absolute magnitude (Table 3). Specifically, that those respondents with a college education (EDUCA6) will have a corresponding decrease in log-odds ratio of having a depression by -1.328. Likewise, those respondents who have lived with someone with depression (ACEDEPRS) will tend to have a corresponding increase in log-odds ratio of having depression by 0.976.

Table 3. Logistic Model Coefficients - Top Two Absolute Values

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.664	3.59E-01	10.203	< 2e-16 ***
EDUCA6	-1.328	3.55E-01	-3.744	0.000181 ***
EDUCA5	-1.280	3.55E-01	-3.609	0.000307 ***
EDUCA4	-0.997	3.55E-01	-2.81	0.004952 **
EDUCA3	-0.984	3.56E-01	-2.763	0.005724 **
ACEDEPRSNo	0.976	3.37E-02	28.966	< 2e-16 ***

Linear and Quadratic Discriminant Analysis. Discriminant analysis can provide for a more robust alternative to logistic regression by allowing for multi-class prediction, classification of classes that are well separated and the classification of small data sets. Discriminant analysis is similar to logistic regression but relies on modeling the distribution of predictors separately for each class (density functions) and the use of Bayes' theorem and the prior probability to calculate the probability of class membership (Pr Y=k |X=x) – the posterior probability. Maximizing the posterior probability of class membership ultimately affords a discriminant function, a means of classification. The added utility of discriminant analysis comes at a cost, mainly that it assumes that the distribution of each dependent variable is normal. Secondly, for linear discriminant analysis (LDA) it is assumed that the covariance matrix of each class is equivalent. Differing covariance matrices will require the use of quadratic discriminant analysis (QDA)

For the purposes of this project, it was immediately recognized that linear or quadratic discriminant analysis may not be appropriate for the dataset in question. The dataset utilized for the preceding logistic model contains a mixture of both categorical and continuous variables. LDA may allow for the use of categorical variables, but their presence clearly violates the normality assumption and may lead to a decrease in model accuracy. Secondly, the dataset has a high number of observations and we are attempting to perform binary classification – a situation that may be best suited for logistic regression.

To aid in a model comparison, both LDA and QDA were carried out with the same set of data. Models were trained on a split dataset and the model performance was determined on a validation test set. Quite surprisingly, the LDA performed equally as well as the logistic fit with a slight improvement in sensitivity. However, both models are still struggling to correctly predict the minority class (depression diagnosis).

Table 4. Model Performance Comparison

	Logistic Regression	LDA	QDA
Accuracy	0.836	0.836	0.775
Sensitivity	0.310	0.350	0.478
Specificity	0.963	0.954	0.847

A stacked histogram of the resulting linear discriminant function values for each observation in the dataset as a function of class membership can afford a sense on how well this model can discriminate between the two classes (Figure 1). Additionally, the values obtained from the discriminant function for specific predictors may be utilized to identify those variables with significant influence. For instance, variables such as *General Health Status, Mental Health Status, Marital Status* and *Exercise Frequency* tended to have the most disparate group means amongst the two classes of interest. In aggregate, the LDA fit is capable of discriminating a fairly minor proportion of the minority class. This inability is unlikely to be derived from the violation of the multivariate normality requirement but is instead more likely derived from the class imbalance problem as described earlier.

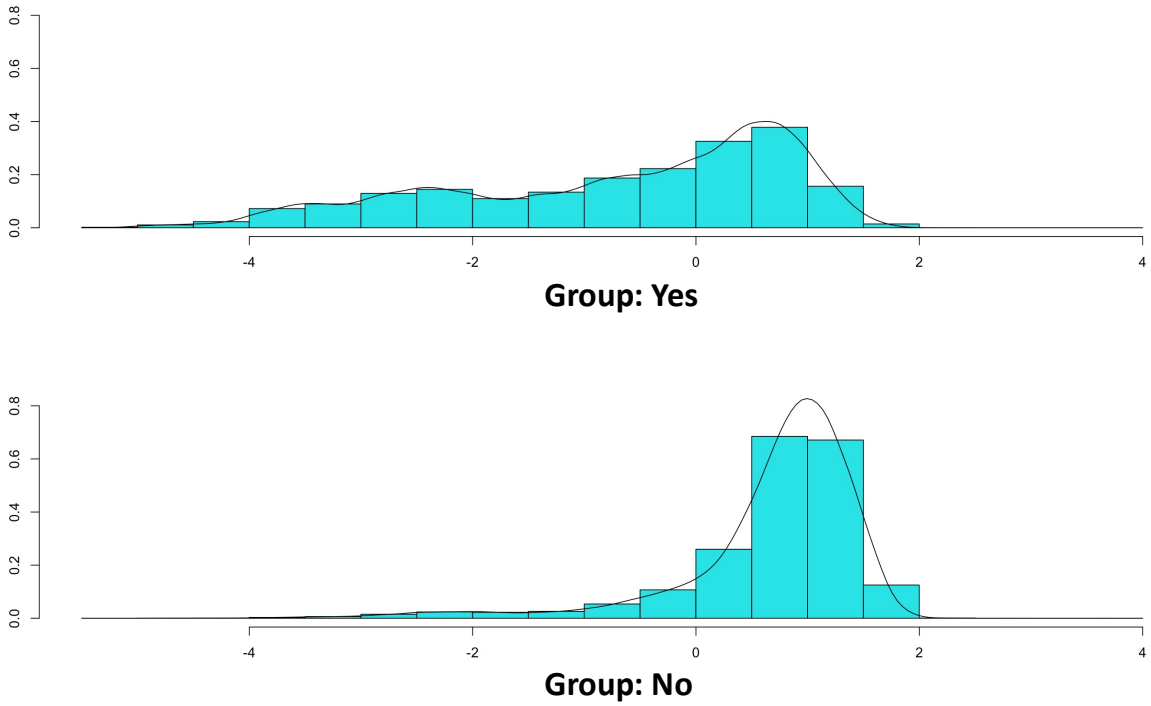


Figure 1. Stacked histograms of linear discriminant values for the minority (top) and majority (bottom) class.

A similar model based upon QDA was also explored. QDA is considered more flexible than LDA as it does not require that the classes have equal covariances matrices. The relaxation of this requirement ultimately allows for the derivation of non-linear decision boundaries, which may be a better option for more complex data. Interestingly, the use of a more flexible QDA model in this case improved model sensitivity from 35 to 48%, allowing for more reliable prediction of the class of interest.

Table 5. Confusion Matrix for QDA Model - Test Data

		Predicted	
		Yes	No
Actual	Yes	12799	16695
	No	13965	92938

Decision Trees. Decision trees present a classification algorithm that is more flexible than that of logistic regression or discriminant analysis. Decision trees perform classification via an algorithm that recursively splits or divides the data space based upon a reduction in Gini impurity or entropy. By allowing for the multiple bisection of the data space, decision trees can effectively

establish more complex and nonlinear decision boundaries, which may ultimately be an advantage for more complex datasets with high dimensionality. In contrast, logistic regression and linear discriminant establish more rigid linear decision boundaries. The increased flexibility of a decision tree can however lead to overfitting, where the model learns patterns specific to the training data set and fails to generalize. Additionally, decision trees are greedy algorithms that will establish decision boundaries sequentially based upon the most optimal information gain for the specific split in question and not for the model as a whole. The greedy nature of the algorithm may/will prove problematic for imbalanced datasets, such as the one explored here.

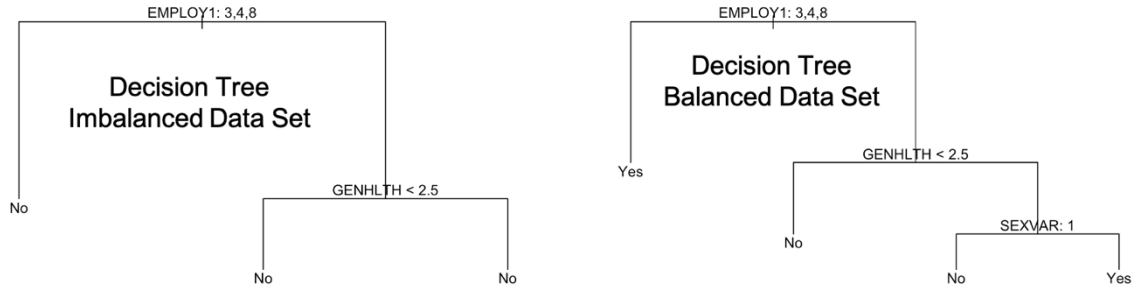


Figure 2. Decision tree models for imbalanced (as-is) and balanced data sets.

Initial efforts were focused on deriving a simple decision tree models capable of reliably predicting the diagnosis of depression based upon the selected risk factors/predictors. The initial decision trees derived on the imbalanced dataset were surprisingly simple and were comprised of a small number of nodes with splits based mainly upon *Employment Status* and *General Health*. These initial trees had misclassification rates that were equivalent to that observed from the null model (19.5%), had a sensitivity and specificity of 0 and 100%, and were incapable of reliably predicting the minority class (Yes for Depression Diagnosis). It was surmised that the poor predictive accuracy of these initial models was due largely in part to the imbalanced nature of the dataset. It was apparent that the greedy nature of the decision tree algorithm produced splits that were heavily biased towards the majority class and thus failed to learn the "patterns" associated with minority class. To address this, the dataset in question was rebalanced by randomly under-sampling observations from the majority class, resulting in a 50/50 dataset with a total of 106,826 observations. Fitting a decision tree with a more balanced dataset resulted in a misclassification rate that was better than the null model (36.2 vs. 50%) and improved the sensitivity rate from 0 to 54%. While still modest in performance, this effort highlights the potential impact of an imbalanced datasets on decision tree outcomes.

Bagged Trees. It is possible via bootstrapping to assemble or aggregate multiple weak decision trees into a stronger model. Bagging, or bootstrap aggregating, of decision trees can help reduce variance and improve overall model performance. For the purposes of this effort, a bagged decision tree using all predictors (mtry=17) was derived utilizing the previously balanced data set. The end result was a model with a relatively elevated out of bag error rate of 35%, but with an improvement in sensitivity from 54 to 65%, relative to the simpler decision tree. The aggregation of multiple decision trees derived from subsets of the data have resulted in a modest improvement

in predictive accuracy for the class of interest. Variable importance plots derived from this model provide further interpretability and information about the association between depression diagnosis and select predictors. For instance, the variable importance plot in Figure 3 indicate that *General Health Status, Employment Status, Gender* and *Adverse Childhood Experiences (Living with Depressed Individual)* are the top predictors that have the most influence on model accuracy.

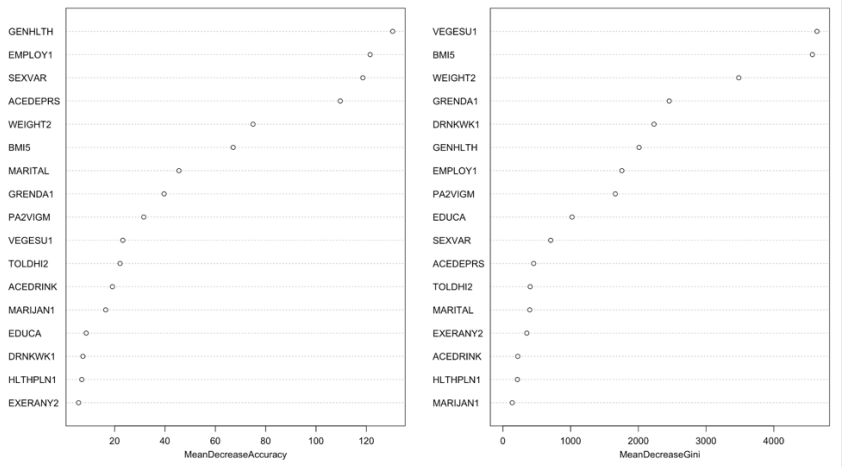


Figure 3. Variable Importance Plot for Bagged Tree (mtry=17).

Further improvements in the bagged decision tree model may be possible by reducing the mtry parameter to a value of 4. This will force the algorithm to take a random sampling of 4 predictor variables as candidates for each split. This method, also known as Random Forest, essentially decorrelates individual trees and reduces the potential influence of strong predictors on the overall model accuracy. A random forest model was derived utilizing the previously balanced dataset in the hopes of further improving accuracy and sensitivity. The end result was a model that performed similarly to the fully bagged decision tree model above. There was very little change in model accuracy and sensitivity. The lack of improvement is not unexpected as the dataset does not seemingly contain strong or correlated predictors that would ultimately benefit from tree decorrelation.

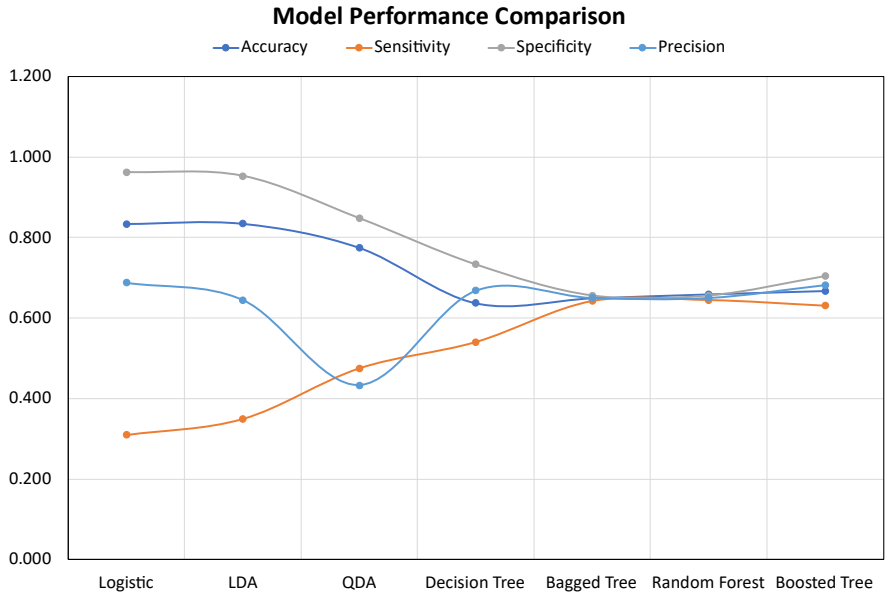
Boosted Trees. The above bagged and random forest trees perform relatively well due to the aggregation of weak learners into a stronger model. Algorithms based upon boosted trees work similarly but improve performance by ensuring that weak learners are derived sequentially and are improved upon iteratively by focusing primarily upon observations that were mis-classed. The algorithm is a slow learner, but after multiple iterations error rates will ultimately decline and reach a value that could be superior to that observed for random forest. Such an algorithm may be ideal for this data set which has weakly significant predictors.

A boosted tree was derived using the GBM package in R and the balanced data set. The model utilized 5000 trees to ensure it reaches an optimal error rate. Additionally, the interaction depth was set at 1; a value that ensures that the individual decision trees have one level (i.e., decision stumps). Although in need of further tuning (i.e. learning rates, interaction depth), a decision tree model based upon a boosted algorithm ultimately improves accuracy from 65 to 67% and

decreases sensitivity from 64 to 63%. Ultimately the derived boosted model performs similarly to that observed for the model based upon random forest.

Validity & Reliability Assessment. The validity of the previous models can be assessed by an inspection of various performance metrics derived via application of the model to a test data set. In this case, key metrics include the following: (1) accuracy - # of cases correctly classified, (2) sensitivity-proportion of positive class correctly identified, (3) specificity – proportion of negative class correctly identified, and (4) precision - ratio of correct positive predictions to the total predicted positives. As indicated above, this effort to derive capable classification models was initially plagued with low sensitivity issues due in large part to the class imbalance issue. Utilization of more flexible classifiers provided more discrimination for the positive class, but this generally came at the cost of accuracy and specificity. A great deal of performance improvement was observed upon rebalancing the dataset, but it is unknown whether these models would generalize well to a population that would be inherently unbalanced. Overall, models based upon bagged trees appear to strike an optimal balance and provide a means to reliably predict a depression diagnosis based upon certain risk factors.

.



Detailed Findings – Regression Models

Introduction. Part two of this final project deals with a problem that is quantitative in nature but similar to that posed for the classification models. Specifically, the proceeding focused primarily on deriving reliable and accurate statistical models capable of predicting the severity of depression within a large population. The main goal was to identify statistically relevant predictors or risk factors that could help quantify the extent of depression symptoms within individuals.

To enable this effort, an aggregated dataset was formed from various modules contained within the 2017 NHANES data set. The quantitative response variable of interest was the PHQ-09 depression severity score determined via the summation of scores related to the presence of certain behaviors and traits within individuals. The PHQ-09 score is a numeric discrete (but ordered) variable shown to be directly correlated to intensity of a depressive disorder. This effort focused mainly upon establishing the relationship (if any) between the PHQ-09 score and various predictor variables related to an individual's demographics, psychographics, general health status and sleep patterns. The final list of dependent and independent variables for this modeling effort are shown in Figure 4. Further details concerning the origin of these variables are provided in the codebook for the 2017 NHANES survey. The clean and aggregated dataset used for this effort had 3832 observations and 24 variables.

Variable ID	Description	Variable ID	Description
DPQ010-090	Depression Scores*	ALQ120	# of Alcoholic Drinks per Day*
RIAGENDER	Gender	HSD010	General Health Status
RIAAGEYR	Age*	DUQ200	Ever Used Marijuana
RIDRETH	Race	INDFMMPI	Poverty Index*
DMDHREDZ	Education Level	OCQ180	Hours Worked per Week*
DMDHRMAZ	Marital Status	PAD680	Minutes of Sedentary Activity per Day*
INDFMIN2	Income*	PAD675	Minutes of Recreational Activity per Day*
BMXWT	Body Weight*	SLD012	Sleep Hours/Day - Week*
BMXHT	Body Height*	SLD013	Sleep Hours/Day - Weekend*
BMXBMI	Body Mass Index*	SLQ050	Ever Have Trouble Sleeping
BPQ080	Cholesterol Diagnosis	SLQ120	How Often Do you Feel Sleepy/Day*

Figure 4. Independent and dependent variables chosen for this regression study. Variables appended with an asterisk are continuous in nature.

Multiple Linear Regression. One of the classic and most well statistical models used to provide predictions for continuous quantitative variables is multiple linear regression. The popularity of this statistical model originates from its ease of implementation and interpretability. It can accommodate both categorical and continuous independent variables. The model seeks to define a linear function (or relationship) between the dependent variable of interest and multiple independent variables by defining a line (or hyperplane) of best fit using standard least squares techniques. The approach will work well for data with linear patterns, but will introduce high levels of bias for datasets that are non-linear. For this reason, the model is considered relatively rigid but because of its parametric nature it affords a high degree of interpretability.

For this project, a multiple linear regression involving all of the dependent variables listed in Figure 4 was initially derived. It is generally not desirable to build a complex regression with a large number of predictors. However, part of this effort entailed utilizing various feature selection or reduction methodologies to result in a more parsimonious.

First a full linear model was fit using all variables and the entire dataset. The model that was derived had a modest residual standard error of 3.5 (average deviation of predicted values from the true regression line). The model did however have a relatively high F-statistic of 56.35, a value sufficiently far from 1 (p-value of <2.2e-16) to suggest that one cannot reject the null hypothesis that none of the variables in the model are significant. Individually, we observe a total of 9 variables that have p-values less than 0.05 and are thus deemed statistically significant. All other variables were eliminated from future models.

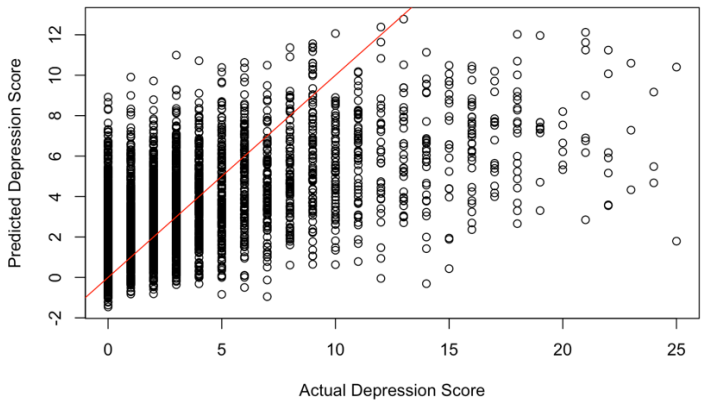


Figure 5. Actuals vs. Predicted Values for Full Linear Regression Model

A plot of the predicted values versus actual values was derived for this model using the entirety of the dataset. The plot indicates a fairly high amount of scatter and suggests that a linear model may not be entirely appropriate for this data set. The residuals for this model are not normal which will ultimately cause issues with establishing and interpreting a linear model. The issue arises mainly out of the fact that the dependent variable of interest is right-skewed and zero inflated. Transformation of this variable (via square root) somewhat improves the fit, but the residuals were still non-normal.

The mean squared error associated with this full model was calculated to be 12.14, which is somewhat lower than the 17.9 calculated for the null model (prediction = average depression severity score). The reduction in MSE and the relatively modest adjusted R^2 value of 0.31 indicate the simple linear model with all predictors does have some predictive power. However, the model is not particularly parsimonious and thus impedes interpretability, suggesting that such a model could ultimately benefit from feature selection (via subsetting), regularization (via ridge or lasso regression) or dimensional reduction (via PCR or PLS).

Multiple Linear Regression – Subsets. To simplify the full linear model, feature selection via best subset selection was conduction. Best subset selection is a powerful function in R that iteratively determines the best combination of predictors for each model size. A final selection of the best (and most parsimonious model) is then determined by cross validation or by examining various test metrics (Cp, AIC, BC or adjusted R2). For our dataset, all four-test metrics were considered as selection criteria. Ultimately it was decided to utilize BIC as the metric to enable model subset selection. Shown in Figure 6 are the determined BIC values as a function of model size, which clearly show a minimum at p=15.

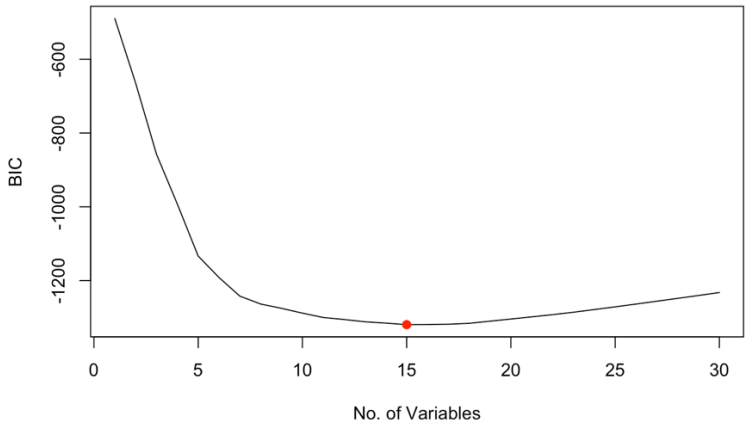


Figure 6. BIC Values vs. Model Size - Sub setting

It is thus evident that the sub setting exercise suggests that one can further reduce the number of variables in the dataset from 22 to 14, affording a much simpler linear regression model. The subset of variables chosen include the following: A*ge, Race, Marital Status, Income, Body Mass Index, Cholesterol Diagnosis, # of Alcoholic Drinks per Day , General Health Status, Drug Use, Poverty Index, , Hours of Sleep , Trouble Sleeping , Times per Day of Sleepiness.* Re-fitting a linear model to this subset of variables resulted in a mean squared error of 12.2 and an R2 value of 0.3125. These values are considerably better than the NULL model and are essentially unchanged relative to a model with all predictors. The subset effort thus allowed for a more parsimonious linear regression model with no decline in predictive accuracy.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.53	0.57	9.65	< 2e-16
HSD010Poor	4.85	0.40	12.01	< 2e-16
SLQ120Often	2.32	0.21	11.18	< 2e-16
HSD010Fair	2.30	0.24	9.70	< 2e-16
SLQ050No	-2.07	0.14	-14.96	< 2e-16

Figure 7. Pareto of Estimates (Absolute Value) for Optimized Linear Model

One of the most powerful aspects of multiple linear regression is its interpretability. In return for sacrificing some level of flexibility, the model allows one to fully quantify the impact of select variables on the response. For instance, the table in Figure 7 identifies those variables with greatest absolute impact on depression score. The coefficients of these estimates allow one to determine the direction and magnitude of this impact. In this case, the categorical variable HSD010 (General Health Status) is listed twice and indicates that those individuals that have a general health status of poor will tend to have an increase in depression severity score of 5.53, with all other variables constant. Similarly, those respondents that do not have trouble sleeping (SLQ050) will have decrease in depression score of -2.07. This interpretability is a powerful aspect of linear regression and allows the researcher to quantify the effect that each dependent variable has on depression score.

Lasso and Ridge Regression – Although not primarily intended as a means for feature selection, the regularization lasso technique can help determine and eliminate insignificant or weakly significant dependent variables. Regularization techniques such as lasso and ridge regression primarily intend to improve model performance and avoid overfitting, by imposing constraints and complexity penalties to the model. The overarching goal is to reward sparsity and penalize complexity, essentially trading off a small amount of bias to improve variance and avoid overfitting. For lasso regression, this can result in shrinking the coefficients of some dependent variables to zero, essentially providing an alternative means for feature selection. Similarly, ridge regression also imposes shrinkage, but the coefficients never equal zero.

For the purposes of this exercise, both ridge and lasso regression were applied to the full dataset that included all of the predictors listed in Figure 4. Specifically, a ridge regression model was derived with an optimal lambda value of 0.207 as determined via cross-validation. This model resulted in a mean squared test error of 12.5, a value that was nearly identical to that observed for the optimized linear fit. In this case, it was evident that although ridge regression shrunk select coefficients, it did not improve the predictive accuracy of a linear model.

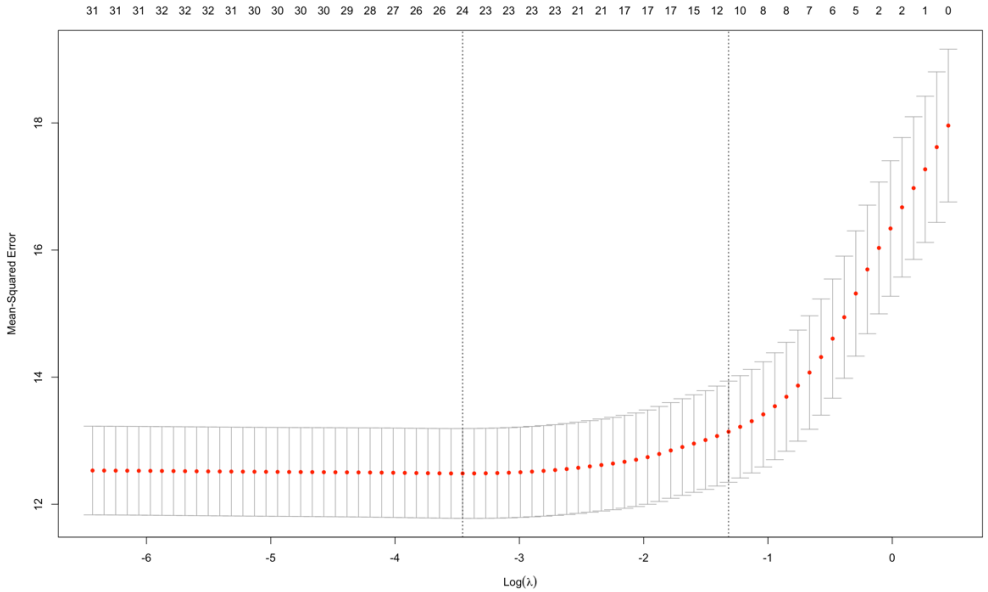


Figure 8. Cross-validated mean square error as a function of log-lambda for a model based on Lasso Regression.

A similar exercise was attempted with lasso regression. Specifically, a model was derived utilizing the original dataset with a full set of predictors. An optimal lambda (shrinkage parameter) was determined via cross-validation (Figure 8) and a full model was derived. The test mean squared error of this lasso model was determined to be 12.51. This value unfortunately doesn't differ much from that observed for the linear or ridge regression fits. However, the lasso regression has successfully shrunk the coefficients of a total of 5 (out of 22) variables to value of zero – essentially removing them from the fit entirely. These five variables were the following: *Education Level, Body Weight, Body Height, Minutes of Sedentary Activity, Sleepiness* (one dummy level).

Overall, both Lasso and Ridge regression have seemingly reduced the complexity of a linear regression, but neither approach resulted in a significant improvement in test mean squared error.

Random Forest Regression. It was surmised that the diminished performance of the linear models above was largely attributed to the nonlinear nature of the data. Non-parametric and more flexible algorithms such as decision trees thus may afford models with improved accuracy. As such, a series of random forest models were derived using the full data set and varying mtry values, ranging from 1 to 6. An optimal model using an mtry value of 4 afforded a test mean squared error of 12.9, a value very close to that observed for the linear models.

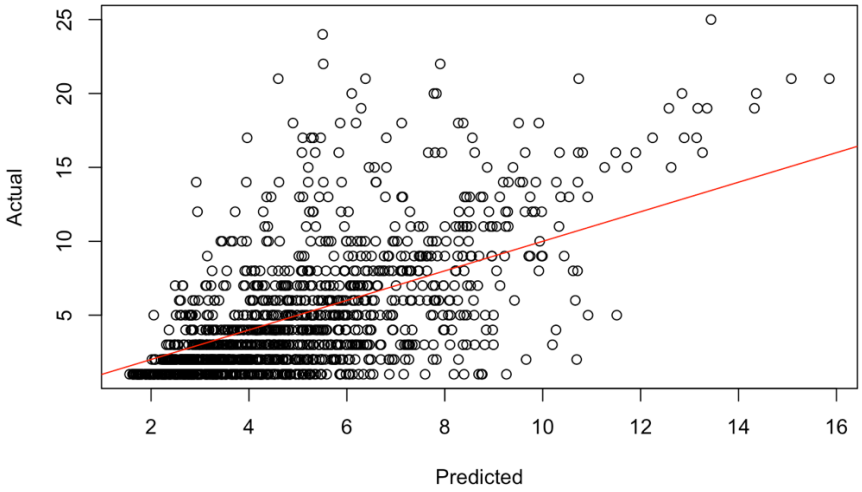
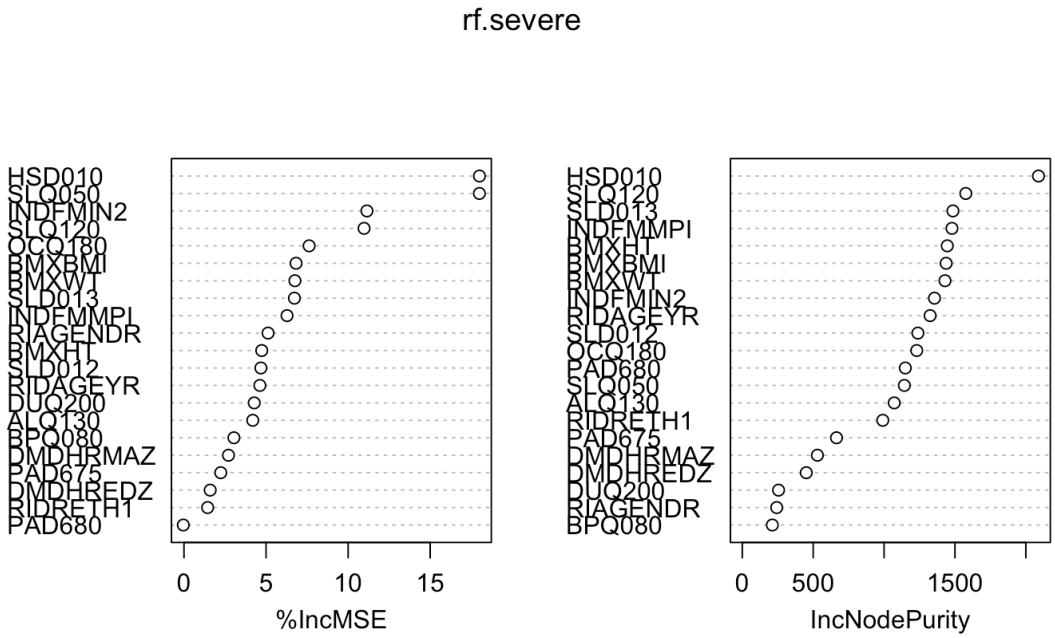


Figure 9. Actual vs. Predicted Test Values for a Random Forest Model - Non-Zero Depression Scores

It is likely that both the linear models and non-parametric decision tress are significantly affected by the significant right-handed skew and high level of zeros for the *Depression Score* variable. In order to probe this effect further, observations with a *Depression Score* of zero were removed from the data set a new random forest model was derived. The end result was a more accurate model with an effective R2 value of 0.65 (Figure 9). A variable importance plot is included below which suggests that general health status, sleeping hours and income are all important predictors for Depression Score.



Detailed Findings – Principal Components Regression (PCR) and Partial Least Squares (PLS)

Introduction. The final part of this project entailed the use of both PCR and PLS on high dimensional datasets in order to demonstrate the utility of these approaches to affect dimensional reduction and provide for robust models. The original intent was to build models utilizing the same aggregated BRFSS data set above, with the primary goal of building models to predict the severity of depression (i.e. depression score). However, all PCR and PLS models derived with this data set consistently afforded a result that was not better than the null model. It was determined that the high frequency of zero values were biasing the results and affording a model with no predictive power.

Subsequent efforts to model the severity of depression symptoms as a function of dietary intake via principal components regression or partial least squares were also largely unsuccessful. The models obtained did not improve predictive accuracy relative to that observed for a standard linear fit and that observed solely for the NULL model. This highlights the difficulty in developing predictive models for human behavior and mental health. Additionally, despite best attempts, the presence of several confounding variables (geography, race, gender, age, health issues, income) could also be significantly affect the predictive accuracy of such a model.

To demonstrate the utility of both PCR and PLS, a new problem was explored. Specifically, the NHANES extensive data set on dietary intake can allow one to predict and determine the relationship between dietary intake and body mass index. Both PCR and PLS are ideal for this effort, as the dataset contains a wealth of continuous and potentially collinear variables that may ultimately complicate models based upon standard least squares regression.

The 2014 version of the NHANES dataset contains a comprehensive module of dietary intake for nearly 8000 individuals. This dietary intake data contains over 67 different measurables ranging from caloric intake, grams of protein, grams of fat and grams of sugar. An aggregated version of this data was derived that affords the total daily intake of each one of these dietary metrics. For the purposes of PCR and PLS, models were built with the goal of predicting body mass index (BMI) based upon diet. For instance, would PCR and PLS be capable of discerning diets responsible for either healthy or unhealthy BMI values?

Principal Components Regression. PCR is a statistical modeling technique that is very similar to linear regression but utilizes a principal components analysis to reduce the original variables to a smaller subset of principal components. In the case of PCR, the principal components are derived to capture the maximal amount of variation in predictors within the data space. As such, the technique can prove useful in dimensional reduction and can simplify the regression onto a select number of principal components. For the purposes of this project, a PCR model was derived utilizing a dataset of dietary intake that had a total of 67 variables and 8000 observations. The high dimensionality and potential presence of collinear variables make this a task well suited for PCR.

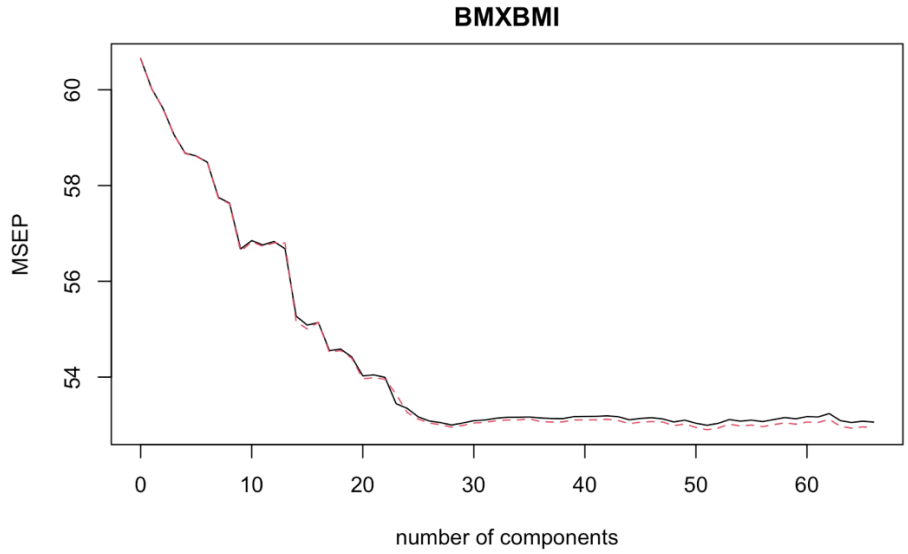


Figure 10. Cross-validated test error as a function of the number of components for the PCR model.

The optimal number of components for a PCR fit with this dataset, determined via cross validation, was 28. As such, this PCR approach was successful in feature reduction and provided for a less complex model. The test MSE for the PCR model with 28 components was 56.8, a value that was only marginally better than that observed for the null model (62.5).

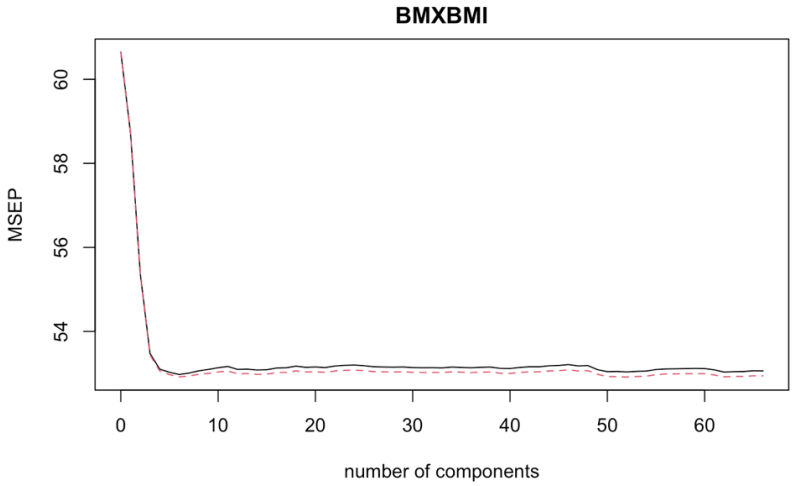


Figure 11. Cross-validated test error as a function of the number of components for the PLS model.

Partial Least Squares. Partial least squares regression is similar to PCR but derives principal components to capture variation from not only the independent variables but also from the dependent variable of interest. The increased relevance to the predictor can reduce bias and

improve overall prediction accuracy. A partial least squares regression was derived for the dietary intake data in an effort to predict BMI values. The optimal number of components for a PLS fit with this dataset, determined via cross validation, was 5. In this case, the use of only 5 components captures 55.3% and 14.4% of the variance associated with the independent and dependent variables, respectively. The test MSE associated with a full PLS model with only 5 components exhibits a mean squared error of 56.6, a value that is moderately better than that observed for the null model. In general, the PLS fit was successful in reducing 66 dependent variables down to only 5 principal components, while still providing for some level of predictive accuracy.

Summary and Conclusions

The preceding three-part project was primarily focused on deriving statistical models that elucidate the relationship between key risk factors/predictors and the diagnosis or severity of a depressive disorder. Of particular interest was identifying key demographic, psychographic, medical and dietary that influence the occurrence and severity of depression. Classification models based upon logistic regression, discriminant analysis and decision trees were explored extensively as a means to predict a depression diagnosis. Models based upon the original dataset resulted in relatively high accuracy and low sensitivity, due mainly to a fairly severe imbalance issue. Models based on discriminant analysis improved sensitivity moderately but still suffered from this issue. Data set balancing via an under sampling of the majority class resulted in the most significant improvement in sensitivity without compromising the overall accuracy of the model. Overall, classification models based upon bagged trees (random forest) provided an ideal combination of both accuracy and sensitivity.

The second part of the project focused on deriving the relationship between various predictors/risk factors and the severity of a depressive disorder. For the purposes of this exercise, a numerical discrete response (Depressive Score) was modeled utilizing multivariate linear regression using a large number of dependent categorical and continuous variables. Linear regression models afforded mean square error values moderately better than the null model, indicating that they possess some level of predictive accuracy. Attempts to improve the model complexity and performance via sub-setting and regularization were largely successful. The end result of this effort was an interpretable linear regression model.

Appendix A – BRFSS Data Processing (Classification Data)

```
# BRFSS 2017 Data Processing - Used for Classification
library(rio)
library(tidyverse)
setwd("~/Desktop/Predictive Modeling Final Project")
df <-import("LLCP2019.XPT") # large file, takes 30-60seconds

# remove underscores
X <- names(df)
X <- sub("_", "", X)
names(df) <- X

# Select Key Variables
# Demographics, Employment, Income, Exercise, Adverse Childhood, Dietary
df.select <- dplyr::select(df, SEQNO, ADDEPEV3, SEXVAR, GENHLTH, MENTHLTH, HLTHPLN1,BPHIGH4, TOLDHI2, MARITAL, EDUCA,
EMPLOY1, INCOME2, WEIGHT2, EXERANY2, ACEDEPRS, ACEDRINK, MARIJAN1, BMI5, DRNKWK1, PA2VIGM, FRUTDA2, GRENDA1, VEGESU1)

remove(df, X)

# Ever told you have a depressive disorder
df.select <- filter(df.select, ADDEPEV3 !=9)
df.select <- filter(df.select, ADDEPEV3 !=7)
df.select$ADDEPEV3 <- df.select$ADDEPEV3 %>% replace_na(2)
df.select$ADDEPEV3 <- as.factor(df.select$ADDEPEV3)
levels(df.select$ADDEPEV3) <- c("Yes", "No")


# Gender
df.select$SEXVAR <- as.factor(df.select$SEXVAR)
levels(df.select$SEXVAR) <- c("Male", "Female")

# General Health
df.select <- filter(df.select, GENHLTH !=9)
df.select <- filter(df.select, GENHLTH !=7)
df.select$GENHLTH <- df.select$GENHLTH %>% replace_na(2)
levels(df.select$GENHLTH) <- c("Excellent", "VeryGood",
                               "Good", "Fair", "Poor")

# No of Days per Month Mental Health is not good
df.select <- filter(df.select, MENTHLTH !=99)
df.select <- filter(df.select, MENTHLTH !=77)
df.select$MENTHLTH[df.select$MENTHLTH==88] <- 0

# On a HealthPlan?
df.select <- filter(df.select, HLTHPLN1 !=9)
df.select <- filter(df.select, HLTHPLN1 !=7)
df.select$HLTHPLN1 <- df.select$HLTHPLN1 %>% replace_na(1)
df.select$HLTHPLN1 <- as.factor(df.select$HLTHPLN1)
levels(df.select$HLTHPLN1) <- c("Yes", "No")

# Blood Pressure High?
df.select <- filter(df.select, BPHIGH4 !=9)
df.select <- filter(df.select, BPHIGH4 !=7)
df.select$BPHIGH4[df.select$BPHIGH4==2] <- 3 #Pregnancy Special Case
df.select$BPHIGH4 <- df.select$BPHIGH4 %>% replace_na(3)
df.select$BPHIGH4 <- as.factor(df.select$BPHIGH4)
levels(df.select$BPHIGH4) <- c("Yes", "No", "Borderline")

# Told had high cholesterol
df.select <- filter(df.select, TOLDHI2 !=9)
df.select <- filter(df.select, TOLDHI2 !=7)
df.select$TOLDHI2 <- df.select$TOLDHI2 %>% replace_na(2)
df.select$TOLDHI2 <- as.factor(df.select$TOLDHI2)
levels(df.select$TOLDHI2) <- c("Yes", "No")

# Marital Status
df.select$MARITAL[df.select$MARITAL==3] <- 2
df.select$MARITAL[df.select$MARITAL==4] <- 2 # Special Case
df.select$MARITAL[df.select$MARITAL==5] <- 2 # Special Case
df.select$MARITAL[df.select$MARITAL==6] <- 2 # Special Case
df.select <- filter(df.select, MARITAL !=9)
df.select$MARITAL <- as.factor(df.select$MARITAL)
levels(df.select$MARITAL) <- c("Married", "NotMarried")

# Education Level
df.select <- filter(df.select, EDUCA !=9)
df.select$EDUCA <- df.select$EDUCA %>% replace_na(6)
df.select$EDUCA <- as.factor(df.select$EDUCA)
levels(df.select$EDUCA) <- c("None", "Elementary", "SomeHS", "HSGrad", "SomeColl", "CollGrad")

# Employment Status
df.select <- filter(df.select, EMPLOY1 !=9)
df.select$EMPLOY1 <- df.select$EMPLOY1 %>% replace_na(1)
```

```
df.select$EMPLOY1 <- as.factor(df.select$EMPLOY1)
levels(df.select$EMPLOY1) <- c("Employed", "SelfEmployed",
                               "UnempShort", "UnempLong", "HomeMaker",
                               "Student", "Retired", "Unable")

# Income
df.select <- filter(df.select, INCOME2 !=99)
df.select <- filter(df.select, INCOME2 !=77)
df.select$INCOME2 <- df.select$INCOME2 %>% replace_na(8)
df.select$INCOME2 <- as.factor(df.select$INCOME2)
levels(df.select$INCOME2) <- c("<10K,", "10-15K", "15-20K", "20-25K",
                               "25-35K", "35-50K", "50-75K",">75K")

# WEIGHT2
df.select <- filter(df.select, WEIGHT2 !=9999)
df.select <- filter(df.select, WEIGHT2 !=7777)
df.select <- filter(df.select, WEIGHT2 !=999)
df.select <- mutate(df.select,
                    WEIGHT2 = case_when( WEIGHT2>8999 ~ (WEIGHT2-9000)*2.2,
                                         TRUE ~ WEIGHT2))
df.select$WEIGHT2 <- df.select$WEIGHT2 %>%
  replace_na(median(df.select$WEIGHT2, na.rm = TRUE))

# EXERANY2 - Did you exercise in the last 30 days
df.select <- filter(df.select, EXERANY2 !=9)
df.select <- filter(df.select, EXERANY2 !=7)
df.select$EXERANY2 <- df.select$EXERANY2 %>% replace_na(1)
df.select$EXERANY2 <- as.factor(df.select$EXERANY2)
levels(df.select$EXERANY2) <- c("Yes", "No")

# ACEDEPRS - Ever lived with someone suicidal
df.select$ACEDEPRS <- df.select$ACEDEPRS %>% replace_na(2)
df.select <- filter(df.select, ACEDEPRS !=9)
df.select <- filter(df.select, ACEDEPRS !=7)
df.select$ACEDEPRS <- as.factor(df.select$ACEDEPRS)
levels(df.select$ACEDEPRS) <- c("Yes", "No")

# ACEDRINK - Ever lived with an alcoholic?
df.select$ACEDRINK <- df.select$ACEDRINK %>% replace_na(2)
df.select <- filter(df.select, ACEDRINK !=9)
df.select <- filter(df.select, ACEDRINK !=7)
df.select$ACEDRINK <- as.factor(df.select$ACEDRINK)
levels(df.select$ACEDRINK) <- c("Yes", "No")

# MARIJAN1 - days per month of MJ use
df.select$MARIJAN1 <- df.select$MARIJAN1 %>% replace_na(0)
df.select <- filter(df.select, MARIJAN1 !=99)
df.select <- filter(df.select, MARIJAN1 !=77)
df.select <- filter(df.select, MARIJAN1 !=30) # outlier
df.select$MARIJAN1[df.select$MARIJAN1==88] <- 0

# BMI5 - BMI Continuous
df.select$BMI5 <- df.select$BMI5 %>%
  replace_na(median(df.select$BMI5, na.rm = TRUE))
df.select$BMI5 <- df.select$BMI5/100

# DRNKWK1 - # alcoholic drinks per week, calculated
df.select$DRNKWK1[df.select$DRNKWK1==99900] <-
  median(df.select$DRNKWK1, na.rm = TRUE)

# PA2VIGM - Calculated min of exercise per week
df.select$PA2VIGM <- df.select$PA2VIGM %>% replace_na(0)
df.select$PA2VIGM <- df.select$PA2VIGM/10

# VEGESU1 - Calculated Veggies per Day
df.select$VEGESU1 <- df.select$VEGESU1 %>% replace_na(0)
df.select$VEGESU1 <- df.select$VEGESU1/100
```

Appendix B – NHANES Data Processing (Regression Models)

```
# NHANES 2017-2018 data at the following URL
# Used this data due to pandemic issues and incomplete data sets for 2019-2020 data
# Variable Descriptions - https://wwwn.cdc.gov/nchs/nhanes/search/variablelist.aspx?Component=Demographics

library(rio)
library(tidyverse)
setwd("~/Desktop/Predictive Modeling Final Project")
demo.df <-import("DEMO_J.XPT") # Demographics

depr.df <-import("DPQ_J.XPT") # PHQ-09 Data

# remove underscores (if necessary)
X <- names(demo.df)
X <- sub("_", "", X)
names(demo.df) <- X

# Goal here is to build models to reliably predict the severity of depression symptoms
## Predictors can include...
### Weight, BMI, Age, Gender, Chronic Health Issues, Income, Physical Activity, etc.

# Need to calculate PHQ-09 Severity Score from DPQ (depression patient questionnaire)
# These are my depressed observations

# Lets pull the label descriptions from each variable
DEPR.labels <- rep(1:ncol(depr.df),1)

for (i in 1:ncol(depr.df)){
  lab <- depr.df[,i] %>% attr('label')
  DEPR.labels[i] <- lab
}

Depression.Labels <- data.frame("variable_name" = colnames(depr.df), "variable_description" = DEPR.labels)

write.csv(Depression.Labels, "DEPR.Labels.csv")

# Add total severity score
library(sticky) # maintain attribute labels
depr.df <- depr.df[,-11]
depr.df <- na.omit(depr.df)
depr.df$DepressionScore = rowSums(depr.df[,-1])
depr.df <- filter(depr.df, DepressionScore<27)

depr.df <- depr.df %>%
  mutate(DepressionLabel= case_when(
    DepressionScore <= 4 ~ 'None',
    DepressionScore > 4 & DepressionScore <= 9 ~ 'Mild',
    DepressionScore > 9 & DepressionScore <= 14 ~ 'Moderate',
    DepressionScore > 14 & DepressionScore <= 19 ~ 'Moderately Severe',
    DepressionScore > 19 & DepressionScore <= 27 ~ 'Severe',)
  )

depr.df<- dplyr::select(depr.df, SEQN, DepressionScore, DepressionLabel)

# Look at Histogram of Total Severity Scores
Tabulated.Scores <- depr.df %>% group_by(DepressionLabel) %>%
  summarize (frequency = n())
write.csv(Tabulated.Scores, "TabulatedScores.csv")

demo.df <- sticky_all(demo.df)
demo.df <- dplyr::select(demo.df, SEQN, RIAGENDR, RIDAGEYR,
                  RIDRETH1, DMDHREDZ, DMDHRMAZ, INDFMIN2)

# Pull Demo Variable Labels
demo.labels <- rep(1:ncol(demo.df),1)

for (i in 1:ncol(demo.df)){
  lab <- demo.df[,i] %>% attr('label')
  demo.labels[i] <- lab
}

Demo.Labels <- data.frame("variable_name" = colnames(demo.df), "variable_description" = demo.labels)

write.csv(Demo.Labels, "DemoLabels.csv")

# Use to pull bodymass index, weight, height
bmx.df <-import("BMX_J.XPT")
bmx.df <- sticky_all(bmx.df)
bmx.df <- dplyr::select(bmx.df, SEQN, BMXWT, BMXHT, BMXBMI)

bmx.labels <- rep(1:ncol(bmx.df),1)

for (i in 1:ncol(bmx.df)){
  lab <- bmx.df[,i] %>% attr('label')
```

```
  bmx.labels[i] <- lab
}

bmx.Labels <- data.frame("variable name" = colnames(bmx.df), "variable description" = bmx.labels)

write.csv(bmx.Labels, "bmxLabels.csv")

chol.df <-import("BPQ_J.XPT")
chol.df <- sticky_all(chol.df)
chol.df <- dplyr::select(chol.df, SEQN, BPQ080)

chol.labels <- rep(1:ncol(chol.df),1)

for (i in 1:ncol(chol.df)){
  lab <- chol.df[,i] %>% attr('label')
  chol.labels[i] <- lab
}

chol.Labels <- data.frame("variable_name" = colnames(chol.df), "variable_description" = chol.labels)

write.csv(chol.Labels, "cholLabels.csv")

Alcohol.df <- import("ALQ_J.XPT")
Alcohol.df <- sticky_all(Alcohol.df)
Alcohol.df <- dplyr::select(Alcohol.df, SEQN, ALQ130)
sum(is.na(Alcohol.df$ALQ130)) # lots of NA's, assume zero?
Alcohol.df[is.na(Alcohol.df)] <- 0


Alcohol.labels <- rep(1:ncol(Alcohol.df),1)

for (i in 1:ncol(Alcohol.df)){
  lab <- Alcohol.df[,i] %>% attr('label')
  Alcohol.labels[i] <- lab
}

Alcohol.Labels <- data.frame("variable_name" = colnames(Alcohol.df), "variable_description" = Alcohol.labels)

write.csv(Alcohol.Labels, "AlcoholLabels.csv")

genhealth.df <-import("HSQ_J.XPT")
genhealth.df <- sticky_all(genhealth.df)
genhealth.df <- dplyr::select(genhealth.df, SEQN, HSD010)
sum(is.na(genhealth.df$HSD010))

genhealth.labels <- rep(1:ncol(genhealth.df),1)

for (i in 1:ncol(genhealth.df)){
  lab <- genhealth.df[,i] %>% attr('label')
  genhealth.labels[i] <- lab
}

genhealth.Labels <- data.frame("variable_name" = colnames(genhealth.df), "variable_description" = genhealth.labels)

write.csv(genhealth.Labels, "genhealthLabels.csv")

SunEx.df <- import("DEQ_J.XPT")
SunEx.df <- sticky_all(SunEx.df)
SunEx.df <- dplyr::select(SunEx.df, SEQN, DED120, DED125)
sum(is.na(SunEx.df$DED125))

sunex.Labels <- rep(1:ncol(SunEx.df),1)

for (i in 1:ncol(SunEx.df)){
  lab <- SunEx.df[,i] %>% attr('label')
  sunex.Labels[i] <- lab
}

sunex.Labels <- data.frame("variable_name" = colnames(SunEx.df), "variable_description" = sunex.Labels)

write.csv(sunex.Labels, "sunexLabels.csv")

poverty.df <- import("INQ_J.XPT")
poverty.df <- sticky_all(poverty.df)
poverty.df <- dplyr::select(poverty.df, SEQN, INDFMMPI) # poverty index
sum(is.na(poverty.df$INDFMMPI))

poverty.Labels <- rep(1:ncol(poverty.df),1)

for (i in 1:ncol(poverty.df)){
  lab <- poverty.df[,i] %>% attr('label')
  poverty.Labels[i] <- lab
}

poverty.Labels <- data.frame("variable_name" = colnames(poverty.df), "variable_description" = poverty.Labels)

write.csv(poverty.Labels, "povertyLabels.csv")
```

```
drug.df <- import("DUQ_J.XPT")
drug.df <- sticky_all(drug.df)
drug.df <- dplyr::select(drug.df, SEQN, DUQ200) # poverty index
# Some NA's here

hoursworked.df <- import("OCQ_J.XPT")
hoursworked.df <- sticky_all(hoursworked.df)
hoursworked.df <- dplyr::select(hoursworked.df, SEQN, OCQ180)
hoursworked.df[is.na(hoursworked.df)] <- 0

hoursworked.Labels <- rep(1:ncol(hoursworked.df),1)

for (i in 1:ncol(hoursworked.df)){
  lab <- hoursworked.df[,i] %>% attr('label')
 hoursworked.Labels[i] <- lab
}

hoursworked.Labels <-
  data.frame("variable_name" = colnames(hoursworked.df), "variable_description" = hoursworked.Labels)

write.csv(hoursworked.Labels, "hoursworkedLabels.csv")

sed.df <- import("PAQ_J.XPT")
sed.df <- sticky_all(sed.df)
sed.df <- dplyr::select(sed.df, SEQN, PAD680, PAD675) # time sitting


sed.Labels <- rep(1:ncol(sed.df),1)

for (i in 1:ncol(sed.df)){
  lab <- sed.df[,i] %>% attr('label')
 sed.Labels[i] <- lab
}

sed.Labels <-
  data.frame("variable_name" = colnames(sed.df), "variable_description" = sed.Labels)

write.csv(sed.Labels, "sedLabels.csv")

sleep.df <- import("SLQ_J.XPT")
sleep.df <- sticky_all(sleep.df)
sleep.df <- dplyr::select(sleep.df, SEQN, SLD012, SLD013, SLQ050, SLQ120) # hours of sleep

sleep.Labels <- rep(1:ncol(sleep.df),1)

for (i in 1:ncol(sleep.df)){
  lab <- sleep.df[,i] %>% attr('label')
 sleep.Labels[i] <- lab
}

sleep.Labels <-
  data.frame("variable_name" = colnames(sleep.df), "variable_description" = sleep.Labels)

write.csv(sleep.Labels, "sleepLabels.csv")

# probably need to add some more predictors
# smoking, or something else

vars.chosen <- rbind(sleep.Labels, chol.Labels,
                     genhealth.Labels,Alcohol.Labels,
                     bmx.Labels, demo.labels,
                     hoursworked.Labels, poverty.Labels,
                     sed.Labels, sunex.Labels)

master.df <- merge(depr.df, demo.df, by="SEQN") %>%
  merge(.,bmx.df, by = "SEQN") %>%
  merge(.,chol.df, by = "SEQN") %>%
  merge(.,Alcohol.df, by = "SEQN") %>%
  merge(.,genhealth.df, by = "SEQN") %>%
  merge(.,drug.df, by = "SEQN") %>%
  merge(.,poverty.df, by = "SEQN") %>%
  merge(.,hoursworked.df, by = "SEQN") %>%
  merge(.,sed.df, by = "SEQN") %>%
  merge(.,sleep.df, by = "SEQN")

# Need to clean up
# Recode categorical variables
# Check for data entry errors/outliers
# Masterplan for missing values.

############## FIX DEMO variables##############
###############################################

# Recode GENDER into categorical with appropriate levels
master.df$RIAGENDR <- as.factor(master.df$RIAGENDR)
levels(master.df$RIAGENDR) <- c("Male", "Female")
```

```
# Recode RACE into categorical with appropriate levels
master.df$RIDRETH1 <- as.factor(master.df$RIDRETH1)
levels(master.df$RIDRETH1) <- c("Mex American", "Other Hispanic",
                                "White", "Black", "Other")

# Remove NA's in EDUCATION LEVEL - Replace with MODE
master.df$DMDHREDZ <- master.df$DMDHREDZ %>% replace_na(2)
master.df$DMDHREDZ <- as.factor(master.df$DMDHREDZ)
levels(master.df$DMDHREDZ) <- c("LessThanHS", "HS", "College")

# Remove NA's in MARITAL STATUS - Replace with MODE
master.df$DMDHRMAZ <- as.factor(master.df$DMDHRMAZ)
master.df$DMDHRMAZ <- master.df$DMDHRMAZ %>% replace_na(1)
levels(master.df$DMDHRMAZ) <- c("Married", "Divorced/Sep", "Never Married")

# INDFMIN2 - Annual Family Income
# Remove 77, 99 and missing values
# Replace missing values with mode
master.df <- filter(master.df, INDFMIN2 !=99)
master.df <- filter(master.df, INDFMIN2 !=77)
master.df$INDFMIN2 <- master.df$INDFMIN2 %>% replace_na(15)

################ FIX BMX VARS ##################
################################################

# BMXWT - weight in kilograms
# replace NA's with median values
master.df$BMXWT <- master.df$BMXWT %>%
  replace_na(median(master.df$BMXWT, na.rm = TRUE))

master.df$BMXHT <- master.df$BMXHT %>%
  replace_na(median(master.df$BMXHT, na.rm = TRUE))

master.df$BMXBMI <- master.df$BMXBMI %>%
  replace_na(median(master.df$BMXBMI, na.rm = TRUE))

######################## BPQ_J - Cholesterol ###########
########################################################
master.df$BPQ080[master.df$BPQ080==9] <- 2
master.df$BPQ080 <- as.factor(master.df$BPQ080)
levels(master.df$BPQ080) <- c("Yes", "No")

################ ALQ_J , Alcohol Use ###################
########################################################
master.df <- filter(master.df, ALQ130 !=999)
master.df <- filter(master.df, ALQ130 !=777)

master.df$ALQ130 <- master.df$ALQ130 %>%
  replace_na(median(master.df$ALQ130, na.rm = TRUE))

########################################################
############# HSQ_J - General Health Status ###########
########################################################
master.df <- filter(master.df, HSD010 !=9)
master.df <- filter(master.df, HSD010 !=7)
master.df$HSD010 <- as.factor(master.df$HSD010)
levels(master.df$HSD010) <- c("Excellent", "VeryGood",
                              "Good", "Fair", "Poor")

########################################################
############# INQ_J - Income/Poverty ###################
########################################################
master.df$INDFMMPI <- master.df$INDFMMPI %>%
  replace_na(median(master.df$INDFMMPI, na.rm = TRUE))

########################################################
############# DUQ_J - Drug Use #########################
########################################################
master.df$DUQ200 <- as.factor(master.df$DUQ200)
master.df$DUQ200<- master.df$DUQ200 %>% replace_na(1)
levels(master.df$DUQ200) <- c("Yes", "No")
# replace NA with mode (Yes for MJ use)

########################################################
############# OCQ_J - Hours Worked #####################
########################################################
master.df <- filter(master.df, OCQ180!=99999)
master.df <- filter(master.df, OCQ180!=77777)
# No NA's

########################################################
############# PAQ_J - Physical Act #####################
########################################################
master.df <- filter(master.df, PAD680!=9999)
master.df$PAD675<- master.df$PAD675 %>% replace_na(0)
master.df <- filter(master.df, PAD675!=9999)
```

```
# PAD680 - Minutes of Sed Activity per Day
# PAD675 - Minutes of Mod Activity per Day

#####################################################
############# SLQ_J - Sleep ##########################
#####################################################
master.df <- filter(master.df, SLQ050!=9)
master.df <- filter(master.df, SLQ120!=9)

master.df$SLD012<- master.df$SLD012 %>%
  replace_na(median(master.df$SLD012, na.rm = TRUE))

master.df$SLD013<- master.df$SLD013 %>%
  replace_na(median(master.df$SLD013, na.rm = TRUE))

master.df$SLQ050 <- as.factor(master.df$SLQ050)
levels(master.df$SLQ050) <- c("Yes", "No")

master.df$SLQ120 <- as.factor(master.df$SLQ120)
levels(master.df$SLQ120) <- c("Never", "Rarely", "Sometimes",
                              "Often", "Always")

#No Missing Values

write.csv(master.df, "master.df.csv")
```

Appendix C – Code for Classification Models

```
library(caret)
library(MASS)
library(caret)
library(Metrics)

# LOGISTIC MODELS

df.clean <- df.select[,-1]
df.clean <- na.omit(df.clean)
log_fit.full <- glm(ADDEPEV3~., data=df.clean, family=binomial)
log_fit.full <- update(log_fit.full, ~ . -BPHIGH4 - INCOME2 - FRUTDA2)
summary(log_fit.full)

# Check misclassification error for full model
preds.full <- predict(log_fit.full, type="response")
probs.full <- ifelse(preds.full >= 0.5, "No", "Yes")
mean(probs.full!=df.clean$ADDEPEV3) # 16.5% misclassifcation
confusionMatrix(as.factor(probs.full), reference=df.clean$ADDEPEV3)

# Remove variables that are not statistically significant
df.clean1<-df.clean[,c(-6, -11, -20),]

# Split Train and Test
set.seed(1234)
train <- sample(1:nrow(df.clean1),nrow(df.clean1)/2)
test <- -train

log_fit.train <- glm(ADDEPEV3~., data=df.clean1[train,], family=binomial)
preds.test <- predict(log_fit.train,
                      newdata = df.clean1[test,], type="response")
probs.test <- ifelse(preds.test >= 0.5, "No", "Yes")
mean(probs.test!=df.clean1$ADDEPEV3[test]) #16.5% misclassifcation
confusionMatrix(as.factor(probs.test), reference=df.clean1$ADDEPEV3[test])
f1(df.clean1$ADDEPEV3[test],as.factor(probs.test))


# Null Model - Only slightly worst, 19.5% misclassification
# Imbalanced dataset - responses
preds.null <- rep("No", nrow(df.clean1))
mean(preds.null!=df.clean1$ADDEPEV3[test])

# Need AUC/ROC curves
library(pROC)
par(pty="s")
roc(df.clean1$ADDEPEV3[train], log_fit.train$fitted.values, plot=TRUE)

#LDA AND QDA MODELS################################
library(MASS)

lda.fit <- lda(ADDEPEV3~., data=df.clean1[train,])
lda.fit
plot(lda.fit, type="b")

# Accuracy of LDA Fit - Test Set
lda.pred <- predict(lda.fit, newdata=df.clean1[test,])
lda.preds <- lda.pred$class
confusionMatrix(lda.preds, reference = df.clean1$ADDEPEV3[test])
# Slight increase in sensitivity rate - true positives
# No problem in finding folks without depression diagnosis

lda.fit$means
lda.fit$means[,2] #General Health
lda.fit$means[,3] #Mental Health Poor
lda.fit$means[,6] #Marital Status
lda.fit$means[,26] # Hours of Vigorous Exercise per Week
lda.fit$means[,27] # Veggies (dietary)

# Relieve constant covariance constraint.
# Would a a more flexible classifier improve results?
qda.fit <- qda(ADDEPEV3~.,, data=df.clean1[train,])
qda.fit

# Training Error
qda.pred.train <- predict(qda.fit)
confusionMatrix(as.factor(qda.pred.train$class),
                reference = df.clean1$ADDEPEV3[train])

# Mean of each variable within each class
# Disparate values indicate a discriminating variable
qda.fit$means[,2] #General Health
qda.fit$means[,3] #Mental Health Poor
qda.fit$means[,6] #Marital Status
qda.fit$means[,26] # Hours of Vigorous Exercise per Week
qda.fit$means[,27] # Dark green veggies per day
```

```
qda.fit$means

# Potential overfit
# Test Accuracy Drops
qda.preds <- predict(qda.fit, newdata = df.clean[test,])
confusionMatrix(as.factor(qda.preds$class),
                reference = df.clean1$ADDEPEV3[test])

############ DECISION TREES ########################

library(tree)
tree.train <- tree(ADDEPEV3 ~.-MENTHLTH, data=df.clean1[train,])
plot(tree.train)
text(tree.train, pretty=0)

tree.train.preds <- predict(tree.train,
                            newdata = df.clean1[-train,],
                            type="class")
mean(tree.train.preds!=df.clean1$ADDEPEV3[-train])
confusionMatrix(tree.train.preds, reference=df.clean1$ADDEPEV3[-train])

# Zero sensitivity - can not learn pattern of minority class
df.minor <- filter(df.clean1, ADDEPEV3=="Yes") # depressed class
df.major <- filter(df.clean1, ADDEPEV3=="No") # normal class

set.seed(1234)
sub <- sample(1:nrow(df.major), size=nrow(df.minor))
df.major.sub <- df.major[sub,]
df.balanced <- rbind(df.major.sub, df.minor)

# Fit tree to more balanced dataset
tree.train2 <- tree(ADDEPEV3 ~.-MENTHLTH, data=df.balanced[train,])
plot(tree.train2)
text(tree.train2, pretty=0)

tree.train2.preds <- predict(tree.train2,
                            newdata = df.balanced[-train,],
                            type="class")
mean(tree.train2.preds!=df.balanced$ADDEPEV3[-train])
confusionMatrix(tree.train2.preds, reference=df.balanced$ADDEPEV3[-train])
# Improved specificity.


## perform CV on training data to determine ideal size
set.seed(7)
cv.tree2 <- cv.tree(tree.train2, FUN= prune.misclass)
names(cv.tree2)
cv.tree2  # No need to prune

# BAGGED TREES################################

library(randomForest)
set.seed(1234)
train<-sample(x=1:nrow(df.balanced), nrow(df.balanced)*0.5)
test<--train

bag.tree <- randomForest(ADDEPEV3~.-MENTHLTH, data=df.balanced,
                            subset=train, mtry=17, importance=TRUE)
# takes 2-3min
bag.tree
summary(bag.tree)
plot(bag.tree)

# how does the bagged model perform on test data?
bag.tree.preds <- predict (bag.tree, newdata = df.balanced[-train,])
mean(bag.tree.preds!=df.balanced$ADDEPEV3[-train]) # 34.8% misclass error
confusionMatrix(bag.tree.preds, reference=df.balanced$ADDEPEV3[-train])

og.preds <- predict(bag.tree, newdata = df.clean1)
confusionMatrix(og.preds, reference=df.clean1$ADDEPEV3)
# applied tree to the same original imbalanced data set
# achieved 82% specificity.

importance(bag.tree)
varImpPlot(bag.tree)

#RANDOM FOREST ############################

rf.tree <- randomForest(ADDEPEV3~.-MENTHLTH, data=df.balanced,
                            subset=train, mtry=4, importance=TRUE)
# square root of number of predictors

# takes 3-5min
rf.tree
summary(bag.tree)
plot(bag.tree)
```

```
# how does the bagged model perform on test data?
rf.tree.preds <- predict (rf.tree, newdata = df.balanced[-train,])
mean(rf.tree.preds!=df.balanced$ADDEPEV3[-train]) # 34.8% misclass error
confusionMatrix(rf.tree.preds, reference=df.balanced$ADDEPEV3[-train])

og.preds <- predict(rf.tree, newdata = df.clean1)
confusionMatrix(og.preds, reference=df.clean1$ADDEPEV3)
# applied tree to the same original imbalanced data set
# lower specificity

importance(rf.tree)
varImpPlot(rf.tree)

# BOOSTED TRESS ############################
library(gbm)
# Build boosted model on balanced dataset
boost.tree <- gbm(ADDEPEV3~.-MENTHLTH, data = df.balanced[train, ],
                  distribution = "gaussian",
                  n.trees = 5000, interaction.depth = 1)

summary(boost.tree)

# partial dependence plots
plot(boost.tree, i = "BMI5")
plot(boost.tree, i = "EMPLOY1")
plot(boost.tree)

# Use boosted model to predict test error
pred.boost <- predict.gbm(boost.tree, newdata = df.balanced[-train, ],
                    n.trees = 5000, type="response")
pred.boost <- ifelse(pred.boost>1.5, "No", "Yes")

boost.test.error <- mean(pred.boost!=df.balanced$ADDEPEV3[-train])
confusionMatrix(as.factor(pred.boost), reference=df.balanced$ADDEPEV3[-train])
```

Appendix D – Code for Regression Models

```
####### Multiple Linear Regression #############

master.df.clean <- master.df[,c(-1,-3)]

plot(master.df.clean[,1:10], col="blue")
plot(master.df.clean[,c(1,11:20)], col="blue")
hist((master.df.clean$DepressionScore))

# Fit with all variables
lm.fit.std <- lm(DepressionScore~., data=master.df.clean)
lm.fit.std.preds <- predict(lm.fit.std, newdata = master.df.clean)
summary(lm.fit.std)

# MSE Error for entire dataset.  No train/test split
lm.fit.MSE.full <-
  mean((lm.fit.std.preds-master.df.clean$DepressionScore)^2)
lm.fit.MSE.full

lm.fit.MSE.null <-
  mean((mean(master.df.clean$DepressionScore)-
          master.df.clean$DepressionScore)^2)
lm.fit.MSE.null

# Plot of Actual vs. Predicted
# High Scatter,  but general relationship
plot(lm.fit.std.preds~master.df.clean$DepressionScore,
     xlab="Actual Depression Score", ylab="Predicted Depression Score")
abline(0,1, col="red")

# p-values and regression coefficients are only valid if assumptions are met...
plot(lm.fit.std)

library(car)
vif(lm.fit.std)

lm.coef <- coef(lm.fit.std)
write.csv(lm.coef, "lm.coef.csv")

############ SUBSETTING ####################

# Full Model, best subset selection
library(leaps) # regsubsets function, library

# best subset selection
regfit.full.best <- regsubsets(DepressionScore~.,
                               data=master.df.clean,
                              nvmax= 30)

sum.reg.best <- summary(regfit.full.best) # can get information

# Model Selection
################### ADJ R2 ##################
which.max(sum.reg.best$adjr2)
sum.reg.best$adjr2
plot(sum.reg.best$adjr2, xlab='No. of Variables',
     ylab='Adjusted R2',type='l')
points(18,sum.reg.best$adjr2[18],pch=19,col='red')
#18 Vars is Best
reg.fit.vars.R2 <- coef(regfit.full.best,18)
write.csv(reg.fit.vars.R2, "reg.fit.vars.R2.csv")
# These are the 18 Best

############## Mallows Cp ####################
which.min(sum.reg.best$cp) # 22 variables
sum.reg.best$cp
plot(sum.reg.best$cp, xlab='No. of Variables',
     ylab='Mallows Cp',type='l')
points(22,sum.reg.best$cp[22],pch=19,col='red')
reg.fit.vars.Cp <- coef(regfit.full.best,22)
write.csv(reg.fit.vars.Cp, "reg.fit.vars.Cp.csv")

############## BIC ############################
which.min(sum.reg.best$bic) # 15 variables
sum.reg.best$bic
plot(sum.reg.best$bic, xlab='No. of Variables',
     ylab='BIC',type='l')
points(15,sum.reg.best$bic[15],pch=19,col='red')
reg.fit.vars.BIC <- coef(regfit.full.best,15)
write.csv(reg.fit.vars.BIC, "reg.fit.vars.bic.csv")

############## Re-Fit Linear Model w/Ideal Subset ########
```

```
subset.vars <- c("DepressionScore", "RIDAGEYR" ,"RIDRETH1", "DMDHRMAZ" , "INDFMIN2", "BMXBMI", "BPQ080" , "ALQ130" ,
"HSD010", "DUQ200", "INDFMMPI" , "OCQ180", "SLD013" , "SLQ050" , "SLQ120" )
master.subset <- master.df.clean[,subset.vars]

subset.lm.fit <- lm(DepressionScore~., data=master.subset)
summary(subset.lm.fit)

MSE.preds <- predict(subset.lm.fit, newdata = master.subset)
mean((MSE.preds-master.subset$DepressionScore)^2)

plot(MSE.preds~master.subset$DepressionScore)
abline(0,1, col="red")

subset.coef <- as.data.frame(as.matrix(coef(subset.lm.fit)))
write.csv(subset.coef, "subset.coef.csv")

vif(subset.lm.fit) # eliminated multi-collinearity


########## POWER TRANSFORM Y ########
lm.fit.root <- lm(DepressionScore^(1/2)~., data=master.subset)
summary(lm.fit.root)
plot(lm.fit.root)

MSE.preds.root <- predict(lm.fit.root, newdata = master.subset)
plot(MSE.preds.root~(sqrt(master.subset$DepressionScore)))
abline(0,1, col="red") # more normal distribution of residuals

mean((MSE.preds.root^2-master.subset$DepressionScore)^2)

############## RIDGE and LASSO REGRESSION ##########

library(glmnet) # load library
# Build model on training set (50/50)
# Optimize model via CV

set.seed(1)
grid <- 10^seq(10, -2, length = 100) # grid of lambda values
train <- sample(x=1:nrow(master.df.clean), size= nrow(master.df.clean)/2)
test <- (-train) # negated index
y <-master.df.clean$DepressionScore
y.test <- y[test] # subset the entire y vector for test data
y.train <- y[train] # subset the entire y vector for training data
x <- model.matrix(DepressionScore~., data=master.df.clean)[,-1]
x.train <- x[train,]

ridge.fit <- glmnet(x=x.train, y=y.train,
                    alpha=0, lambda = grid, thresh = 1e-12)

plot(ridge.fit)

cv.ridge.out <- cv.glmnet(x.train, y.train, alpha = 0)
plot(cv.ridge.out)
bestlam <- cv.ridge.out$lambda.min
bestlam # small lambda value

# Check Test MSE
ridge.fit.pred <- predict(ridge.fit, s=bestlam,
                          newx = x[-train,])

ridge.MSE <- mean((ridge.fit.pred-master.df.clean$DepressionScore[-train])^2)
ridge.MSE # not much better than standard lm

# Fit to Full Model
# Determine Coefficients
ridge.final <- glmnet(x, y, lambda=bestlam, alpha=0)
ridge.coef <- predict(ridge.final, type="coefficients")
ridge.coef <-as.data.frame(as.matrix(ridge.coef))
plot(ridge.fit, xvar = "lambda", label=TRUE)
write.csv(ridge.coef, "ridge.coef.csv")

# Calculate R2 Value of Ridge Model
y.predicted <- predict(ridge.final, s = bestlam, newx = x)
ridge.full.MSE <- mean((y.predicted-y)^2)

#find SST and SSE
ss.residuals <- sum((y - mean(y))^2)
ss.error <- sum((y.predicted - y)^2)
rsq <- 1 - ss.error/ss.residuals
rsq #0.321

# LASSO

# Fit Lasso on training Data
lasso.fit = glmnet(x.train, y.train, alpha = 1, lambda = grid)

# find optimal lambda by cross validation
```

```
set.seed(1)
cv.lasso = cv.glmnet(x.train, y.train, alpha = 1) # Fit lasso model on training data
plot(cv.lasso) # Plot training MSE as a function of lambda
bestlam.lasso <- cv.lasso$lambda.min # Select best lambda
lasso.pred <- predict(lasso.fit, s = bestlam.lasso, newx = x[-train,]) # Calculate test MSE
mean((lasso.pred - y.test)^2) # Test MSE, not much improvement

# Fit Full Lasso Model
lasso.fullfit <- glmnet(x, y, alpha = 1, lambda = bestlam.lasso)
lasso.full.coef <- predict(lasso.fullfit, type = "coefficients", s = bestlam)
lasso.coef <-as.data.frame(as.matrix(lasso.full.coef))
write.csv(lasso.coef, "lasso.coef.csv")

# Calculate MSE of full model
y.pred.lasso <- predict(lasso.fullfit, s = bestlam.lasso, newx = x)
lasso.full.MSE <- mean((y.pred.lasso-y)^2)
# 12.18

#find SST and SSE
ss.residuals <- sum((y - mean(y))^2)
ss.error.lasso <- sum((y.pred.lasso - y)^2)
rsq.lasso <- 1 - ss.error.lasso/ss.residuals
rsq.lasso #0.319, about the same


###### RANDOM FOREST REGRESSION ###########

library(randomForest)

set.seed(1234)
train <- sample(x=1:nrow(master.df.clean),
                size= nrow(master.df.clean)/2)
test <- (-train) # negated index
y <-master.df.clean$DepressionScore
y.test <- y[test] # subset the entire y vector for test data
y.train <- y[train] # subset the entire y vector for training data
x.train <- x[train,]
x.test <- x[test,]

RF.model.1 <- randomForest(DepressionScore~., data=master.df.clean,
                           subset=train, mtry=1, importance=TRUE)

RF.model1.pred <- predict(RF.model.1, newdata = master.df.clean[test,])
RF.model.1.mse <- mean((RF.model1.pred-master.df.clean$DepressionScore[test])^2)

RF.model.2 <- randomForest(DepressionScore~., data=master.df.clean,
                           subset=train, mtry=2, importance=TRUE)

RF.model2.pred <- predict(RF.model.2, newdata = master.df.clean[test,])
RF.model.2.mse <- mean((RF.model2.pred-master.df.clean$DepressionScore[test])^2)

RF.model.3 <- randomForest(DepressionScore~., data=master.df.clean,
                           subset=train, mtry=3, importance=TRUE)

RF.model.3.pred <- predict(RF.model.3, newdata = master.df.clean[test,])
RF.model.3.mse <- mean((RF.model.3.pred-master.df.clean$DepressionScore[test])^2)

RF.model.4 <- randomForest(DepressionScore~., data=master.df.clean,
                           subset=train, mtry=4, importance=TRUE)
RF.model.4.pred <- predict(RF.model.4, newdata = master.df.clean[test,])
RF.model.4.mse <- mean((RF.model.4.pred-master.df.clean$DepressionScore[test])^2)

RF.model.5 <- randomForest(DepressionScore~., data=master.df.clean,
                           subset=train, mtry=5, importance=TRUE)
RF.model5.pred <- predict(RF.model.5, newdata = master.df.clean[test,])
RF.model.5.mse <- mean((RF.model5.pred-master.df.clean$DepressionScore[test])^2)

RF.model.6 <- randomForest(DepressionScore~., data=master.df.clean,
                           subset=train, mtry=6, importance=TRUE)
RF.model6.pred <- predict(RF.model.6, newdata = master.df.clean[test,])
RF.model.6.mse <- mean((RF.model6.pred-master.df.clean$DepressionScore[test])^2)

RF.mse <- c(RF.model.1.mse,RF.model.2.mse,RF.model.3.mse,RF.model.4.mse,RF.model.5.mse,RF.model.6.mse)
mtry <- c(1,2,3,4,5,6)
plot(RF.mse~mtry, type="l", xlab="mtry value", ylab="Test MSE")

# Optimized RF Regression Model
rf.final <- randomForest(DepressionScore~., data=master.df.clean,
                           , mtry=4, importance=TRUE)
importance(rf.final)
varImpPlot(rf.final)

# Cutoff Values for Moderate and Severe Depression
df.clean.severe <- filter(master.df.clean, DepressionScore!=0)

set.seed(1234)
train.rf <- sample(x=1:nrow(df.clean.severe), nrow(df.clean.severe)*0.5)
```

```
test.rf <- -train.rf

rf.severe <- randomForest(DepressionScore~., data=df.clean.severe,
                          subset=train.rf, mtry=4, importance=TRUE)

RFpred <- predict(rf.severe, newdata = df.clean.severe[test,])
RFpredMSE<- mean((RFpred-df.clean.severe$DepressionScore[test])^2)

SSR=sum((RFpred-df.clean.severe$DepressionScore[test])^2)
SSE=sum((df.clean.severe$DepressionScore[test]-mean(df.clean.severe$DepressionScore[test]))^2)
RSQ<- 1-SSE/SSR

plot(df.clean.severe$DepressionScore[test]~RFpred,
     xlab="Predicted", ylab="Actual")
abline(0,1, col="red")

plot(rf.severe)
varImpPlot(rf.severe)
```

[i] Kroenke K, Spitzer RL, William JB. The PHQ-9: validity of a brief depression severity measure. J Gen Intern Med 2001; 16: 1606-13