STAT 602 - Homework 11

John Herbert

Document External Libraries

- **NHANES** package for *NHANES* dataset
- **knitr** package used for *kable* function used to format tables
- **ggplot2** package for graphs
- **GGally** package for pairs plot
- **dplyr** package for restructuring data frames
- **tidyverse** package for restructuring data frames
- **gridExtra** package for formatting multiple **ggplot2** graphs
- **splines** package for spline regression modeling
- **randomForest** package used for feature selection modeling
- **class** package for KNN models
- **MASS** package for LDA and QDA models
- **fast Dummies** package for dummyvariable conversion
- **klaR** package for LDA and QDA plots
- **nnet** package for neural network model

Reusable Functions

- The misclassification function created in homework 3 (*misclass.fun.JH*) will be reused for questions in this homework.

Exercises

Use set.seed(202111) when appropriate to make results reproducible.

Question 1 (MDSR 8.1 pg 201)

(Modified from 8.1 pg 201 in **Modern Data Science with R**.) The ability to get a good night's sleep is correlated with many positive health outcomes. The `NHANES` data set contains a binary variable `SleepTrouble` that indicates whether each person has trouble sleeping. For each of the listed models - Logistic Regression, Neural network, K - Nearest Neighbors, LDA, and QDA, repeat all of the following steps:

Task 1: Data Cleansing, Train/Test Split, Scaling, & Feature Selection

Using the Validation Set Approach with a split of 90/10, build a classifier for `SleepTrouble` on the training data. You will have to use a subset of the variables.

Answer

Step 1: Importing and Examining the Data I imported the data from the NHANES package, which was a helath statistic survey taken in 2009-2012. The propose of this study is to use the *SleepTrouble* variable as the target (either a Yes or No), and a number of features to determine if we can predict whether a particpants has sleep trouble or not.

```
## The NHANES dataset's contains 10000 rows or observations and 76 variables or columns
```

Step 2: Removing Null Target Observations and Subsetting Variables According to the documentation, the survey only asked participants if they had sleep trouble when they were 16 years old or less. Therefore, all particpants 16 and younger were dropped. This reduced the number of observations from 10,000 to 7,772.

```
## The number of rows after removing null target observations 7772
```

There are also a number of variables that have a large number of null values that will not add much insight and mostly noise into the model. Therefore, any variable that has 80% null values will be removed from the dataset. The variables removed are listed below as well as their corresponding percantage of null values.

Table 1: Null Values over 80%

Variable	Class	Perc_Null
Length	numeric	1.000
HeadCirc	numeric	1.000
BMICatUnder20yrs	factor	0.968
UrineVol2	integer	0.841
UrineFlow2	numeric	0.842
DiabetesAge	integer	0.921
TVHrsDayChild	integer	1.000
CompHrsDayChild	integer	1.000
AgeRegMarij	integer	0.824

```
## Reduction of variables with over 80% null reduces number of variables to 67
```

Upon studying the literature and examining the other variables, there were a number of variables that were very similar or correlated to other variables, repetitive, or unuseful:

- **ID**: Used to identify the particpant and not useful in my analysis
- **SurveyYr**: Used to determine the when the survey was done and unuseful for my analysis
- **AgeMonths**: Number of Age variables, and since we are looking at partipants 16 and older, the months old is unnecessary
- **Race3**: Was not used for the 2009-10 survey, **Race1** was used instead
- **Testosterone**: Was not used for the 2009-10 survey
- **SleepHrsNight**: Very similar to the target variable and not useful as a predictor
- **Height**: Unlikely to be a factor in sleep trouble (unless tall past a certain point can't find a appropiately sized bed). Also, not a proactive variable in how to improve sleep trouble.
- **BMI_WHO**: Opted to use the *BMI* variable instead of this categorical variable giving the same data
- **HHIncomeMid**: Opted to use **HHIncome** as a predictor instead
- **Poverty**: Opted to use **HHIncome** as a predictor instead
- **TVHrsDay**: Was not used for the 2009-10 survey
- **CompHrsDay**: Was not used for the 2009-10 survey
- **Smoke100n**: No documentation on variable, however appears to be the same as **Smoke100**

- **BPSys1,BPSys2,BPSys3**: Used **BPSysAve** instead
- **BPDia1,BPDia2,BPDia3**: Used **BPDiaAve** instead
- **SexAge**: There are a number of other sex related variables that were used instead and more focused on current sexual activity
- **SexNumPartnLife**: There are a number of other sex related variables that were used instead and more focused on current sexual activity
- **Age1stBaby**: There are a number of other pregnancy related variable that were used and more focused on current pregnancy activity
- **nPregnancies**: There are a number of other pregnency related variable that were used and more focused on current pregnancy activity
- **SmokeAge**: There are a number of other smoking related variable that were used and more focused on current smoking activity
- **AgeFirstMarij**: There are a number of other marijuana related variable that were used and more focused on current marijuana activity
- **DirectChol**: Very similar to **TotChol**, which was used instead
- **Weight**: Generally not a good metric to use a health statistic as it will depend on a number of other factors. **BMI** was used instead.

```
## The number of columns after removing repetitive/unuseful variables is 40
```

Seperating variables between binary, categorical, and continuous. Also converting heirarchical categorical variables into numeric values, as there is importance in the order of the factor. For example, *AgeDecade* is a categorical variable but has meaning in the order of each category. In addition, I removed the target variable from these datasets to make missing value and scaling easier.

Converting the heirarchial variables mentioned in the step above to numeric and binding it with the other numeric variables.

```
## Dimensions of numeric variables is 7772 22 and the dimension of categorical variables is 7772 17
```

Step 4: Handling Missing Categorical Variables Replace missing values for categorical variables. Showing the total missing values of the categorical features in the dataset.

Table 2: Total Null Values of Categorical Variables

	Variable	Null
17	PregnantNow	6076
9	SmokeNow	4561
16	SexOrientation	2930
11	Marijuana	2831
12	RegularMarij	2831
13	HardDrugs	2007
14	SexEver	2005
15	SameSex	2004
8	Alcohol12PlusYr	1192
3	MaritalStatus	541
10	Smoke100	537
4	HomeOwn	58
6	Diabetes	2
5	Work	1
1	Gender	0
2	Race1	0
7	PhysActive	0

Each categorical missing value was replaced with the following for each feature in the dataset:

- **PregnantNow**: If observation is male, then not applicable, if observation is older than 59 then not applicable, else missing
- **SexOrientation**: If observation is missing and age is not 18-59, then not applicable, else missing
- **Marijuana**: If observation is missing and not age 18-59 then not applicable, else missing
- **RegularMarij**: If observation is missing and not age 18-59 then not applicable, else missing
- **HardDrugs**: If observation is missing and not age 18-59 then not applicable, else missing
- **SexEver**: If observation is missing and not age 18-59 then not applicable, else missing
- **SameSex**: If observation is missing and not age 18-59 then not applicable, else missing
- **Alcohol12PlusYr**: If observation is missing and age is younger than 18 then not applicable, else missing. Since it was not asked of partipants younger than 18, the survey assumes they do not drink 12 or more drinks in one year, which may or may not be true.
- **Smoke100**: If observation is missing and age is younger than 20 then not applicable, else missing. Since it was not asked of partipants younger than 20, the survey assumes they have not smoked 100 cigarettes or more in their life, which may or may not be true.
- **SmokeNow**: If observation is missing and age is younger than 20 then not applicable, else missing
- **HomeOwn, Diabetes, Work**: Asked of all particapants, therefore all missing values are truly missing

Runinng the missing value count again to confirm that all missing values have been replaced in the categorical set.

Table 3: Total Null Values of Categorical Variables

Variable	Null
Gender	0
Race1	0
MaritalStatus	0
HomeOwn	0
Work	0
Diabetes	0
PhysActive	0
Alcohol12PlusYr	0
SmokeNow	0
Smoke100	0
Marijuana	0
RegularMarij	0
HardDrugs	0
SexEver	0
SameSex	0
SexOrientation	0
PregnantNow	0

Step 5: Splitting Data into Train/Test Sets Before I can replace missing values for numeric features, I will have to split the data in the training and test sets. Since I will be replacing some of the missing values with the means, I cannot let my training set influence my test set. To split the data, I used the *sample.int* function with a 90% training and 10% test split. This was done for the target, categorical, and numeric variables. In addition, there is an inbalance between the categories of the target (Yes and No). Therefore, each class was seperated then randomly split between train and test, then binded back together.

Finally, while the **Age** variable was not used as a predictor, it was used as a filter to seperate the categories. Therfore, it was removed from the numeric train/test sets but maintained as a seperate variable.

```
## Total observations in the training set is 6994 and the total for the test set is 778
```

Step 6: Replacing Missing Values for Continuous Variables in Training Set The process for replacing missing numeric variable is similar to the categorical one in that each feature was calculated inidvidually depending on the specification of the survey. The total number of missing values per feature is:

Table 4: Total Null Values of Continuous Variables

	Variable	Null
11	nBabies	4825
12	PhysActiveDays	3493
13	AlcoholDay	2582
15	SexNumPartYear	2565
14	AlcoholYear	1666
20	LittleInterest	990
21	Depressed	986
9	DaysPhysHlthBad	713
10	DaysMentHlthBad	710
19	HealthGen	706
18	HHIncome	591
17	Education	498
8	UrineFlow1	476
6	TotChol	387
16	AgeDecade	305
4	BPSysAve	265
5	BPDiaAve	265
3	Pulse	255
7	UrineVol1	94
2	BMI	63
1	HomeRooms	60

Each categorical missing value was replaced with the following for each feature in the dataset. For all mean replacements, I used the mean for the age decade as to estimate the actual value better than using the mean of the entire population. I did not use the Age mean as there may be a low number of ages and this would have too much noise for a generalization.

- **AgeDecade**: Missing values are only age 80, so those are clasified to the group '70+'
- **nBabies**: Female particpants aged 20 years or older. If male or younger than 20, then 0, else mean of AgeDecade
- **PhysActiveDays**: Particpants 12 years or older, however in the dataset the youngest age is 16 so irrelevant, therefore, the mean of the physical activity days for the AgeDecade was used for all missing values
- **AlcoholDay**: Participants 18 years and younger assumes that drinks consumed is zero. All participants 18 years and older with missing values are average of AgeDecade
- **SexNumPartYear**: Participants not 18-59 assumed to have no sex partners, all other missing values replaced with average of AgeDecade
- **AlcoholYear**: Participants younger than 18 assumes zero drinks. All other missing values uses mean of AgeDecade
- **LittleInterest**: Participants younger than 18 assumes no days of little interest in doing things, all other missing values uses mean of AgeDecade
- **Depressed**: Participants younger than 18 assumes no depression, all other missing values uses mean of AgeDecade

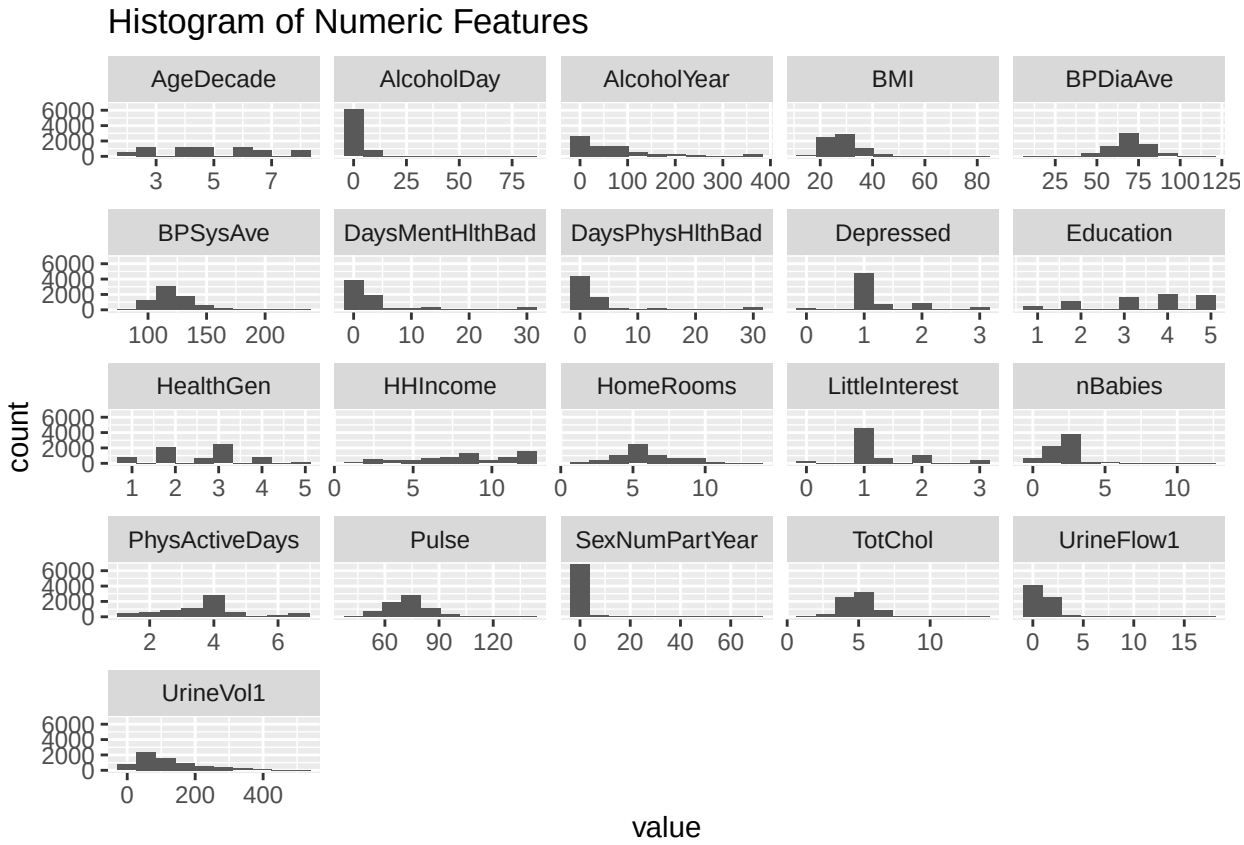
- **DaysPhysHlthBad**: Reported for participants 12 and older, however the youngest in the set is 16, so all missing values takes average of AgeDecade
- **DaysMentHlthBad**: Reported for participants 12 and older, however the youngest in the set is 16, so all missing values takes average of AgeDecade
- **HealthGen**: Reported for participants 12 and older, however the youngest in the set is 16, so all missing values takes average of AgeDecade
- **HHIncome**: No age restriction, so all missing values takes average of AgeDecade
- **Education**: Participants 16-17 assumes 9-11thGrade and 18-19 assumes HighSchool, all other missing values uses mean of AgeDecade
- **UrineFlow1**: Asked of participants 6 and older, therefore, all missing values takes average of AgeDecade
- **TotChol**: Asked of participants 6 and older, therefore, all missing values takes average of AgeDecade
- **BPSysAve, BPDiaAve**: Taken for all participants, therefore, all missing values takes average of AgeDecade
- **UrineVol1**: Taken for participants 6 and older, therefore, all missing values takes average of AgeDecade
- **Pulse**: Taken for all participants, therefore, all missing values takes average of AgeDecade
- **BMI**: Taken for all participants over 2 years old, therefore, all missing values takes average of AgeDecade
- **HomeRooms**: Asked of all participants, therefore, all missing values takes average of AgeDecade

Reran missing value check to confirm no null values are in the numeric data set.

Table 5: Total Null Values of Continuous Variables

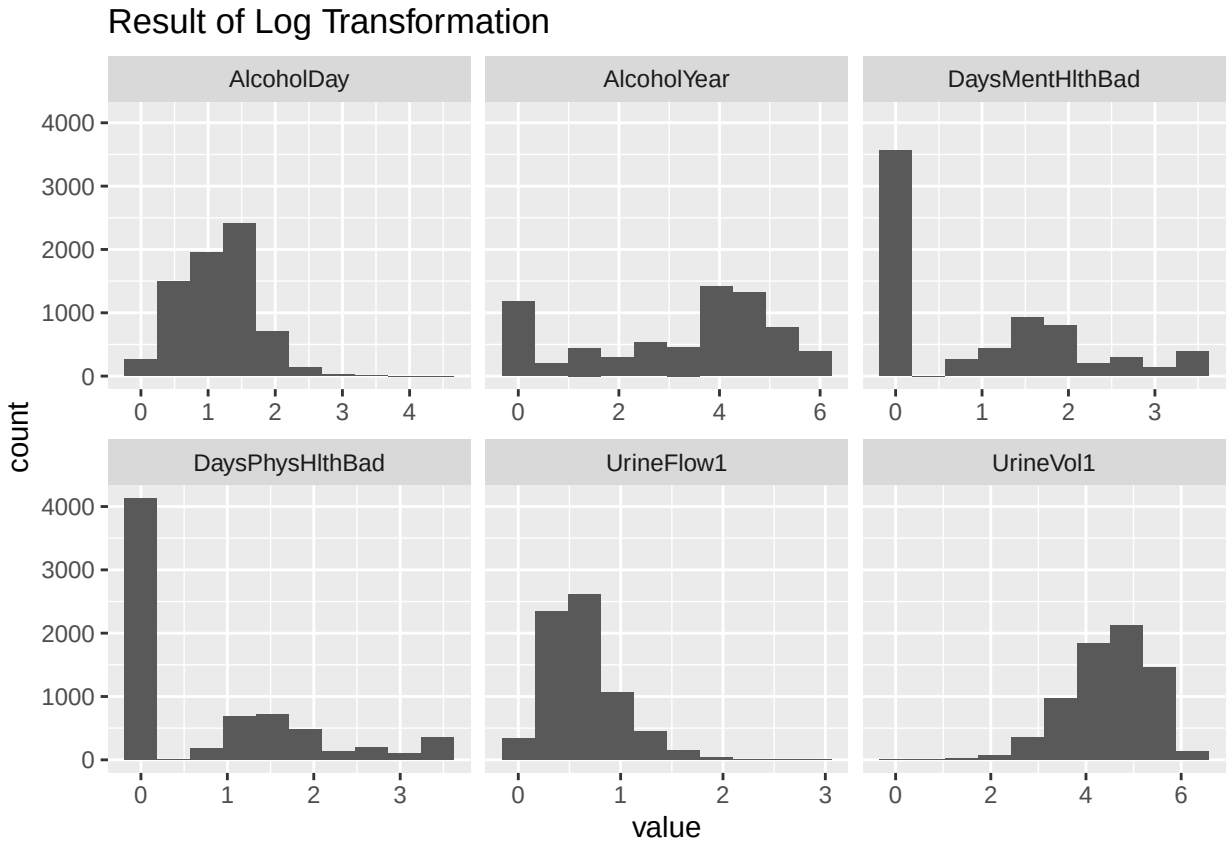
	Variable	Null
11	nBabies	0
12	PhysActiveDays	0
13	AlcoholDay	0
15	SexNumPartYear	0
14	AlcoholYear	0
20	LittleInterest	0
21	Depressed	0
9	DaysPhysHlthBad	0
10	DaysMentHlthBad	0
19	HealthGen	0
18	HHIncome	0
17	Education	0
8	UrineFlow1	0
6	TotChol	0
16	AgeDecade	0
4	BPSysAve	0
5	BPDiaAve	0
3	Pulse	0
7	UrineVol1	0
2	BMI	0
1	HomeRooms	0

Step 6: Log Transformation of Numeric Variables I created a histogram of the numeric variables to test for skewness in the data and determine if any variables need log transformation to normalize them.



According to the histograms, *AlcoholDay*, *AlcoholYear*, *DaysMentHlthBad*, *DaysPhysHlthBad*, *UrineFlow1*, and *UrineVol1* all have right skewness and may benefit from log transformation by normalizing the data.

Therefore, I will take the log transformation of these variables and rerun the histrogram to see if there are improvements.

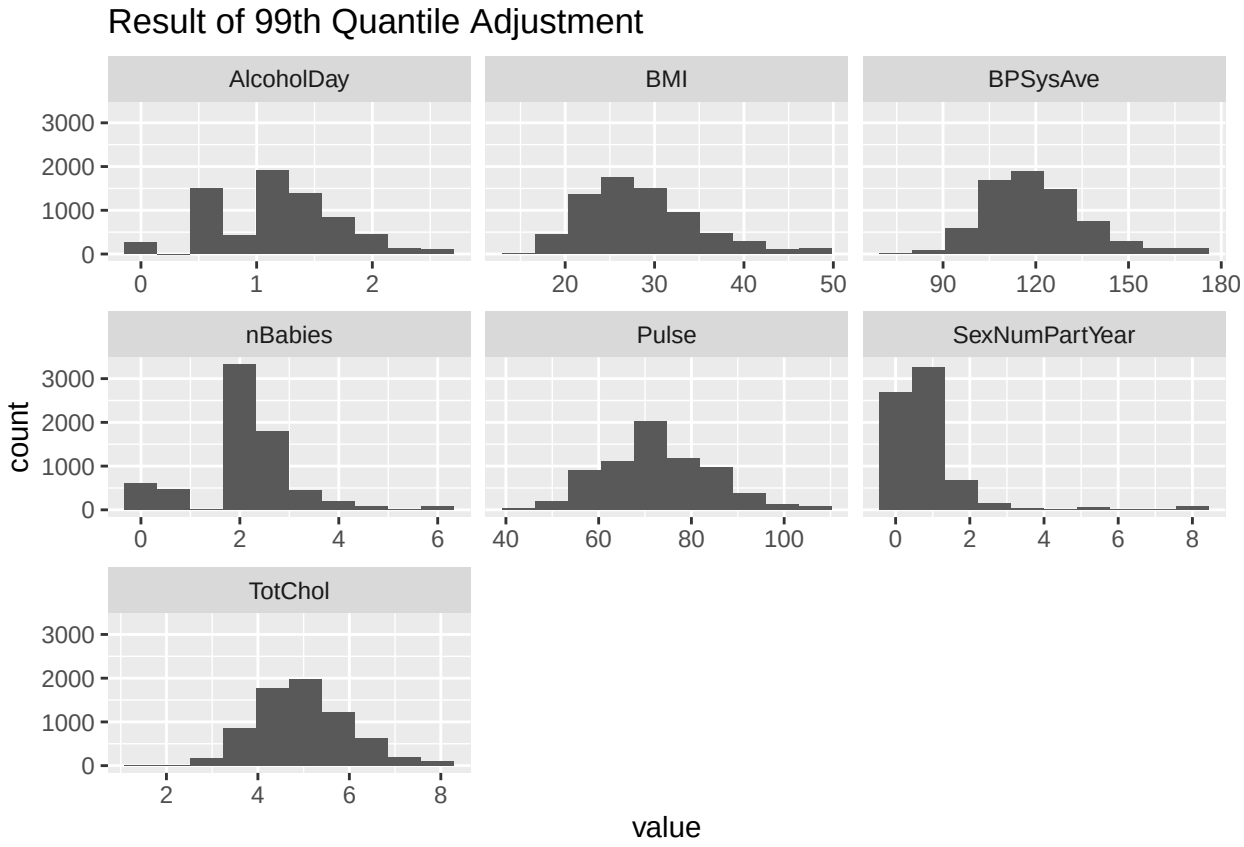


According to the histogram, most of the variables improved from the transformation, but not all are normalized still. For example, **DaysPhysHlthBad** and **DaysMentHlthBad** have a dispropriately large number of zero values. However, this will be normalized during the scaling process.

Step 7: Adjusting Extreme Values in Numeric Variables Examining variables in the histograms above, there are features that have large, extreme values that are skewing the mean and median. I will adjust these features by changing all values over the 99th quantile to the 99th quantile reduce the skew.

Table 6: Summary Statistics of Large Skewed Variables

AlcoholDay	BMI	BPSysAve	SexNumPartYear	TotChol	nBabies	Pulse
Min. :0.0000	Min. :15.02	Min. : 78.0	Min. : 0.000	Min. : 1.530	Min. : 0.000	Min. : 40.00
1st Qu.:0.6931	1st Qu.:23.80	1st Qu.:109.0	1st Qu.: 0.000	1st Qu.: 4.270	1st Qu.: 1.767	1st Qu.: 64.00
Median :1.1017	Median :27.50	Median :118.0	Median : 1.000	Median : 4.945	Median : 2.161	Median : 72.00
Mean :1.1956	Mean :28.51	Mean :120.4	Mean : 0.965	Mean : 5.003	Mean : 2.156	Mean : 72.62
3rd Qu.:1.3960	3rd Qu.:31.90	3rd Qu.:129.0	3rd Qu.: 1.000	3rd Qu.: 5.610	3rd Qu.: 2.641	3rd Qu.: 80.00
Max. :4.4188	Max. :81.25	Max. :226.0	Max. :69.000	Max. :13.650	Max. :12.000	Max. :136.00



Step 8: Replace Numeric Variables in the Test Set Using the same method as with the training set, however I will replace the missing values with the mean of the training set as to not corrupt the test set in the validation process.

Confirming there are no null values in the test set.

Table 7: Total Null Values of Continuous Test Variables

	Variable	Null
11	nBabies	0
12	PhysActiveDays	0
13	AlcoholDay	0
15	SexNumPartYear	0
14	AlcoholYear	0
20	LittleInterest	0
21	Depressed	0
9	DaysPhysHlthBad	0
10	DaysMentHlthBad	0
19	HealthGen	0
18	HHIncome	0
17	Education	0
8	UrineFlow1	0
6	TotChol	0
16	AgeDecade	0
4	BPSysAve	0
5	BPDiaAve	0

	Variable	Null
3	Pulse	0
7	UrineVol1	0
2	BMI	0
1	HomeRooms	0

Step 9: Replace Categorical Variables with Dummy Variables Since some of the model I will be fitting require only numeric values, I am converting all the categorical features into dummy variables using the *dummy_cols* function.

Since there are some variables that are only present in the training set and not the test, I will remove these from the test set. These are the *Work_missing* and *Diabetes_No* features. Also since *Work_missing* is removed, there is no reason for *Work_No* as if I left it in I would fall into the dummy variable trap.

Step 10: Numeric Feature Scaling Scale each numeric variable for feature selection and model building using Robust Scaling method. This method is used because many of the features are not normal and I want to incorporate the extreme values. The formula for the robust scaling is below or the value of each feature minu the median of the feature divideed by the difference between the features 75th and 25th quantiles.

$$\frac{X_i - \eta_i}{Q_{75i} - Q_{25i}}$$

In addition, since I do not want to corrupt my test set, I will use the training median and quantiles when scaling the test data.

Step 11: Merge Categorical and Numeric Sets for Training and Test Since all the data conversions and adjustments are done (for the most part), I will bind the numeric and categorical variables together into the X train and X test sets.

```
## The Dimensions of the train set is now 6994 70 and the dimensions of the test set is now 778 70
```

Step 12: Feature Selection: Remove Reduntant Features At the beginning of the analysis, I did some manual variable removels, however there are still a large number of variables in the data set that should be removed in order to reduce the time it takes to run the models and drop any repetitive values that will not add much new information into the fits.

I ran a correlation matrix on all the features and set a cutoff of 75% correlation rate to remove the highly correlated variables. The list of features are below and will be removed from the model.

Table 8: Correlated Variables over 75%

x
HardDrugs_No
SameSex_No
SexOrientation_notapp
SexOrientation_Heterosexual
PregnantNow_No
MaritalStatus_Separated
Smoke100_Yes
Work_Working
UrineVol1

```
## The Dimensions of the train set is now 6994 61 and the dimensions of the test set is now 778 61
```

Step 13: Feature Selection: Random Forest Accuracy-Based Importance Next, I will use the accuracy-based importance measure in the random forest to determine which variables are most importanct to the model. This function examine each variable and randomly determines the difference in the mean scores when it is in the model and then removed. The variables with the largest difference gets a higher score.

	Variable	Importance
42	BMI	170.7206
47	UrineFlow1	163.0910
46	TotChol	153.0463
44	BPSysAve	144.1026
45	BPDiaAve	142.9742
43	Pulse	135.1808
49	DaysMentHlthBad	118.6909
48	DaysPhysHlthBad	108.0388
53	AlcoholYear	106.1661
57	HHIncome	106.0302
41	HomeRooms	95.3618
50	nBabies	81.6472
52	AlcoholDay	80.0369
51	PhysActiveDays	75.0897
58	HealthGen	68.9961
55	AgeDecade	65.8049
56	Education	60.6118
60	Depressed	58.1699
59	LittleInterest	51.8895
54	SexNumPartYear	37.8286

According to the Random Forest model, the top 3 important variables are **BMI**, **UrineFlow1**, and **TotChol**. This makes logical sense as people that are overweight will generally have trouble sleeping, people who have to use the bathroom frequently will be woken up in the middle of the night to use the restroom, and those will the high cholesterol are generally unhealthly.

It seems that there still maybe some relationship between blood pressure, total cholesterol, pulse and body mass index, so I will run interactions between these variables to see if the p values improve more than the variables themselves.

After running many tests, BPSys and BPDia were the only ones that the interaction had a better p-value that the variables individually. Therefore, I added the interaction and dropped each feature from the training and test sets.

Table 10: Interaction Analysis of BPSys & BPDia

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.1063	0.0293	-37.7297	0.0000
BPSysAve	0.0659	0.0361	1.8241	0.0681
BPDiaAve	-0.0175	0.0376	-0.4651	0.6419
BPSysAve:BPDiaAve	0.0742	0.0322	2.3083	0.0210

```
## The Dimensions of the train set is now 6994 59 and the dimensions of the test set is now 778 59
```

I will rerun the random forest importance test to determine the final set of variables I will use as predictors in the models.

	Variable	Importance
42	BMI	180.2432
45	UrineFlow1	173.5652
59	BP	168.2217
44	TotChol	163.6836
43	Pulse	141.3446
47	DaysMentHlthBad	122.2470
51	AlcoholYear	110.0251
55	HHIncome	109.2235
46	DaysPhysHlthBad	108.6775
41	HomeRooms	101.6210
48	nBabies	85.7625
50	AlcoholDay	83.7794
49	PhysActiveDays	80.7276
56	HealthGen	74.4984
53	AgeDecade	66.7429
54	Education	64.0131
58	Depressed	59.3433
57	LittleInterest	54.4500
52	SexNumPartYear	41.1797
1	Gender_male	33.2486

The largest drop in importance score is between **UrineVol1** and **Pulse**, however this only gives me 5 predictors, therefore I chose the third largest drop which is between **LittleInterest** and **SexNumPartYear** for the features in my models. This will give 19 predictor variables and captures are large amount of the importance.

```
## The Dimensions of the final train set is now 6994 19 and the dimensions of the final test set is now
```

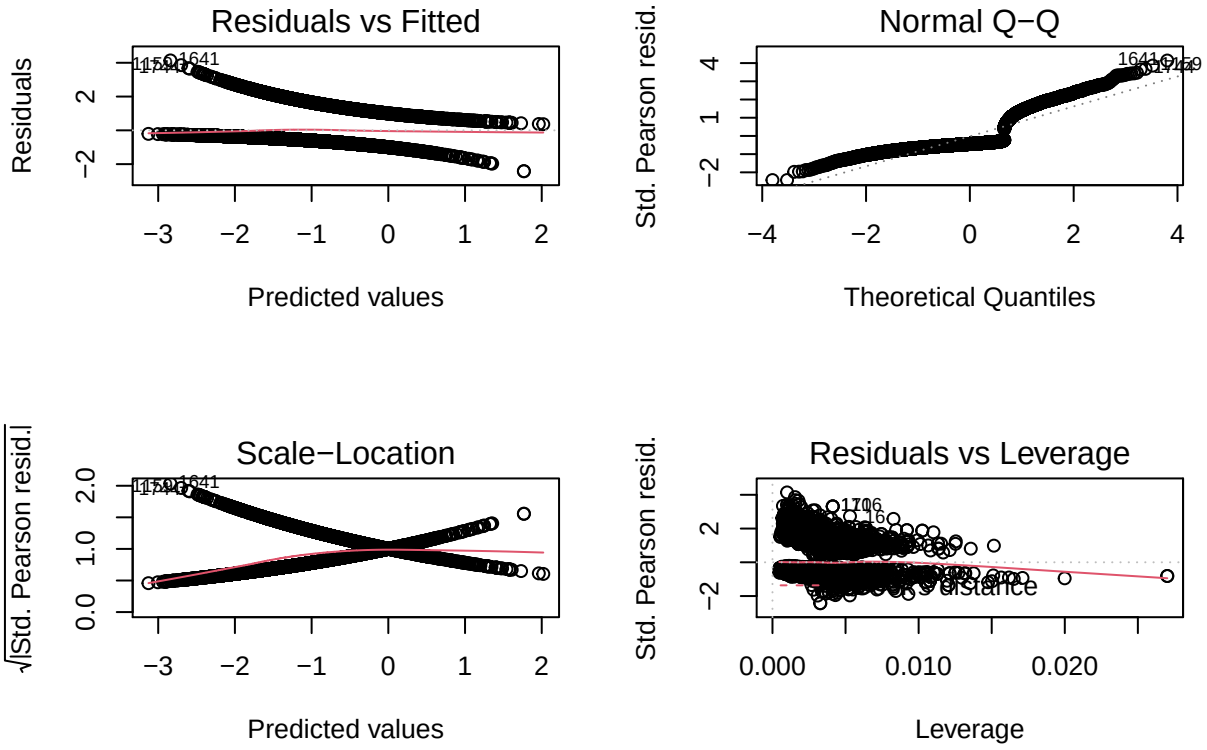
Task 2: Model Building, Testing, & Visualization

Model 1: Logistic Regression As a base model to see if any other models will show improvement, I conducted a *glm* fuction for a logistic regression. Many of the variable chosen by the Random Forest have p-values under 0.05 except for *TotChol*, *UrineVol1*, *AlcoholYear*, and *AlcoholDay*.

Table 12: Coeifficents for Logistic Regression Model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.6300	0.0563	-28.9350	0.0000
BMI	0.0807	0.0377	2.1399	0.0324
UrineFlow1	0.1058	0.0352	3.0046	0.0027
BP	0.0939	0.0312	3.0114	0.0026
TotChol	-0.0516	0.0395	-1.3076	0.1910
Pulse	0.1172	0.0417	2.8090	0.0050
DaysMentHlthBad	0.4601	0.0531	8.6604	0.0000
AlcoholYear	-0.0846	0.0481	-1.7589	0.0786
HHIncome	-0.1695	0.0547	-3.0997	0.0019
DaysPhysHlthBad	0.2872	0.0390	7.3605	0.0000

	Estimate	Std. Error	z value	Pr(>|z|)
HomeRooms	0.1014	0.0429	2.3631	0.0181
nBabies	-0.0985	0.0333	-2.9614	0.0031
AlcoholDay	-0.0540	0.0518	-1.0426	0.2972
PhysActiveDays	0.0428	0.0204	2.0993	0.0358
HealthGen	0.1803	0.0376	4.7964	0.0000
AgeDecade	0.4019	0.0484	8.3040	0.0000
Education	0.2799	0.0552	5.0707	0.0000
Depressed	0.0637	0.0171	3.7346	0.0002
LittleInterest	0.0448	0.0183	2.4425	0.0146
SexNumPartYear	-0.0004	0.0299	-0.0137	0.9891



In addition, the coefficient estimate are in the direction I would except, for example as BMI goes up, the probabiliy of sleep trouble increases. Also, as urine flow goes up, the probability of sleep trouble goes up. A unexpected results is as Total cholestrol goes down sleep toruble goes up which is counterintuitive. Also, since the units are scaled, I am un able to see if it is a large or small effect on the prediction being made.

Also, according to the graphs, the residuals are not random as there is a clear seperation between each class of the target. Also the Normal Q-Q plot shows a large curve near the median which suggests the model is not linear.

To determine the best fit for the logistic regression, I will run the predictions against the test data. In addition, I plotted the misclassification rate, TPR, and NPR to determine the optimal threshold.

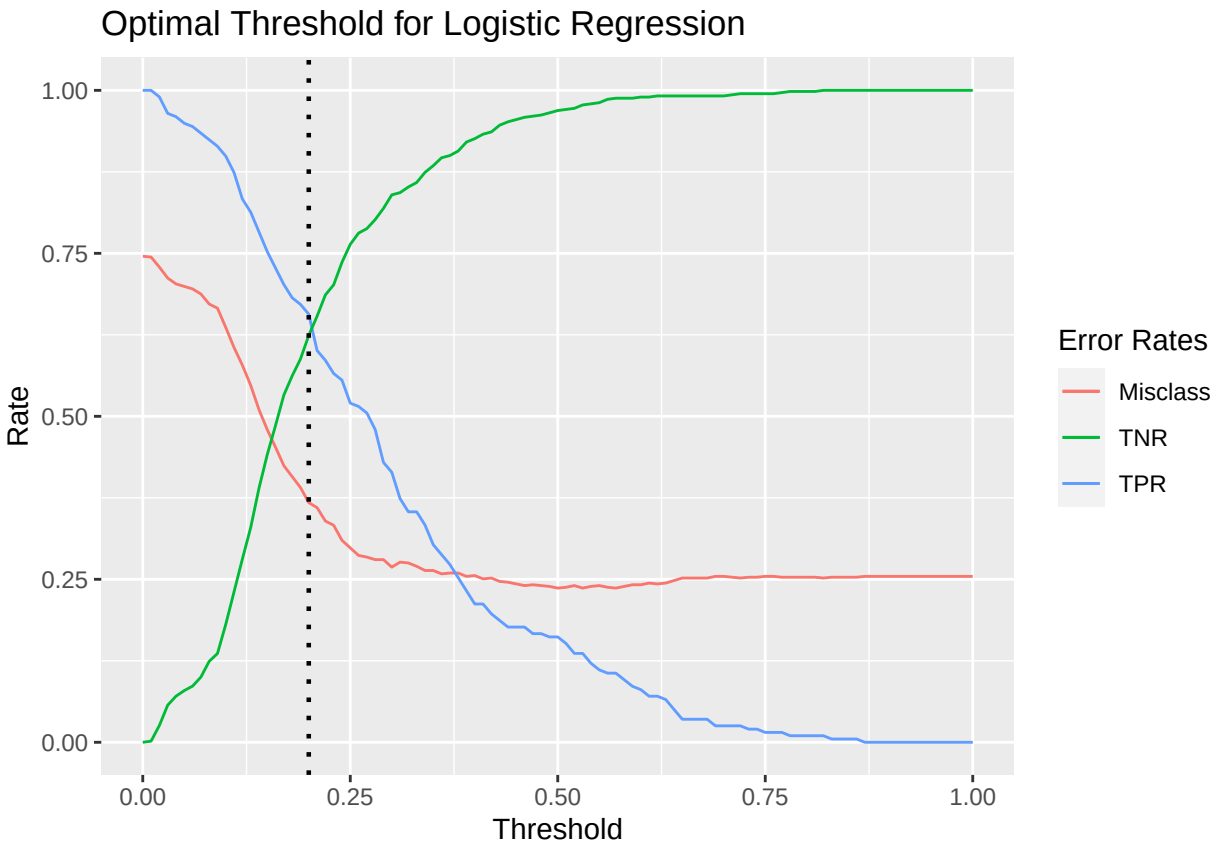
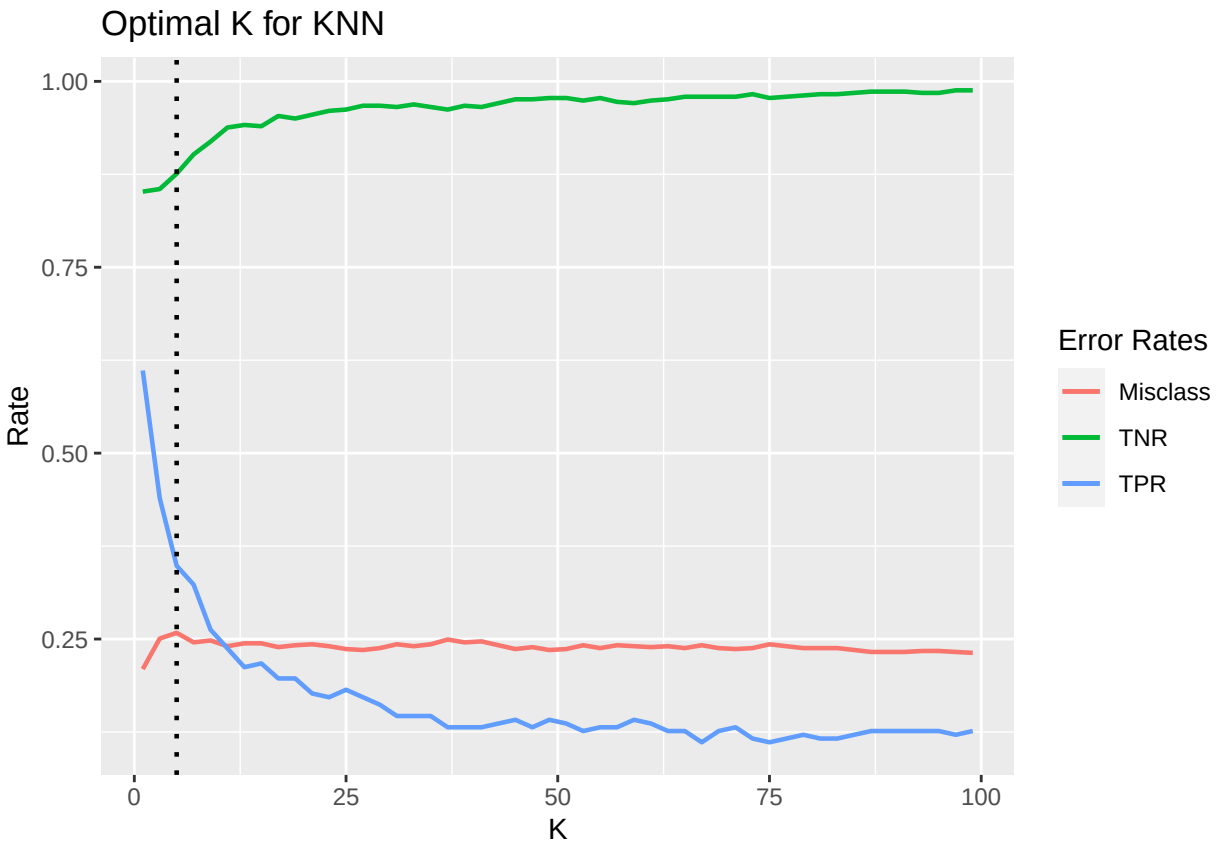


Table 13: Error Rates Threshold=0.2

Metric	Value
Misclass	0.3676
TPR	0.6566
TNR	0.6241

According to the graph, the optimal threshold for the logistic regession model is 0.2. This yields a misclassificatio n rate of 0.36, TPR of 0.65, and TNR of 0.63. I chose that as a optimal threshold because it is generally where the TPR and NPR. In addition, this is most likely the case because the total number of Yes class is 25% of the total data set (both train and test).

Model 2: K Nearest Neighbor (KNN) Using the *knn* function, I created a KNN model and ran a for loop to iterate through a sequence of k values to use in my model.



According to the above results, a k=1 is the best fit for the model where the misclassification rate is the lowest, and TPR is the highest. However, this is most likely picking up on noise as this would be very risky to use to predict sleep toruble with a new sample set. Therefore, I chose k=5 for the best fit model.

Table 14: Error Rates K=5

Metric	Value
Misclass	0.2558
TPR	0.3434
TNR	0.8810

THe results of the KNN model are worse than the logistic regression in regards to the TPR, which is the target rate I am focusing on to determine the best model due to the imbalance of classes in the target variable, **SleepTrouble**.

Model 3: Linear Discriminant Analysis (LDA) For the LDA model, I tested 3 of the 4 methods:

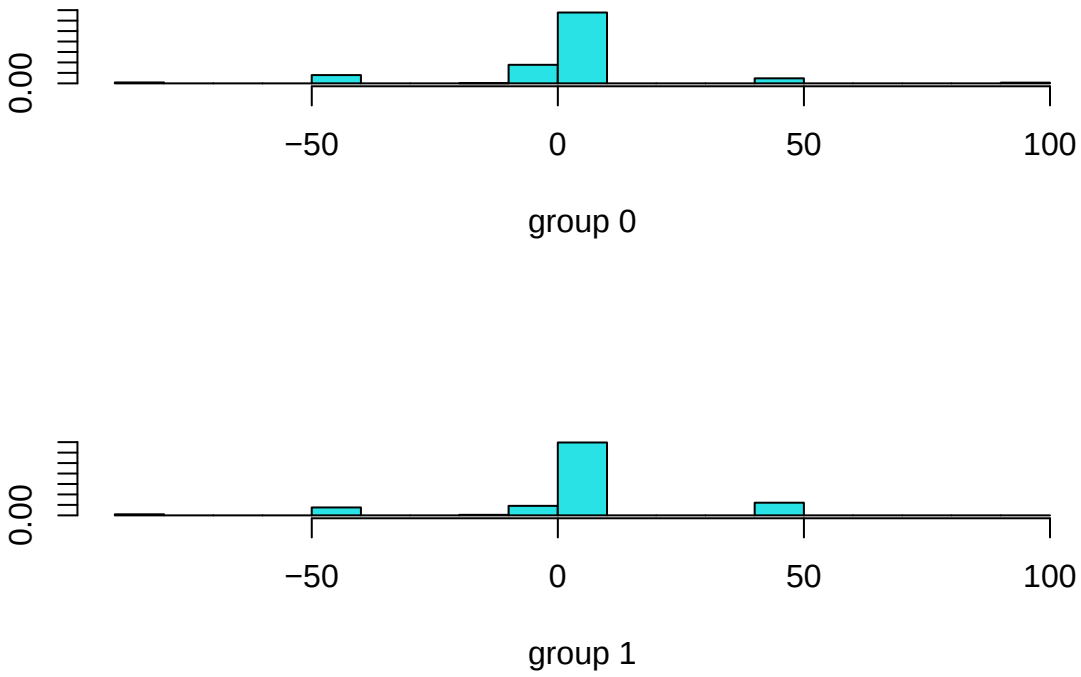
- *Moment*: Standard estimators of the mean and variace
- *mle*: Maximum Likelihood Estimates
- *mve*: Minimum volue ellipsoid, used for high-breakdown robust estimator of multivariate location and scatter.

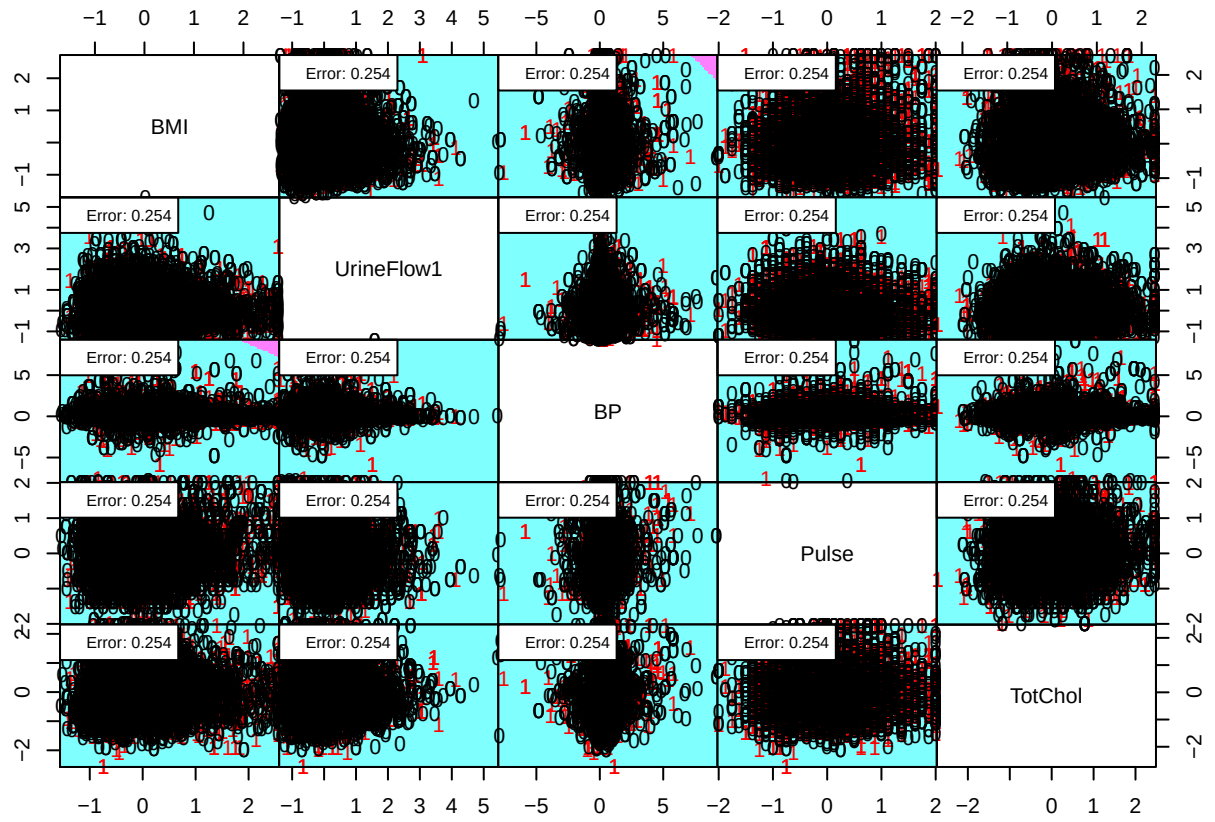
Of the 3 models, the one with the highest ROC was chosen as the best fitting model. I also plotted the targe class seperation hisograms to see if there are clear, linear clusters the model can predict, and also examined

the 5 most important variables and plotted those against the target to see if the model could linearly seperate the classes based on these.

Table 15: Error Rates All Methods

Method	Misclass	TPR	TNR
moment	0.2365	0.1717	0.9655
mle	0.2365	0.1717	0.9655
mve	0.2468	0.2323	0.9310





The best fitted model would be the *mve* method, however this is worse preforming than the logistic regression model while greatly favoring just the negative (zero) class of the target.

According to the analysis above, there does not appear to be a clear linear seperation in the model that would warrant the use of LDA. The histogram shows a near complete overlap of the 2 classes (SleepTrouble:Yes and SleepTrouble:No) with no visible shift in the means.

Also, when examining the graphs of the top 5 features of the model, I do not see any speration in the classes.

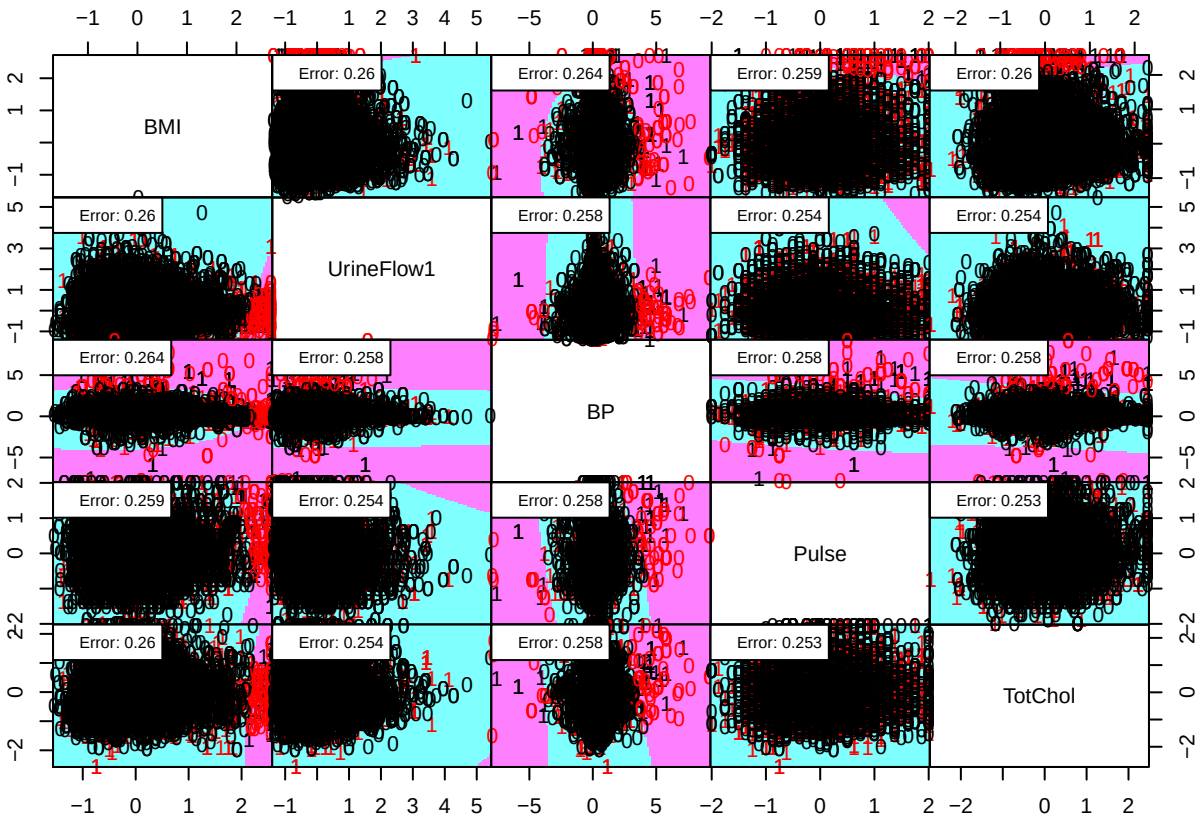
Model 4: Quadratic Discriminant Analysis (QDA) For the QDA model, I ran 3 methods to determine which one was the best fit:

- *moment*: Standard estimators of the mean and variance
- *mle*: Maximum Likelihood Estimates
- *t*: Robust estimates based on a t distribution

I then chose the best fitting model based on a TPR score and then graphed the top 5 features of the model to see if there is a quadratic relationship between these and the target.

Table 16: Error Rates All Methods

Method	Misclass	TPR	TNR
moment	0.3560	0.4596	0.7069
mle	0.3560	0.4596	0.7069
t	0.3625	0.5354	0.6724



According to the analysis above, the t method prefoms the best with a 0.54. However, while the misclassification rate is better than the logistic regression model, the TPR is not. In addition, the graph does show some seperation between the classes on a number of the top 5 variables which is a great improvement from the LDA model. However, there is still overlap between the two and BP seems to have the largest seperation between classes.

Model 5: Neural Network For the neural network model, I attempted to tune the hyper parameters, however R does not handle neural network models well as it only uses a single core. While I would have preferred, and most likely gottne better results, with a random search hyper parameter tuner, I picked a range of *size*,*decay*, and *maxit*. I then ran a for loop each sequence of the hyper parameters and chose the best fit for each to determine the best fit nueral network model.

In addition, the **nnet** package does not allow you to use more than 1 hidden layer, however due to Rs ability to run high computational functions, this will take to long to run fi more hidden layers were added. However, if I was using Python, I would test it with 1-3 hidden layers and a sequence of number of hidden nodes per layer.

Table 17: Error Rates Across Size

Size	Misclass	TPR	TNR
5	0.2429	0.2778	0.9207
10	0.2314	0.3384	0.9155
15	0.2275	0.2980	0.9345
20	0.2339	0.2929	0.9276
25	0.3201	0.5758	0.7155
30	0.2224	0.4697	0.8828

Size	Misclass	TPR	TNR
35	0.3522	0.5556	0.6793
40	0.2853	0.5051	0.7862
45	0.3380	0.5859	0.6879

Table 18: Error Rates Across Decay

Decay	Misclass	TPR	TNR
1e-04	0.3393	0.6313	0.6707
1e-03	0.3380	0.5859	0.6879
1e-02	0.2982	0.5455	0.7552
5e-02	0.2995	0.5303	0.7586
1e-01	0.2429	0.4343	0.8672

Table 19: Error Rates Across Maxit

Maxit	Misclass	TPR	TNR
10	0.2455	0.1313	0.9672
100	0.3380	0.5707	0.6931
500	0.3419	0.5909	0.6810
1000	0.3380	0.5859	0.6879
2000	0.3380	0.5859	0.6879

According to the hyperparameter tuning, the best fit model is a size of 45 hidden nodes, a decay rate of 0.0001, and max iterations of 500. Therefore I will use these hyper parameters to fit my neural network and adjust the thresholds to see if this can improve.

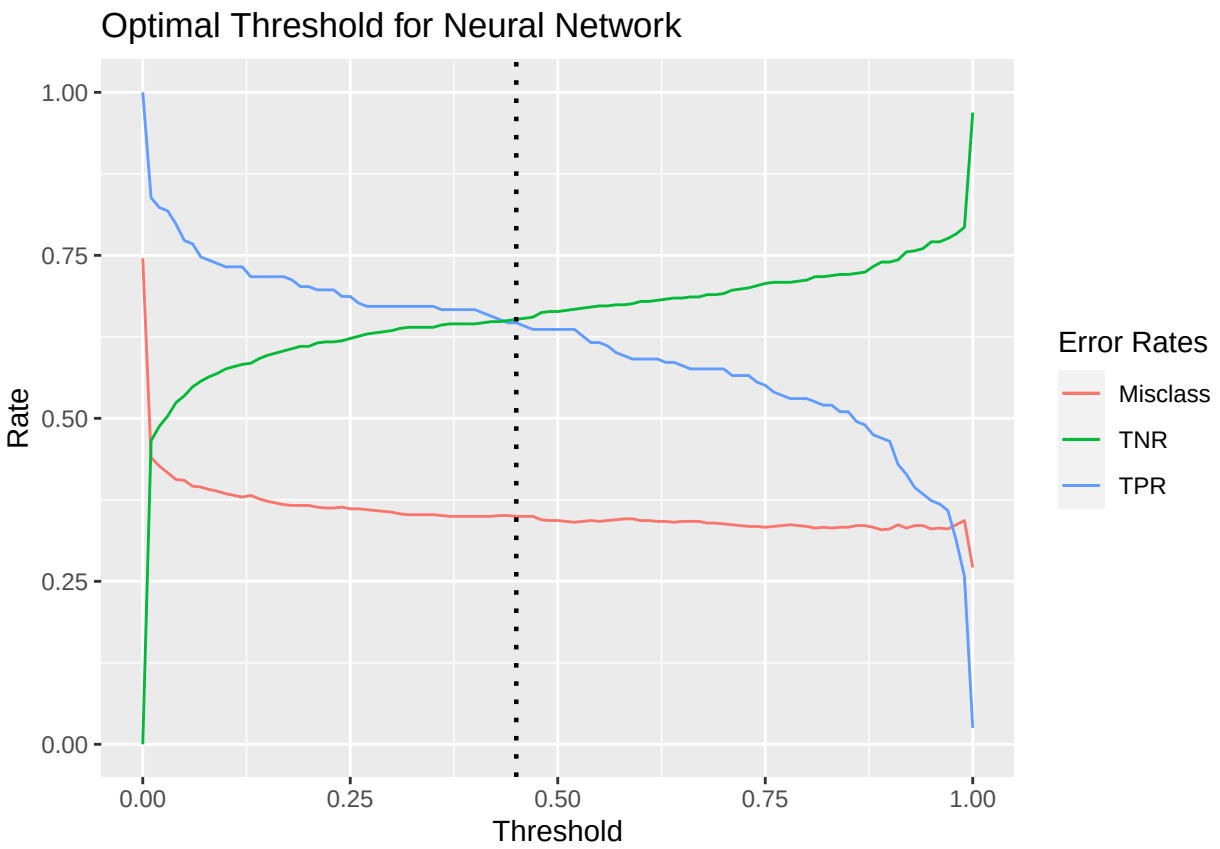


Table 20: Error Rates Threshold=0.45

Metric	Value
Misclass	0.3496
TPR	0.6465
TNR	0.6517

According to the threshold analysis, the optimal threshold level is at 0.45. This now has a close split between each of the classese and out preforms the linear regression model. In addition, we did not have to adjust the threshold as much as the logisti regression model to even out the TPR and TNR, which mean it does a better job at fitting imbalanced class targets.

Question 2

What classifier do you recommend from Exercise 1 and why?

Answer

I will compare each of the optimal tuned models to each other using misclassification rate, TPR, and TNR to determine the best fit model recommendation. I also will show the nueral network at a threshold of 0.45.

Table 21: Model Summary

Measure	Logistic	KNN	LDA	QDA	NNet
Misclass	0.3676	0.2558	0.2468	0.3625	0.3496
TPR	0.6566	0.3434	0.2323	0.5354	0.6465
TNR	0.6241	0.8810	0.9310	0.6724	0.6517

According to the summary above, I would chose the neural network as my final model. While the logistic regression isn't that much of a difference, I am concerned about the residual plots and the lack of randomness in the residuals as well as the large adjustment of the threshold to even out the TPR and TNR. I believe if I was able to modify the number of hidden layers, as well as tune the other hyper parameters better, the neural network may preform even better.

In addition, I would also try using a different scaling method, and swap out some of the variables I dropped. For example, I would use **Age** instead of **AgeDecade** to see if that improved the model fit.

Finally, I learned that sleep trouble appears to be mostly caused by health issues such as being overweight, bladder issues, and elevated heart rate. I also noticed that there is not one or 2 causes to sleep trouble. That different health issues cause sleep problems for different people as I noticed that the model improved when I used more of the top important variables then less (for example the top 20 vs. the top 10 or 5). Also, I would look at more variable (such as polynomial) conversions and see if more interaction variables are more useful.

Sources

lda: Linear Discriminant Analysis, Quadratic Discriminant Analysis, How to show code but hide output in RMarkdown?, varying classification threshold to produce ROC curves., Package 'nnet', predict.nnet: Predict New Examples by a Trained Neural Net, Classification: LDA and QDA Approaches, A Gentle Introduction to Threshold-Moving for Imbalanced Classification, How is Variable Importance Calculated for a Random Forest?, Feature Importance - How to choose the number of best features?, FEATURE SELECTION TECHNIQUES WITH R, Making dummy variables with dummy_cols(), N/A (Not Applicable) cases, How to deal with it? - duplicate, Remove Multiple Values from Vector in R (Example), How should we handle the missing values in test data?, and Subset columns using their names and types