Predicting Alzheimers Through Statistical Learning

Presented to

CSUF

COLLEGE OF

Natural Sciences and Mathematics

California State University, Fullerton

Math 533 Fall 2023

Dr. Sam BEHESTA

Prepared by

Paul LOPEZ

Emilio VASQUEZ

September 9, 2023

1 Introduction and Overview

As requested, this analysis will focus on the Alzheimer's data set provided by the National Alzheimer's Coordinating Center (NACC).

The task at hand requires us to build:

- Multiple regression model to predict a response variable of the ratio of hippocampus volume to the total intracranial volume as a function of features such as MMSE score , motor disturbance severity, disinhibition severity, anxiety severity, GDS Score, subject systolic blood pressure, subject diastolic blood pressure, heart rate, age, years of education, female or not, height, and weight.
- Logistic regression model to predict the binary version of a 3 class diagnosis reduced to normal cognition or impaired cognition.
- Multinomial logistic regression with the initial 3 categories of response of normal cognition (0), mild cognitive impairment due to AD (1), or dementia due to AD (2).

Through starting off with a saturated model that had over 13 predictors and performing variable selection by analyzing statistically significant predictors and performing various forms of cross validations, models were tuned and saw moderately successful results when used for prediction.

2 Analysis

2.1 Exploratory Data Analysis

The dataset itself is a comprehensive in nature with 3 overarching groups of data known as clinical and demographic features, neuropsychological and behavioral variables, and MRI features. All together, there are a total of 2,700 patients within the data with 55 different variables to build models with. Exploratory data analysis was performed on each group of variables as each of these groups offer some insight and predictive power into understanding impairments to cognitive ability and decline due to Alzheimer's disease, or AD, as frequently referred to throughout this report.

2.1.1 Clinical and Demographic Features

Clinical and demographic features provide information about the subject such as height, weight, resting heart rate, and education. This portion of the data serves as a nice introduction that allows one to investigate common notions about AD and dementia.

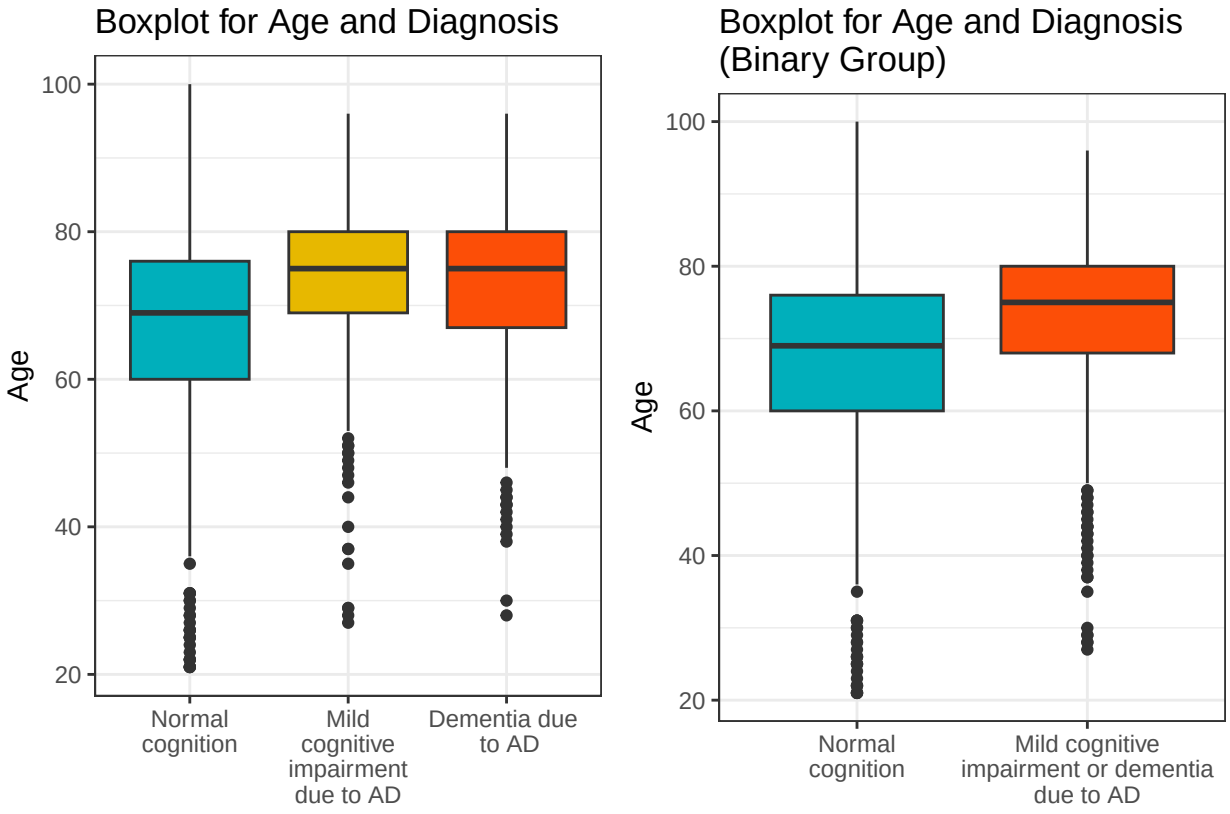


Figure 1. Boxplot for the different diagnosis groups and age

One common association with AD is age where the general consensus is that the older an individual is, the higher of a risk they are at to develop it. This sentiment can be confirmed in looking through the boxplots. The group who has mild cognitive impairment and the group who has dementia have an inter quartile range that covers individuals who are older when compared to the normal cognition group as shown in the box plot on the left.

One of the requested tasks required was to transform the three groups of normal cognition, mild cognitive impairment, and dementia into a binary group as shown on the figure on the right. This will be a common procedure performed throughout this exploratory data analysis. The median age for the normal cognition group is considerably lower than those with mild impairment and dementia.

Below, one can see the distribution of the age groups and their mental status. The majority of the dataset contains individuals who are past their mid to late 50s.

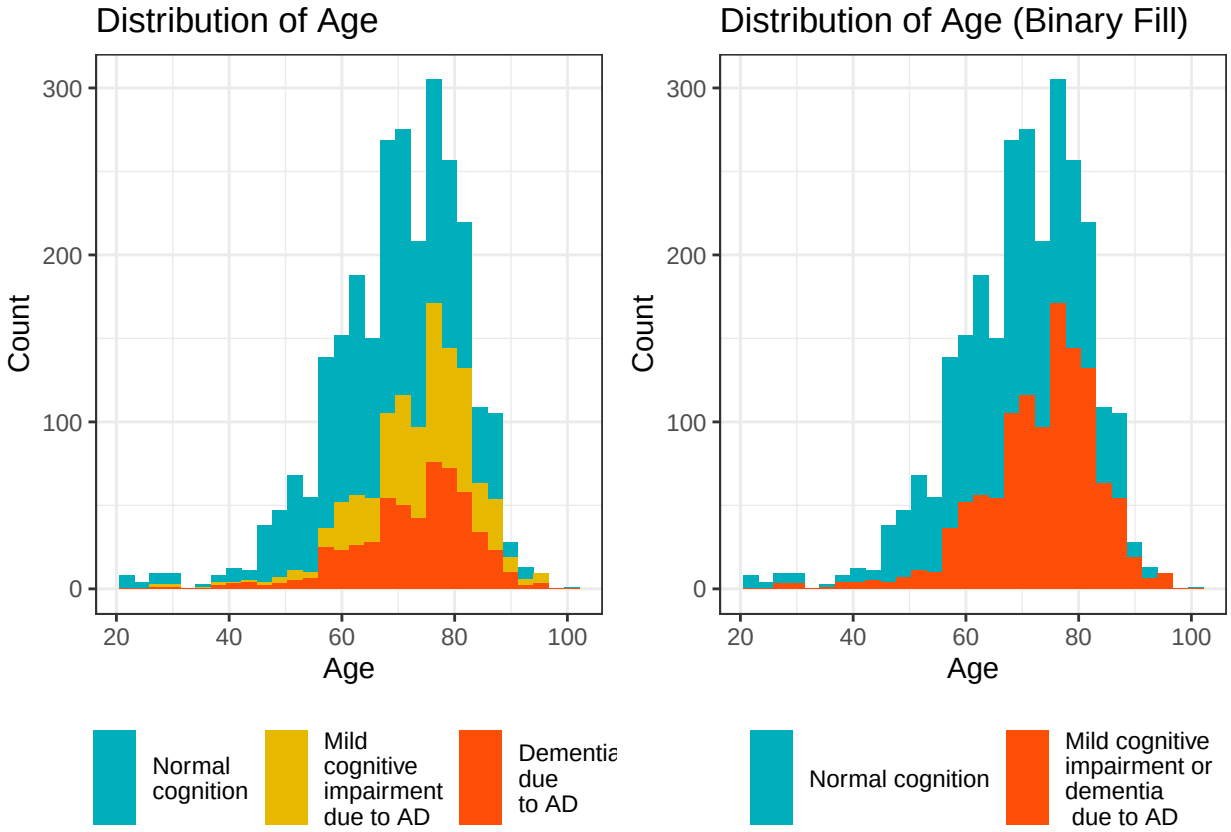


Figure 2. Distribution of age in the dataset

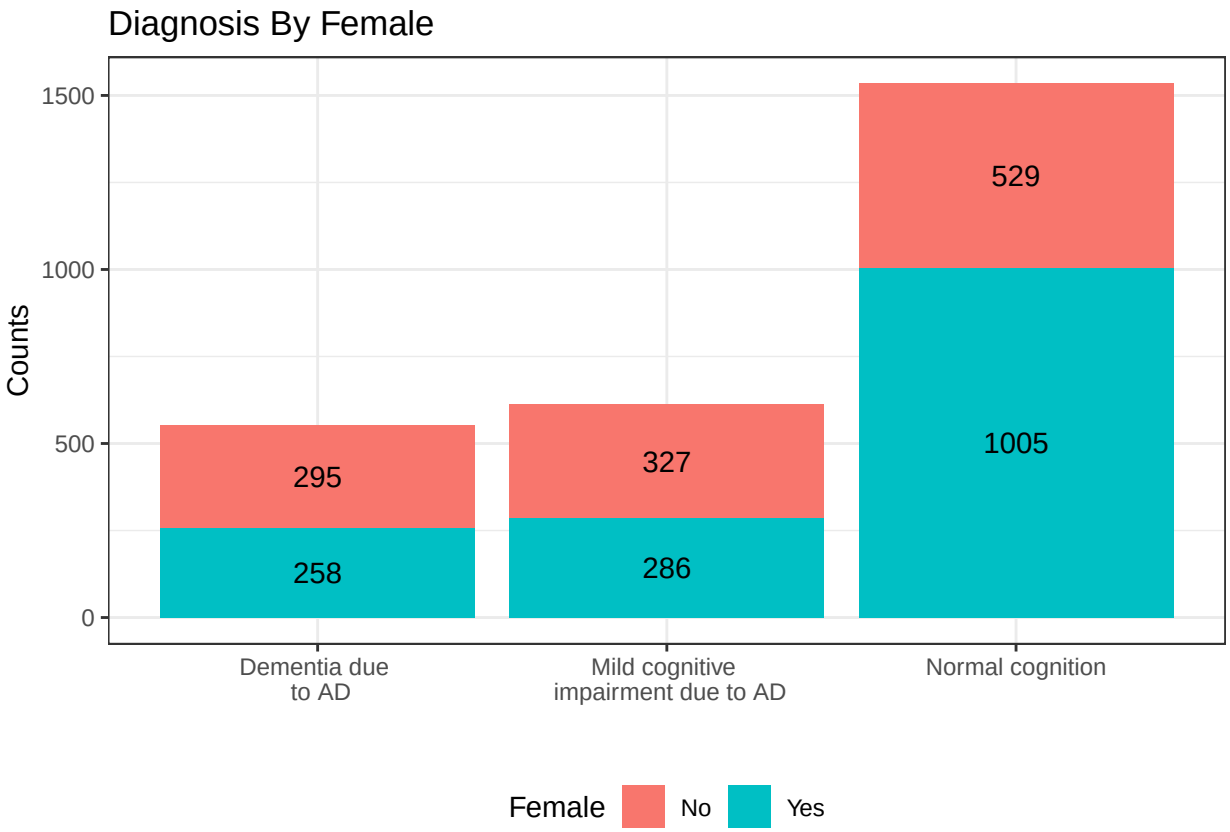


Figure 3. Sex of patients in the dataset with their respective group classification

One predictor available in the dataset is whether or not the subject is female. Overall, there is a good balance between female (1549) or not (1151). From the figure above, it appears that dementia and mild cognitive impairment due to AD are more prevalent in non-females than females in the provided data.

Additional data analysis can be found in the appendix for clinical and demographic features.

2.1.2 Neuropsychological and Behavioral Variables

Neuropsychological and behavioral variables capture subject data either through a survey questionnaire or through observation from a medical professional that capture behavioral or neuropsychological information on a subject in either a continuous or discrete format. Questions in the discrete group have a schema that allow n-possible question responses. When analyzing severity-related questions of medical conditions, a subject has the option to select normal, mild, moderate, or severe. An example group of questions from the data that follow this schema is present below.

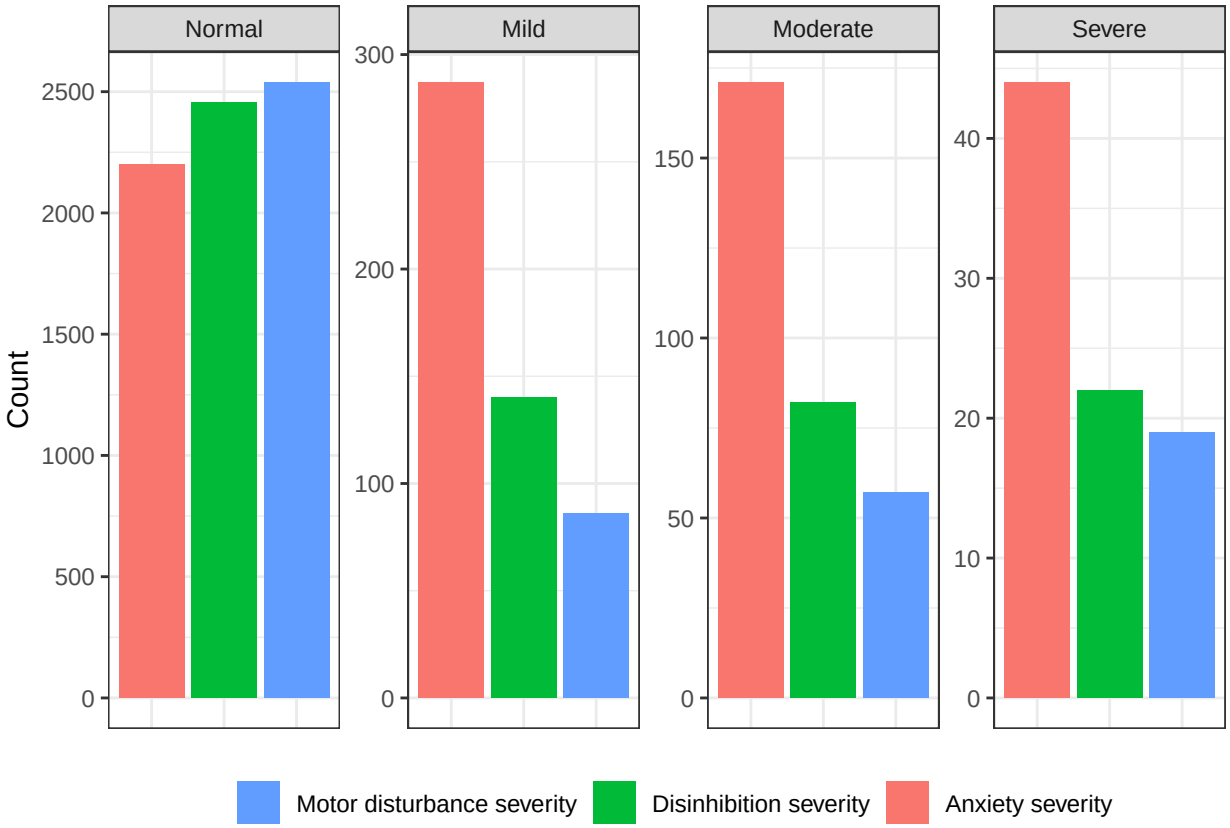


Figure 4. Survey questions part of the provided linear model in following sections

The data provides level of severity information on a subject's motor disturbance, disinihibition (or the ability to withhold an inappropriate/unwanted behavior), and anxiety. The majority of the respondents in the data set were marked normal level of severity for these three questions. Patients were marked having mild, moderate, and severe level of anxiety at a higher rate than the motor disturbance and disinhibition questions.

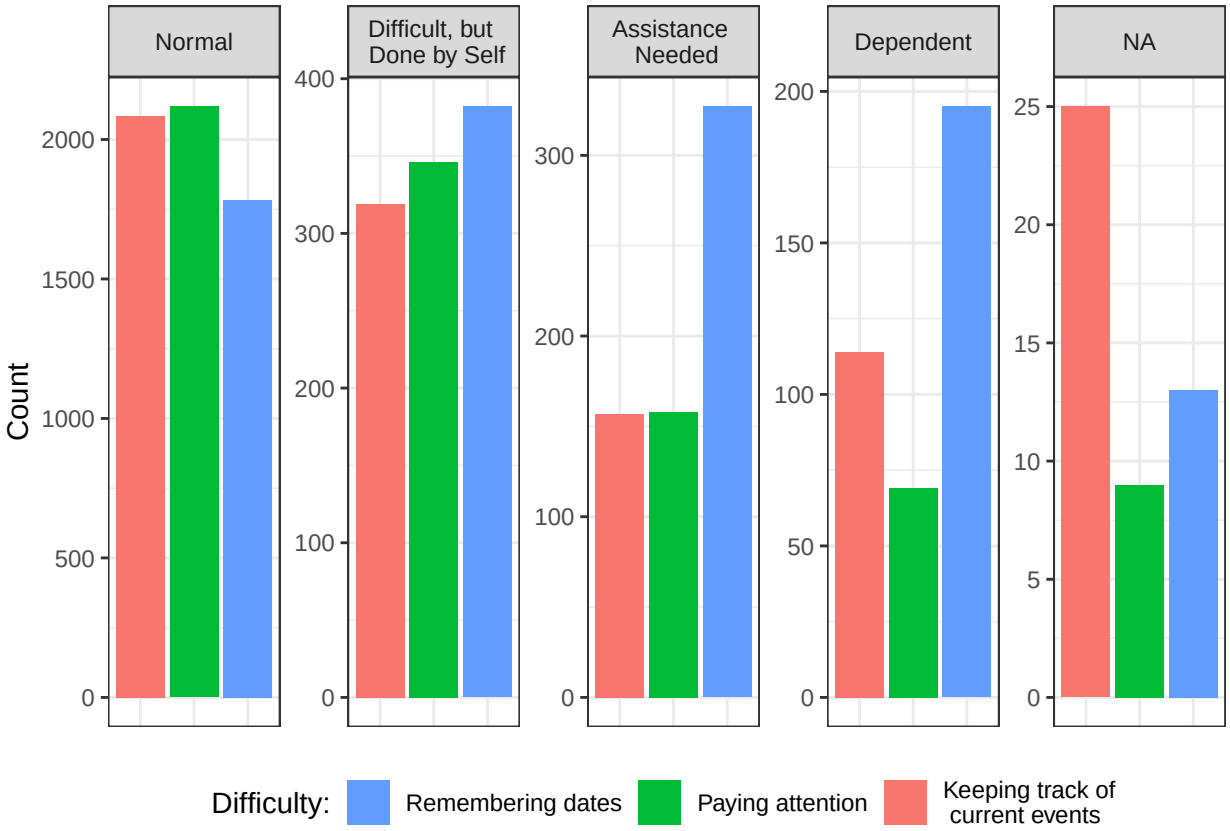


Figure 5. Questions relating to difficulty of tasks

There are another set of questions within the data that measure the difficulty in performing tasks. The majority of the respondents have normal level of difficulty in remembering dates, paying attention, and keeping track of current events. Remembering dates is where subjects had the most difficulty in performing these subset of tasks. Subjects marked that remembering dates was difficult but done by self, assistance needed, and dependent more often than the paying attention and keeping track of current events questions. When comparing those who marked assistance needed and dependent for remembering dates, subjects marked these answer choices by almost double (blue bars) compared to paying attention and keeping track of current event questions as shown in the 3rd and 4th plots from the left in figure 5.

A continuous variable found within the neuropsychological and behavioral variables would be the Mini-Mental State Examination (MMSE) which comprises of 11 questions that doctors use to check for cognitive impairment such as thinking, understanding, memory, and communication. The score ranges from 0 to 30. Anything above 25 is could be considered normal by a medical professional and anything below 25 is considered to be abnormal, indicating possible cognitive impairment.

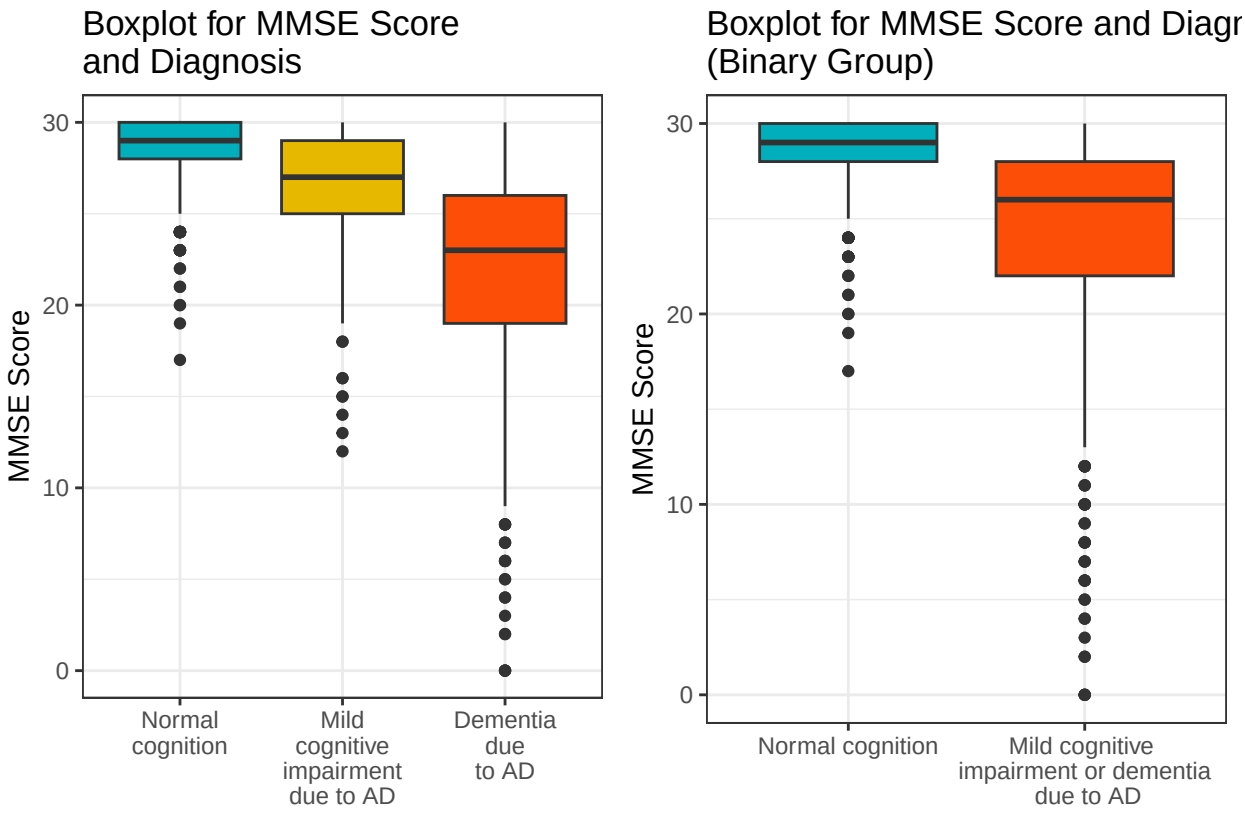


Figure 6. Boxplot of total intracranial volume and diagnosis

In the figure above, the provided dataset confirms this notion largely. In the leftmost boxplot, the whisker of patients diagnosed with normal cognition have a score of above 25. There are a few outliers that fell below this and that had a MMSE score as low as almost 15 but were deemed to have normal cognition. As the diagnosis worsens, one can see that the median (denoted by horizontal line in box) and the box and whisker drops lower and lower. Focusing on the right boxplot where the subjects were classified into a binary grouping, a similar story can be seen. There is quite a disparity between the two medians.

2.1.3 MRI Feature Variables

MRI features predictors is data there was gathered from MRI diagnosis that pertain to the brain related metrics such as regional gray matter volumes and the regional cortical thicknesses.

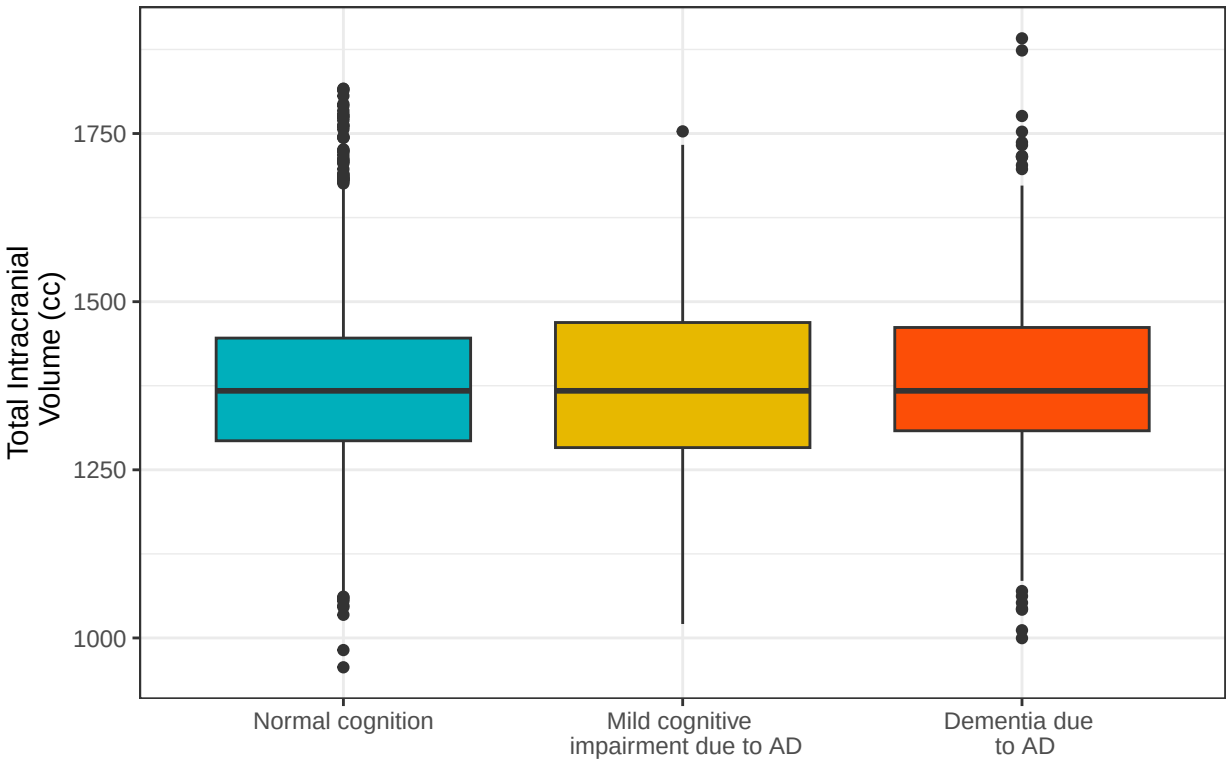


Figure 7. Boxplot of total intracranial volume and diagnosis

In the figure above, one can see visually that there is not much variation in the box plots between total intracranial volume and the diagnosis. The median is just about the same for each diagnosis as are the inter quartile ranges.

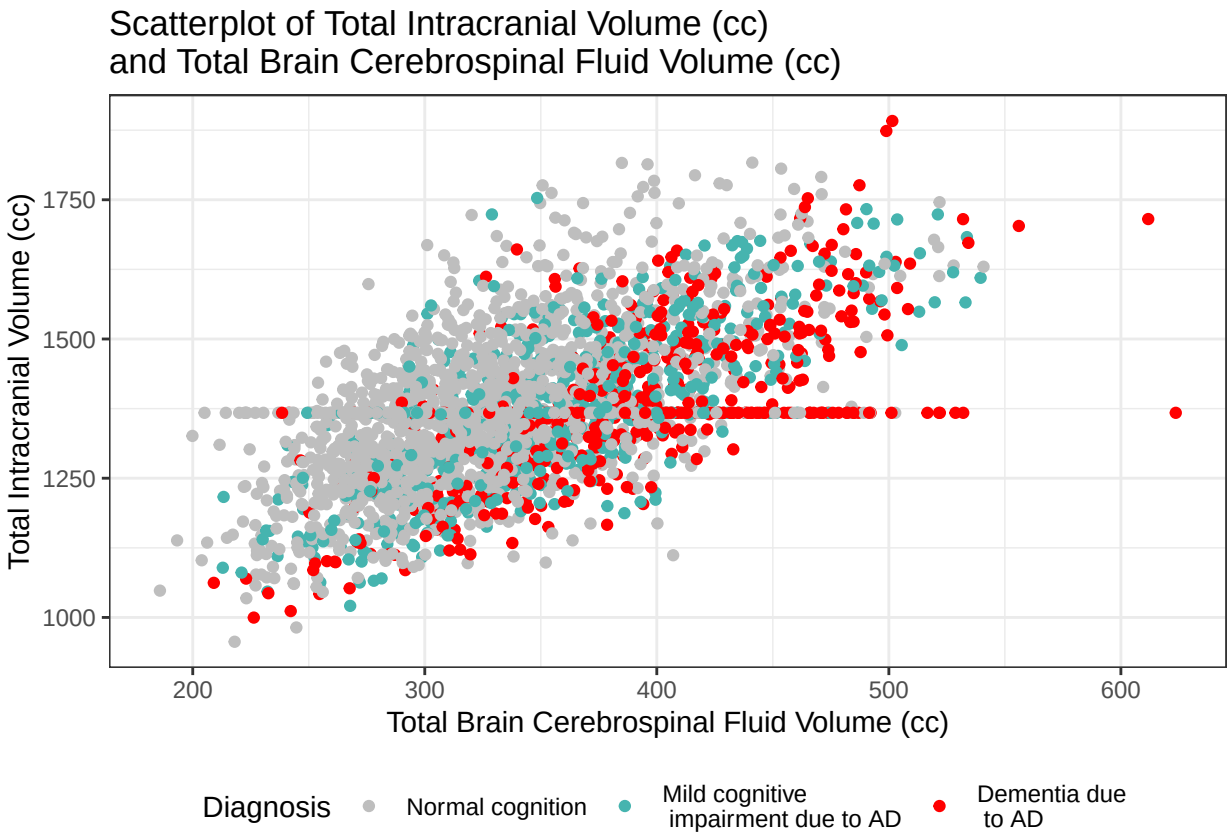


Figure 8. Boxplot of total intracranial volume and diagnosis

When looking at the scatter plot between total intracranial volume and total brain cerebrospinal fluid volume, there is a some-what clear division between the diagnosis states. As the brain cerebrospinal fluid volume and total intracranial volume increase, the subjects experienced mild cognitive impairment and dementia more so than subjects who had a lower brain cerebrospinal fluid volume and lower total intracranial volume. Between the 200 and 400 cc in total brain cerebrospinal fluid volume, there is a cloud of gray, which signifies normal cognition.

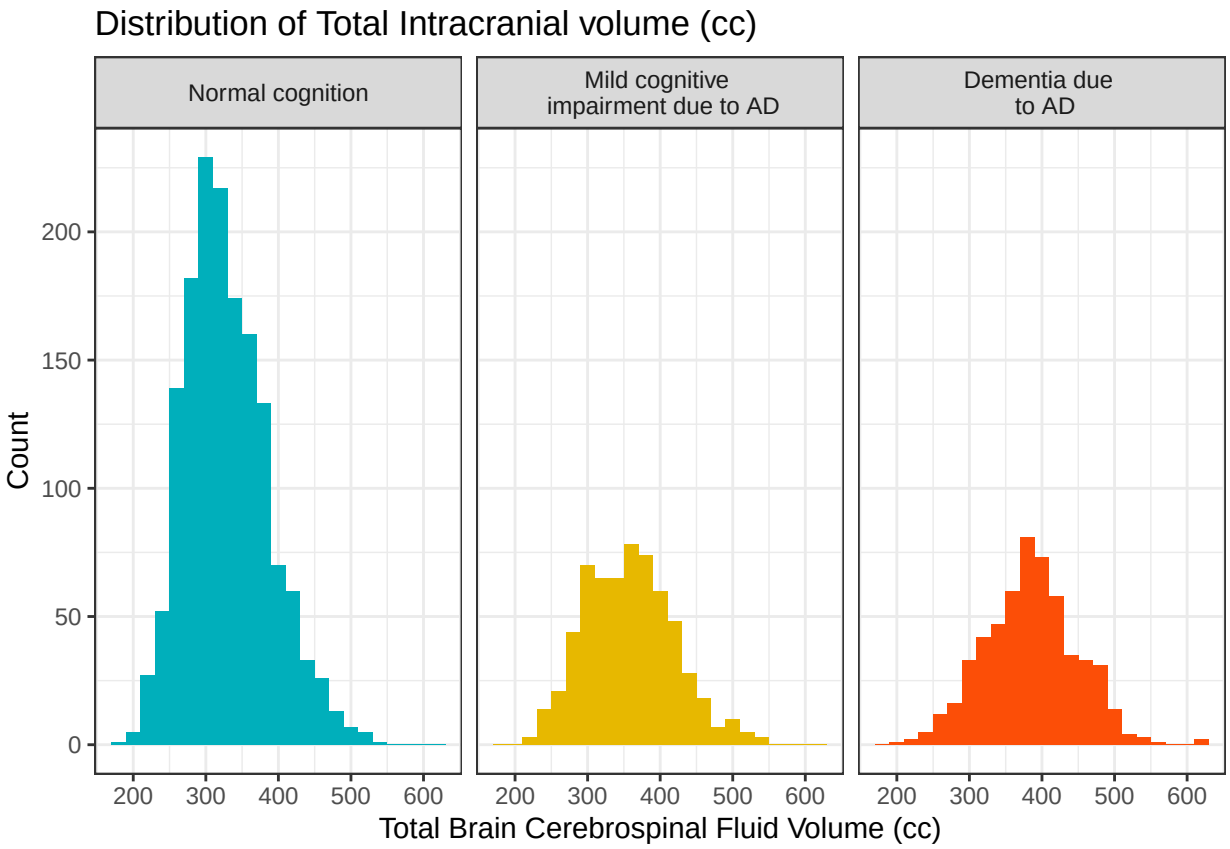


Figure 9. Distribution of intracranial volume for each diagnosis group

Exploring the total intracranial volume for each diagnosis group, those with normal cognition had a distribution that skewed to the left. As the diagnosis progresses to dementia, the distribution of the data begins to center around 400 as shown in the right-most graph.

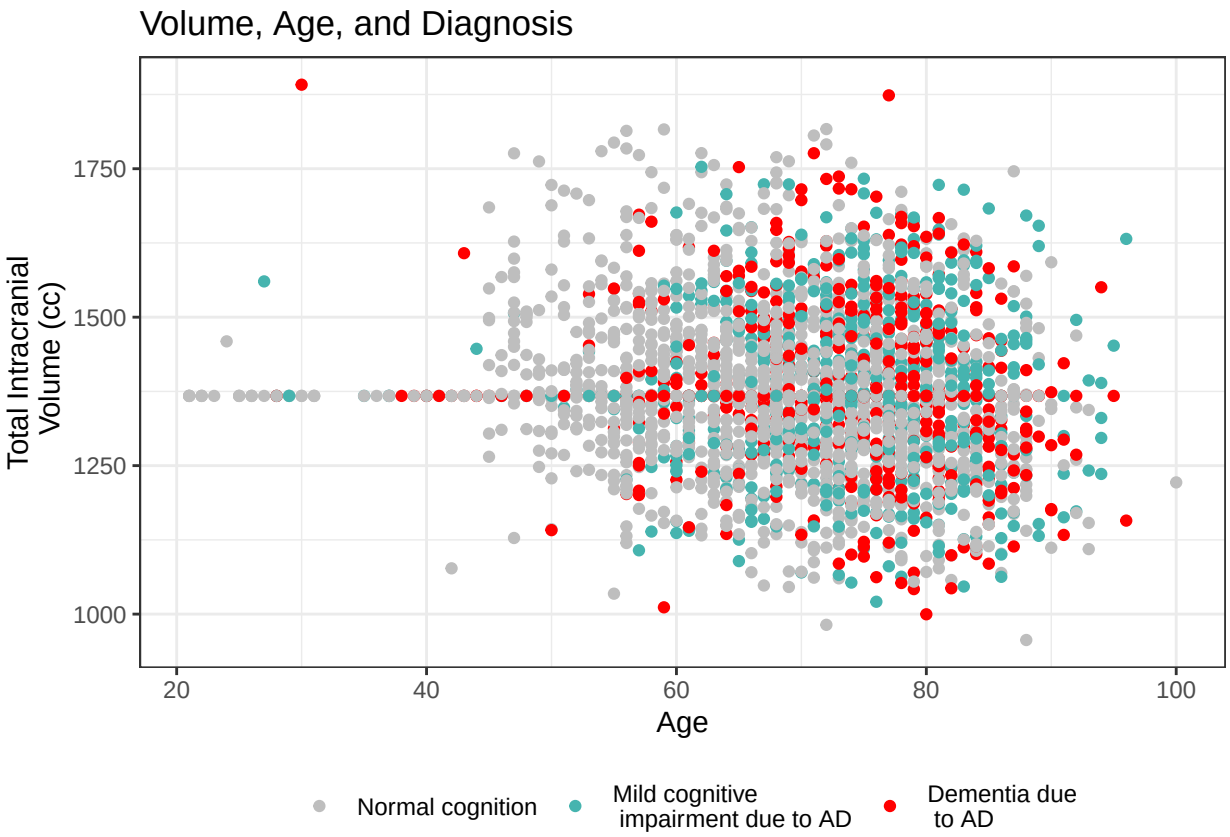


Figure 10. Total intracranial volume and age, two common factors

In the figure above, one can see that as age increases, the total intracranial volume decrease. The ellipse like formation of points trends downward. From a diagnosis classification perspective, there is not a discernible boundary in the color, or diagnosis of patient, of the points.

3 Building Models

Now that exploratory data analysis has been completed, one can begin to build models for prediction and inference. The task at hand was to produce three models. The first is to estimate the ratio of left hippocampus volume to total intracranial volume using a multiple regression model. Next is to build a logisitic model that classifies a binaried diagnosis (normal cognition or not). Lastly, a multinomial logistic regression model is utilized using the three possible diagnosis as the response.

3.1 Multiple Regression Model

As requested, the following predictors were used in the saturated model to model ratio of left hippocampus volume to total intracranial volume.

- Total MMSE score (naccmmse)
- Motor disturbance severity (motsev)
- Disinhibition severity (disnsev)
- Anxiety severity (anxsev)
- Total GDS Score (naccgds)
- Subject blood pressure (sitting), systolic (bpsys)
- Subject blood pressure (sitting), diastolic (bpdias)
- Heart rate of patient (hrate)
- Age of patient (age)
- Education years (educ)
- Female or not (female)
- Patient's height and weight

Many of these predictors were analyzed in the exploratory data analysis portion of this report. An initial saturated multiple linear regression model created to start that consisted of all these variables to predict the requested ratio. More information pertaining to this model can be found in the appendix.

3.2 Final Multiple Regression Model and Feature Selection

Once this initial linear model was built, variable selection was performed by looking at the model summary as a starting point. Variables that had the least statistical significance in the model were dropped to help determine the final model. Variables such as motor disturbance severity (motsev), total GDS Score (naccgds), Subject blood pressure diastolic (bpdias), heart rate of patient (hrate) had high p-values in the linear model which means they did not have much statistical significance in the context of the linear regression model. So, these variables were removed for the final model to be a function of the following :

```
##
## Call:
## lm(formula = second_lm_model_pred, data = data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.668e-03 -1.774e-04  2.615e-05  1.948e-04  1.191e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.771e-03  1.723e-04  16.084  < 2e-16 ***
## naccmmse     2.122e-05  1.539e-06  13.788  < 2e-16 ***
## disnsev1    -5.211e-05  2.685e-05  -1.941   0.0524 .
## disnsev2    -2.079e-05  3.479e-05  -0.597   0.5502
## disnsev3    -6.926e-05  6.609e-05  -1.048   0.2947
## anxsev1     -6.135e-05  1.931e-05  -3.178   0.0015 **
## anxsev2     -4.308e-05  2.501e-05  -1.723   0.0850 .
## anxsev3     -8.826e-06  4.729e-05  -0.187   0.8520
## bpsys        4.530e-07  3.303e-07   1.371   0.1704
## age         -7.398e-06  5.565e-07 -13.292  < 2e-16 ***
## educ        -7.240e-06  1.855e-06  -3.902 9.77e-05 ***
## female1      2.856e-05  1.666e-05   1.715   0.0865 .
## height      -1.115e-05  2.335e-06  -4.773 1.91e-06 ***
## weight       9.453e-07  1.990e-07   4.751 2.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0003005 on 2686 degrees of freedom
## Multiple R-squared:  0.1951, Adjusted R-squared:  0.1912
## F-statistic: 50.09 on 13 and 2686 DF,  p-value: < 2.2e-16
```

Figure 11. Model summary of the final linear model. Final selection of variables can be seen under the coefficients portion of the readout.

By looking at the p-values to the right. One can see that the variables that had the most statistical impact on ratio of left hippocampus volume to total intracranial volume were total MMSE score

age, years of education, height, and weight as denoted by the asterisk "*". The more asterisk present, the more statistically significant the variables are.

3.2.1 Model Selection Methodology

To ensure that the final model above was a more competitive model compared to the saturated model, various forms of cross validation were performed utilizing sum of squared residuals (SSR) as the objective measure of fit. The lower the SSR, the better fitting the model was deemed. Below are the results of the cross-validation procedure:

	Leave One Out CV	Single Fold CV	5 Fold CV	10 Fold CV
Saturated Model	0.000246763	0.000274918	0.0002462672	0.0002463828
Final Model	0.0002459618	0.0002745154	0.000245021	0.0002456736

Figure 12. Results of cross-validation procedures

In every cross validation scenario, the final model outperformed the saturated model by having a lower sum of squared residuals, but not by much.

3.2.2 Predictive Power of the Model

To assess the predictive power of this model, the results of the train/test split from the 10 fold cross-validation procedure were stored. So at each fold, the actual response, the predicted response, and the difference from actual and predicted were stored. Below are visual representations of how well the final model predicted the data in this setting.

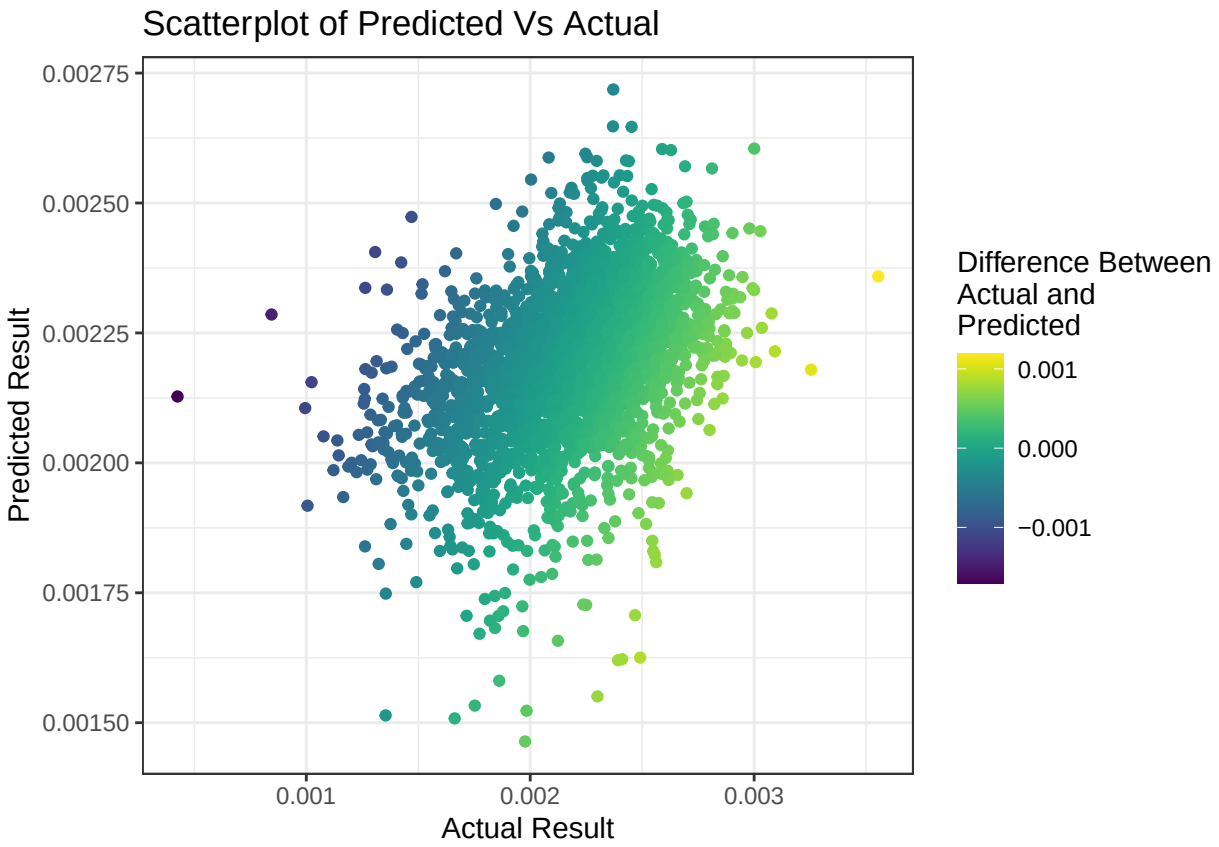


Figure 13. Scatterplot of predicted and actual results from the final model

The linear nature of the scatter plot is what we wanted. In an ideal scenario, the points would form a diagonal line (y=x) to indicate the predictions match the actual values. Looking at the color of the plots, the greener a point is, the smaller the difference was between the actual and predicted score.

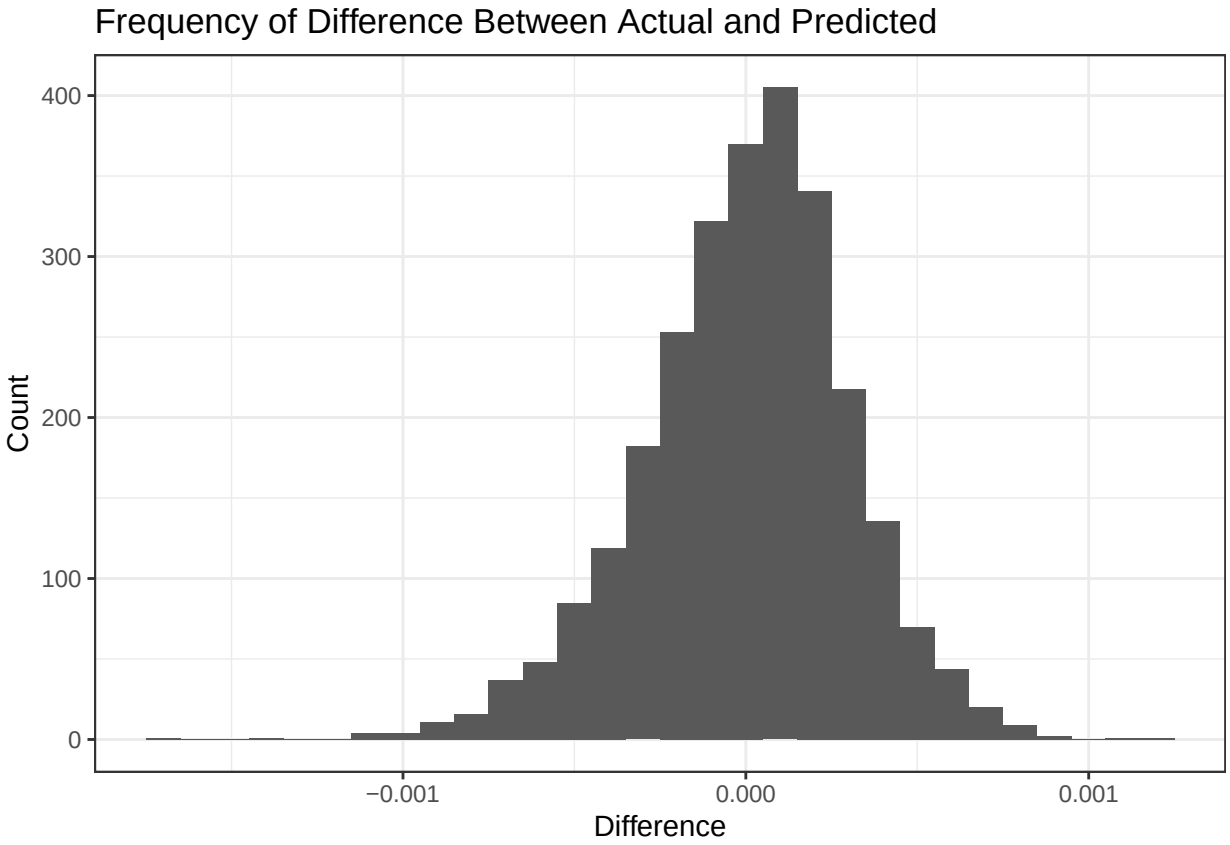


Figure 14. Histogram of the difference between actual and predicted response

Above is a distribution of the difference between the actual score and the predicted. The majority of the data fell between -0.001 and 0.001 difference which is quite small. The majority of the predictions fell around 0, which would suggest that the final model has some prediction power.

3.3 Predicting Binarized Version of Diagnosis Using Logisitic Regression

As requested, a logistic regression model was created to predict a binarized version of the diagnosis variable. The diagnosis variable of normal cognition, mild cognitive impairment due to AD, and dementia due to AD were reclassified with normal cognition remaining the same and mild impairment and dementia grouped into another group.

3.3.1 Final Logistic Regression Model

A saturated model was created utilizing the same predictors above with a model summary present in the appendix. Some predictors were not statistically significant in looking at the high p-values

and lack of asterisks. The variables were dropped from the logistic regression model as a result. In addition, some new variables were added to the model that were not originally present in the saturated model because of the exploratory data analysis completed earlier in the report. Variables such as total intracranial volume (cc) and total brain cerebrospinal fluid volume (cc) proved to be statistically significant by looking at the p-values in the finalized model below.

```
##
## Call:
## glm(formula = log_reg_pred_2, family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4898  -0.5079  -0.3046   0.2662   2.5881
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.4539312  1.5875981   5.955 2.60e-09 ***
## naccicv     -0.0018751  0.0006893  -2.720  0.00653 **
## csfvol       0.0101446  0.0015439   6.571 5.00e-11 ***
## naccmmse    -0.5513430  0.0353309 -15.605  < 2e-16 ***
## remdates1    1.9799632  0.1597222  12.396  < 2e-16 ***
## remdates2    2.7047843  0.2640912  10.242  < 2e-16 ***
## remdates3    3.9055306  0.7502706   5.205 1.93e-07 ***
## remdates8    0.6893111  0.8240388   0.837  0.40287
## naccgds      0.1440366  0.0271095   5.313 1.08e-07 ***
## age          0.0019801  0.0066343   0.298  0.76535
## educ         0.0551497  0.0210145   2.624  0.00868 **
## height       0.0627572  0.0213451   2.940  0.00328 **
## weight      -0.0092618  0.0020947  -4.422 9.80e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3692.7  on 2699  degrees of freedom
## Residual deviance: 1794.2  on 2687  degrees of freedom
## AIC: 1820.2
##
```

```
## Number of Fisher Scoring iterations: 7
```

3.3.2 Selecting the Final Logistic Regression Model

In order to solidify that the model above was a more competitive model over the saturated model, 5 fold cross validation was performed on the data. In each fold of the cross-validation, the test/train split was 80%/20%. At each fold, the predicted probability was captured and a probability of greater than 0.5 was deemed that the patient had cognitive issues. This 5 fold CV allowed every data point to be predicted once and to not be used in the training of the model.

	5 Fold CV Model Prediction Accuracy
Saturated Model	82.7%
Final Model	85.5%

By dropping some statistically insignificant predictors and adding a few more variables that showed potential predictive power in the exploratory data analysis, model accuracy was improved by almost 3 %. Since the final model had a higher prediction accuracy, that model was deemed more competitive than the saturated model.

3.4 Predicting Non-Binary Diagnosis using Multinomial Logistic Regression

Lastly as requested, a multinomial regression model was created to predict the diagnosis variable as presented. The task at hand is to build a model to predict a diagnosis of normal cognition, mild cognitive impairment due to AD, or dementia due to AD. To start, a saturated model using the same predictors as the previous two models were used to train the model. The model readout of this model can be found in the appendix.

3.4.1 Final Multinomial Regression Model

The final multinomial regression model below utilized the same predictors of the final logistic regression model due to having moderately successful results in being able to correctly predict the binary classification.

```
## # weights:  42 (26 variable)
## initial  value 2966.253179
## iter  10 value 2116.135340
```

```
## iter  20 value 1885.938957
## iter  30 value 1392.927596
## iter  40 value 1392.003339
## final  value 1392.003286
## converged

## Call:
## multinom(formula = multi_nom_pred_2, data = data)
##
## Coefficients:
##   (Intercept)      naccicv      csfvol   naccmmse remdates1 remdates2 remdates3
## 1    7.270469 -0.001296492 0.007927556 -0.5016280  1.791420  2.210199  3.227745
## 2   17.562428 -0.005004191 0.022513070 -0.8045597  3.057932  4.437394  5.792326
##   remdates8   naccgds         age       educ     height       weight
## 1 0.5077225 0.1471901  0.01180062 0.04616126 0.06263515 -0.008568416
## 2 1.7948999 0.1195533 -0.04903107 0.10761574 0.05943171 -0.013190357
##
## Std. Errors:
##    (Intercept)      naccicv      csfvol   naccmmse remdates1  remdates2
## 1 0.0008341763 0.0006886390 0.001511271 0.02980657 0.1425639 0.09054896
## 2 0.0006750984 0.0009893609 0.002166204 0.03729747 0.0786351 0.08319247
##   remdates3   remdates8    naccgds         age       educ     height
## 1 0.1142527 0.004268541 0.02698863 0.005887011 0.02104108 0.01728905
## 2 0.1137708 0.003097064 0.03695725 0.008181282 0.02992595 0.02376467
##        weight
## 1 0.002097661
## 2 0.003171287
##
## Residual Deviance: 2784.007
## AIC: 2836.007
```

3.4.2 Selecting the Final Multinomial Regression Model

In order to confirm the initial hypothesis that the predictors used in the final logistic regression model was more competitive than a saturated multinomial model, 5 fold cross-validation was completed again. Similar as before, this procedure utilized a 80%/20% train/test split. During each fold, the test set estimated probabilities were stored. Since there are 3 different possible responses, the multinomial model predicted a probability that a given patient belongs to a group.

The group that had the highest estimated probability was used to classify that patient into that group.

	5 Fold CV Model Prediction Accuracy
Saturated Model	74.2%
Final Model	77.4%

This model saw an overall drop in model prediction accuracy but still had a similar performance difference in model prediction accuracy between the final and saturated model as the logistic regression model. As a result, the final model with the reduced predictors was the more competitive model in this scenario yet again.

3.4.3 Comparing and Contrasting the Logstic Regression Model and Mutinomial Regression Model

One key difference between these two approaches to predicting cognitive issues was drop in accuracy when predicting in a non-binary setting.

Variable	Logistic Coeff Est	Multinomial Coeff Est	Logistic Odds Ratio	Multinomial Odds Ratio
Intercept	9.4539312	7.270469	12758.22193	1437.224353
naccicv	-0.0018751	-0.001296492	0.998126657	0.998704348
csfvol	0.0101446	0.007927556	1.010196231	1.007959062
naccmmse	-0.551343	-0.501628	0.576175487	0.605544031
remdates1	1.9799632	1.79142	7.242476457	5.99796353
remdates2	2.7047843	2.210199	14.95109139	9.117530601
remdates3	3.9055306	3.227745	49.67643123	25.22271557
remdates8	0.6893111	0.5077225	1.992342536	1.66150281
naccgds	0.1440366	0.1471901	1.154926378	1.158574187
age	0.0019801	0.01180062	1.001982062	1.011870522
educ	0.0551497	0.04616126	1.056698791	1.047243276
height	0.0627572	0.06263515	1.064768282	1.064638335
weight	-0.0092618	-0.008568416	0.990780958	0.991468188

Between the two models, the coefficients and by extension the estimated log-odds were pretty close.

In interpreting the coefficients of the logistic regression, it is important to remember we are trying to model the probability of a binary outcome. We have the general model:

$$P(Y = 1|X) = \frac{1}{1 + exp(-\sum_{i=0}^{n}(\beta_i X_i))}$$

Assuming $X_0 = 1$, β_0 is our intercept term, and it means it is the log-odds of the even Y = 1 occurring when all the predictor variables, X_i, are 0. If you exponentiate β_0 you will get the odds ratio of the event occurring when all the predictors are 0.

In the same vein, β_i, where i > 0 represents the change in the log-odds for a one unit increase in X_i, holding all the other predictors constant. If you exponentiate β_i, you will get the odds ratio associated with a one-unit increase in X_i. As a quick example if we have the case where $e^{\beta_2} = 2$, then a one-unit increase in X_2 would double the odds of the event occurring, holding all else constant.

For a multinomial logistic regression we are trying to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables. Unlike the logistic regression previously discussed, we now have more than two categories for our dependent variable.

We have our dependent variable Y with K categories. We'll take (K-1) of these categories and compare it to a reference category (the one that remains). For n predictors $(X_1, ..., X_n)$ and a given category k our model can be represented as:

$$log\big(\frac{P(Y = k|X)}{P(Y = ref|X)}\big) = \sum_{i=0}^{n} \beta_{ik} X_i$$

Similar to before, β_{0k} is our intercept term for category k. This is the log-odds of the event of being in category k compared to the reference category when all predictors are 0.

For β_{ik} where i > 0, this represents the change in the log-odds of being in category k for a one-unit increase in X_i, while holding all other predictors constant. If you exponentiate β_{ik} (i.e. $e^{\beta_{ik}}$) you get the odds ratio that is associated with a one unit increase in X_1 for category k versus the reference category.

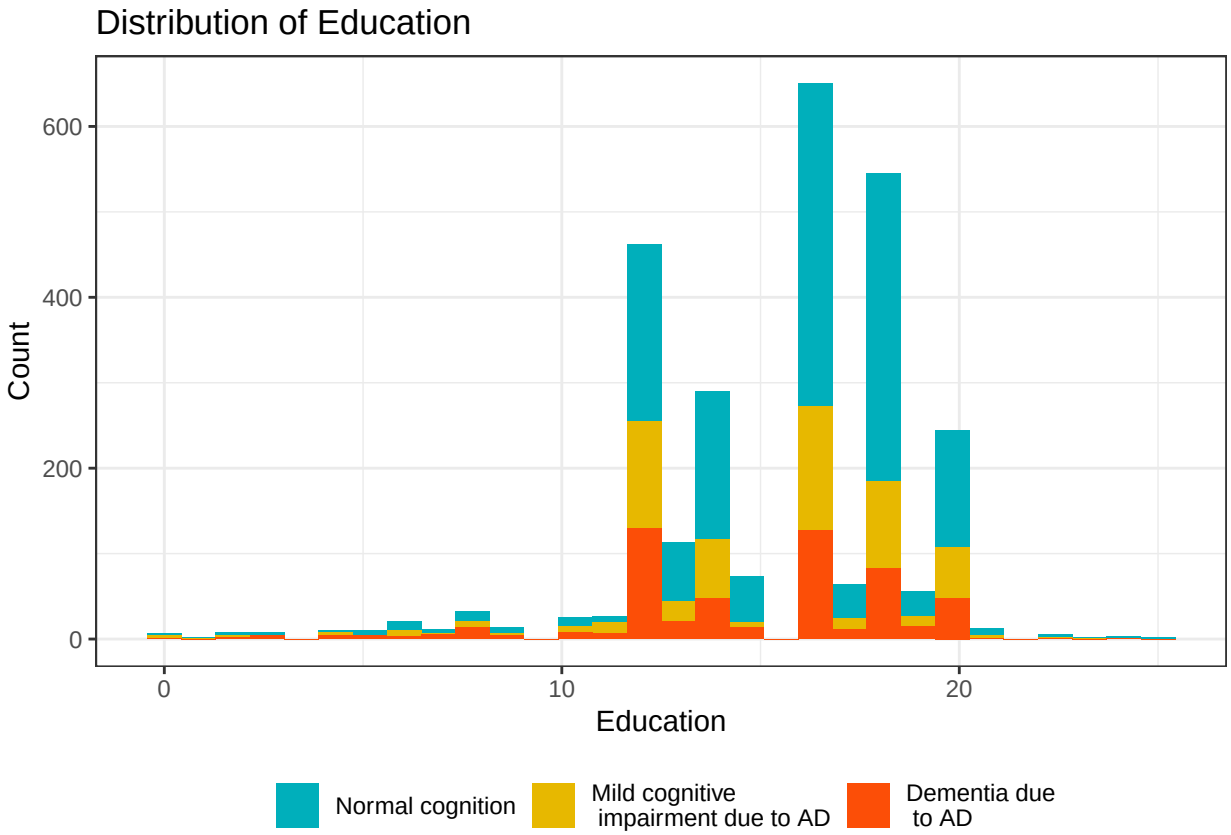
4 Next Steps

Overall the three final models were competitive when compared to their saturated model counterparts, but there is room for improvement particularly with the logistic and multinomial models. Although the performance of the models worked well at predicting data with prediction rates as low as 74%, it is possible that with some more advanced machine learning techniques such as random forest or support vector machines we may be able to make classifications of cognitive impairments better. However, this would come at the cost of interpretability of the model.

5 Appendix

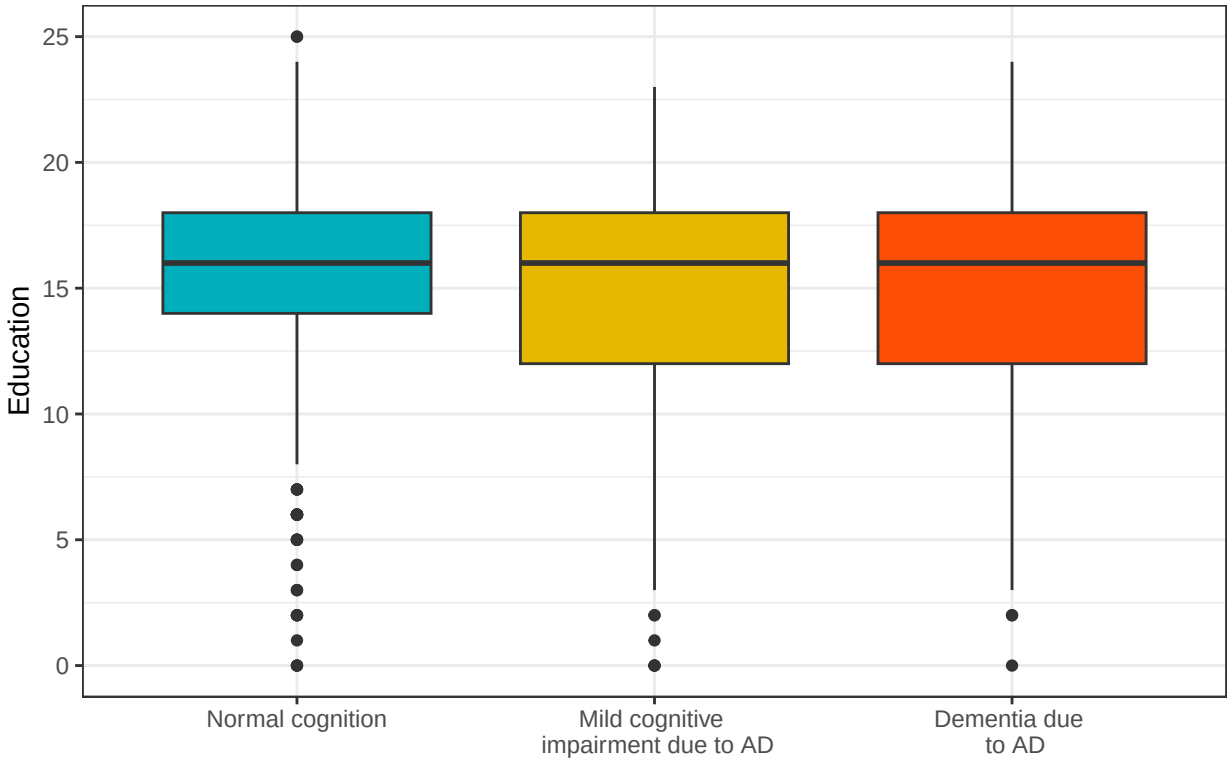
5.1 Additional Clinical and Demographic Features Data Analysis

5.1.1 Distribution of Education



It can be seen here that within the data, the majority of subjects had between 10 and 20 years of education.

5.1.2 Boxplot of Education and Diagnosis



In the boxplot above, there is potential evidence to suggest that those with lower years of education have mild cognitive impairment or dementia.

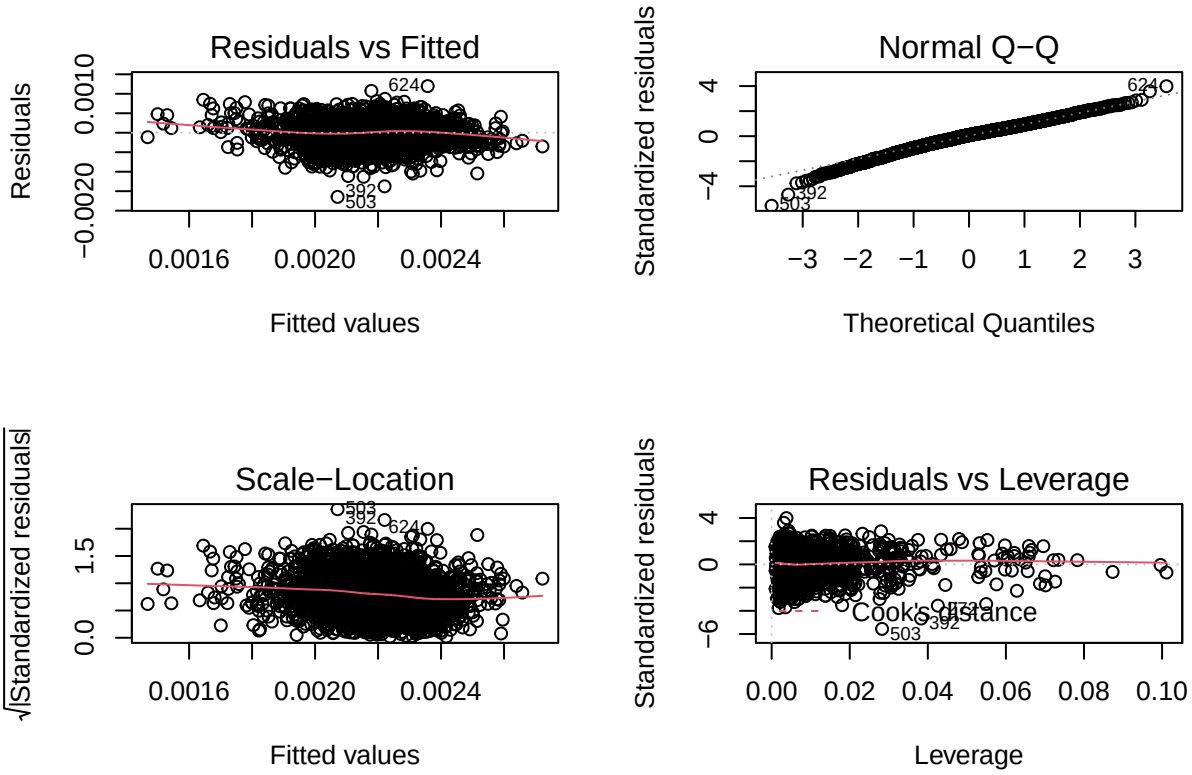
5.2 Linear Model

5.2.1 Saturated Linear Model Summary

```
##
## Call:
## lm(formula = sat_model_pred, data = data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.646e-03 -1.768e-04  2.574e-05  1.930e-04  1.197e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  2.765e-03  1.813e-04  15.252  < 2e-16 ***
## naccmmse     2.154e-05  1.579e-06  13.639  < 2e-16 ***
## motsev1     -2.903e-05  3.410e-05  -0.851  0.39473
## motsev2      9.049e-05  4.389e-05   2.062  0.03931 *
## motsev3     -8.784e-07  7.098e-05  -0.012  0.99013
## disnsev1    -5.330e-05  2.713e-05  -1.965  0.04952 *
## disnsev2    -3.533e-05  3.547e-05  -0.996  0.31940
## disnsev3    -7.749e-05  6.650e-05  -1.165  0.24396
## anxsev1     -6.037e-05  1.962e-05  -3.077  0.00211 **
## anxsev2     -4.738e-05  2.555e-05  -1.854  0.06381 .
## anxsev3     -3.007e-05  4.848e-05  -0.620  0.53515
## naccgds      6.364e-07  2.642e-06   0.241  0.80966
## bpsys        6.140e-07  3.830e-07   1.603  0.10904
## bpdias      -5.484e-07  7.018e-07  -0.781  0.43459
## hrate        1.419e-07  5.495e-07   0.258  0.79626
## age         -7.462e-06  5.766e-07 -12.942  < 2e-16 ***
## educ        -6.990e-06  1.877e-06  -3.724  0.00020 ***
## female1      2.901e-05  1.677e-05   1.729  0.08386 .
## height      -1.106e-05  2.339e-06  -4.729 2.37e-06 ***
## weight       9.484e-07  2.000e-07   4.741 2.24e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0003005 on 2680 degrees of freedom
## Multiple R-squared:  0.197,  Adjusted R-squared:  0.1913
## F-statistic:  34.6 on 19 and 2680 DF,  p-value: < 2.2e-16
```

5.2.2 Saturated Linear Model Diagnostics



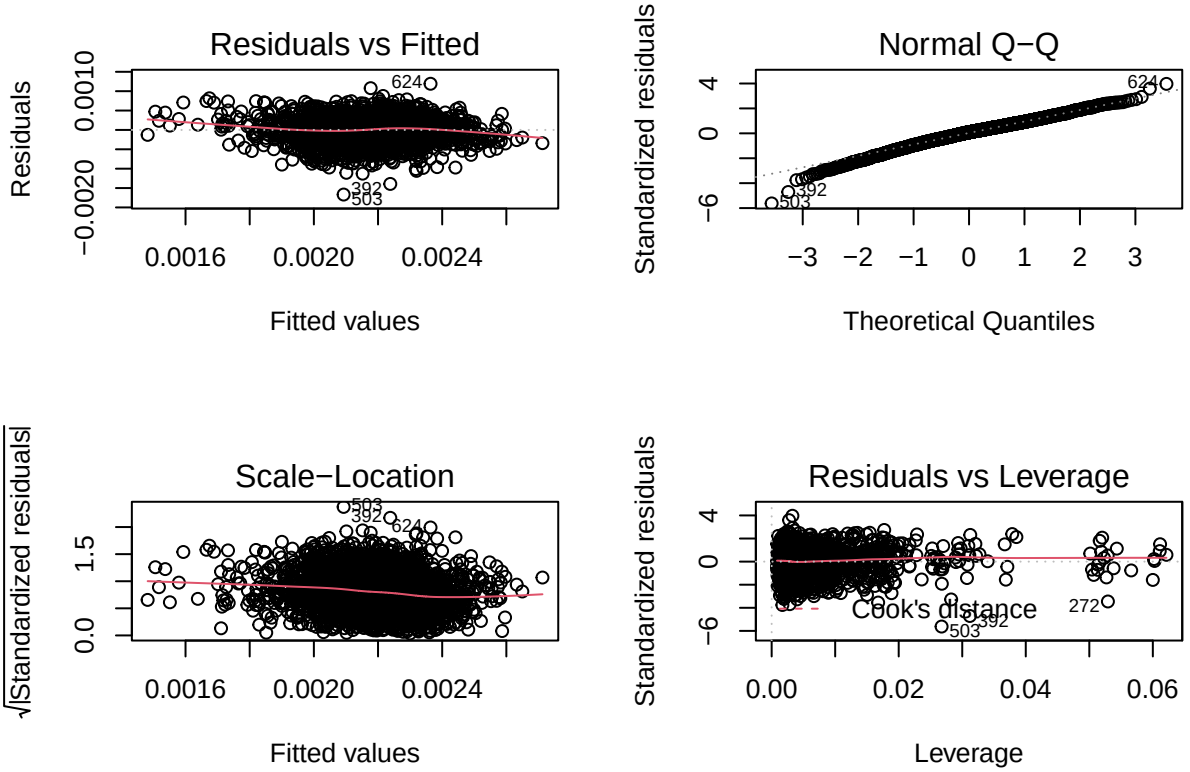
The initial model did meet the assumptions of a linear regression model. There exists constant variance, normality in the residual diagnostics, and independence in the way the data is structured.

5.2.3 Final multiple linear regression model summary

```
##
## Call:
## lm(formula = second_lm_model_pred, data = data)
##
## Residuals:
##        Min          1Q      Median          3Q         Max
## -1.668e-03 -1.774e-04  2.615e-05  1.948e-04  1.191e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.771e-03  1.723e-04  16.084  < 2e-16 ***
## naccmmse     2.122e-05  1.539e-06  13.788  < 2e-16 ***
```

```
## disnsev1    -5.211e-05  2.685e-05  -1.941   0.0524 .
## disnsev2    -2.079e-05  3.479e-05  -0.597   0.5502
## disnsev3    -6.926e-05  6.609e-05  -1.048   0.2947
## anxsev1     -6.135e-05  1.931e-05  -3.178   0.0015 **
## anxsev2     -4.308e-05  2.501e-05  -1.723   0.0850 .
## anxsev3     -8.826e-06  4.729e-05  -0.187   0.8520
## bpsys        4.530e-07  3.303e-07   1.371   0.1704
## age         -7.398e-06  5.565e-07 -13.292  < 2e-16 ***
## educ        -7.240e-06  1.855e-06  -3.902 9.77e-05 ***
## female1      2.856e-05  1.666e-05   1.715   0.0865 .
## height      -1.115e-05  2.335e-06  -4.773 1.91e-06 ***
## weight       9.453e-07  1.990e-07   4.751 2.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0003005 on 2686 degrees of freedom
## Multiple R-squared:  0.1951, Adjusted R-squared:  0.1912
## F-statistic: 50.09 on 13 and 2686 DF,  p-value: < 2.2e-16
```

5.2.4 Final multiple linear regression residual diagnostics



5.2.5 Leave One Out CV Code

```
########################
# Cross validation time
########################
n <- dim(data)[1]
p <- dim(data)[2]

###
# Leave one out
###

# Observation that is left out for LOOCV
index.mat = matrix(sample(1:n, n),nrow=n) # Just a vector of numbers
SSR.CV_mod1 = 0
SSR.CV_mod2 = 0
```

```
for(k in 1:n){
  train_data = data[-index.mat[k,],]
  test_data = data[index.mat[k,],]

  # Model 1
  model_lm_1 <- lm(sat_model_pred, data=train_data)
  y.hat = predict(model_lm_1,newdata=test_data)
  SSR.CV_mod1 = SSR.CV_mod1 + sum((test_data$response - y.hat)^2)

  # Model 2
  model_lm_2 <- lm(second_lm_model_pred, data=train_data)
  y.hat = predict(model_lm_2,newdata=test_data)
  SSR.CV_mod2 = SSR.CV_mod2 + sum((test_data$response - y.hat)^2)
}

cat('SSR LOOCV Mod 1: ', SSR.CV_mod1, '\n',
    'SSR LOOCV Mod 2: ', SSR.CV_mod2, '\n')
```

5.2.6 Single Fold CV

```
###
# Single Fold CV
###

# Straight from lecture with some modifications

# Gather row indices to decide which 80/20 train test split
a=seq(1,n,1)
b=sample(a,n *.8 ,replace = F)

# Split dataset into train and test based on selected indices
train_data<- data[b, ]
test_data<- data[-b,]

# train models
# Model 1
```

```
model_lm_1 <- lm(sat_model_pred, data=train_data)

# Model 2
model_lm_2 <- lm(second_lm_model_pred, data=train_data)

actual <- test_data$response

ssr_cv_mod1 <- sum((actual - predict( data  = test_data, model_lm_1))^2)
ssr_cv_mod2 <- sum((actual - predict( data  = test_data, model_lm_2))^2)

cat("model 1 CV SSR", ssr_cv_mod1, '\n',
    "model 2 CV SSR", ssr_cv_mod2)
```

5.2.7 5 Fold CV Code

```
###
# 5 fold CV
###

# Rows that will be split for 5 fold CV
index.mat = matrix(sample(1:n, n),nrow=5)
SSR.CV_mod1 = 0
SSR.CV_mod2 = 0

for(k in 1:5){
  train_data = data[-index.mat[k,],]
  test_data = data[index.mat[k,],]

  # Model 1
  model_lm_1 <- lm(sat_model_pred, data=train_data)
  y.hat = predict(model_lm_1,newdata=test_data)
  SSR.CV_mod1 = SSR.CV_mod1 + sum((test_data$response - y.hat)^2)

  # Model 2

  model_lm_2 <- lm(second_lm_model_pred, data= train_data)
```

```
  y.hat = predict(model_lm_2,newdata=test_data)
  SSR.CV_mod2 = SSR.CV_mod2 + sum((test_data$response - y.hat)^2)

}

cat('SSR 5 fold CV Mod 1: ', SSR.CV_mod1, '\n',
    'SSR 5 fold CV Mod 2: ', SSR.CV_mod2)
```

5.2.8 10 Fold CV Code

```
# 10 fold CV

# Rows that will be split for 10 fold CV
index.mat = matrix(sample(1:n, n),nrow=10)
SSR.CV_mod1 = 0
SSR.CV_mod2 = 0

# Initialize an empty data frame to store the combined results
combined_resid <- data.frame(Test_Data = numeric(0), Predicted = numeric(0))

for(k in 1:10){
  train_data = data[-index.mat[k,],]
  test_data = data[index.mat[k,],]

  # Model 1
  model_lm_1 <- lm(sat_model_pred, data=train_data)
  y.hat = predict(model_lm_1,newdata=test_data)
  SSR.CV_mod1 = SSR.CV_mod1 + sum((test_data$response - y.hat)^2)


  # Model 2
  model_lm_2 <- lm(second_lm_model_pred, data= train_data)
  y.hat = predict(model_lm_2,newdata=test_data)
  SSR.CV_mod2 = SSR.CV_mod2 + sum((test_data$response - y.hat)^2)

  # Keep track of predicted vs actual for data visualization in the next step
```

```
  residual_track_i <- data.frame(Test_Data = test_data$response, Predicted = y.hat)
  combined_resid <- rbind(combined_resid, residual_track_i)
}

cat('SSR 10 fold CV Mod 1: ', SSR.CV_mod1, '\n',
    'SSR 10 fold CV Mod 2: ', SSR.CV_mod2)
```

5.3 Logistic Regresion Model

5.3.1 Saturated Model Readout

```
##
## Call:
## glm(formula = log_reg_pred_1, family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5077  -0.5872  -0.3365   0.4306   2.6422
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.067e+01  1.885e+00   5.662 1.50e-08 ***
## naccmmse    -6.486e-01  3.276e-02 -19.798  < 2e-16 ***
## motsev1      8.762e-01  4.073e-01   2.151 0.031458 *
## motsev2      6.375e-01  5.712e-01   1.116 0.264341
## motsev3      1.381e+01  3.875e+02   0.036 0.971574
## disnsev1     1.332e+00  2.977e-01   4.473 7.70e-06 ***
## disnsev2     7.434e-01  3.821e-01   1.945 0.051724 .
## disnsev3     6.263e-02  7.783e-01   0.080 0.935860
## anxsev1      8.013e-01  1.886e-01   4.249 2.15e-05 ***
## anxsev2      1.209e+00  2.736e-01   4.418 9.97e-06 ***
## anxsev3      9.468e-01  5.627e-01   1.682 0.092478 .
## naccgds      1.661e-01  2.626e-02   6.327 2.50e-10 ***
## bpsys       -8.227e-04  3.692e-03  -0.223 0.823677
## bpdias       2.029e-03  6.678e-03   0.304 0.761223
## hrate        1.361e-03  5.386e-03   0.253 0.800483
## age          3.374e-02  5.751e-03   5.867 4.45e-09 ***
```

```
## educ         8.097e-02  1.929e-02   4.197 2.70e-05 ***
## female1     -5.400e-01  1.599e-01  -3.377 0.000734 ***
## height       6.940e-02  2.249e-02   3.086 0.002030 **
## weight      -1.005e-02  2.018e-03  -4.983 6.26e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3692.7  on 2699  degrees of freedom
## Residual deviance: 2073.1  on 2680  degrees of freedom
## AIC: 2113.1
##
## Number of Fisher Scoring iterations: 15
```

5.3.2 Final Model Summary

This summary is also presented in the main report.

```
# Finallogistic regression model
diag.logistic_final <- glm( log_reg_pred_2, data = data, family=binomial)
summary(diag.logistic_final)
```

```
##
## Call:
## glm(formula = log_reg_pred_2, family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4898  -0.5079  -0.3046   0.2662   2.5881
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  9.4539312  1.5875981   5.955 2.60e-09 ***
## naccicv     -0.0018751  0.0006893  -2.720  0.00653 **
## csfvol       0.0101446  0.0015439   6.571 5.00e-11 ***
## naccmmse    -0.5513430  0.0353309 -15.605  < 2e-16 ***
## remdates1    1.9799632  0.1597222  12.396  < 2e-16 ***
```

```
## remdates2    2.7047843  0.2640912  10.242  < 2e-16 ***
## remdates3    3.9055306  0.7502706   5.205 1.93e-07 ***
## remdates8    0.6893111  0.8240388   0.837  0.40287
## naccgds      0.1440366  0.0271095   5.313 1.08e-07 ***
## age          0.0019801  0.0066343   0.298  0.76535
## educ         0.0551497  0.0210145   2.624  0.00868 **
## height       0.0627572  0.0213451   2.940  0.00328 **
## weight      -0.0092618  0.0020947  -4.422 9.80e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3692.7  on 2699  degrees of freedom
## Residual deviance: 1794.2  on 2687  degrees of freedom
## AIC: 1820.2
##
## Number of Fisher Scoring iterations: 7
```

5.3.3 5 Fold CV for Logistic Regression

```
###
# 5 fold CV
###

# Rows that will be split for 5 fold CV
index.mat = matrix(sample(1:n, n),nrow=5)

# Initialize empty dataframe to store predicted results
combined_resid_saturated <- data.frame(Test_Data = numeric(0),
                                       Predicted = numeric(0))
combined_resid_final <- data.frame(Test_Data = numeric(0),
                                   Predicted = numeric(0))

# Probability threshold to convert predicted probability to 0 or 1
probability_threshold <- 0.5
```

```r
for(k in 1:5){
  train_data = data[-index.mat[k,],]
  test_data = data[index.mat[k,],]

  # Model 1
  diag.logistic_sat <- glm(log_reg_pred_1, data = train_data, family=binomial)

  # Calculate probability pi
  probability_sat<-diag.logistic_sat %>% predict(test_data,type="response")

  # Set the threshold as 0.5 for positive diagnosis
  predicted.classes <- ifelse(probability_sat > probability_threshold, "1", "0")

  # Keep track of actual and predicted for each fold
  probability_track_i_sat <- data.frame(Test_Data = test_data$diag.binary,
                                        Predicted = predicted.classes)
  combined_resid_saturated <- rbind(combined_resid_saturated,
                                    probability_track_i_sat)


  # Model 2
  diag.logistic_final <- glm(log_reg_pred_2, data = train_data, family=binomial)

  # Calculate probability pi
  probability_final<-diag.logistic_final %>% predict(test_data,type="response")

  # Set the threshold as 0.5 for positive diagnosis
  predicted.classes <- ifelse(probability_final > probability_threshold,"1","0")

  # Keep track of actual and predicted for each fold
  probability_track_i_final <- data.frame(Test_Data = test_data$diag.binary,
                                          Predicted = predicted.classes)
  combined_resid_final <- rbind(combined_resid_final, probability_track_i_final)
}

dim(combined_resid_saturated)
dim(combined_resid_final)
```

```r
#Accuracy for Saturated Model
mean(combined_resid_saturated$Test_Data == combined_resid_saturated$Predicted )

# Accuracy for Final Model
mean(combined_resid_final$Test_Data == combined_resid_final$Predicted )
```

5.4 Multinomial Logistic Regression

5.4.1 Saturated Multinomial Logistic Regression Summary

```
## # weights:  63 (40 variable)
## initial  value 2966.253179
## iter  10 value 2039.701013
## iter  20 value 1976.569078
## iter  30 value 1654.263607
## iter  40 value 1612.553023
## iter  50 value 1611.906680
## iter  60 value 1611.894000
## final  value 1611.893972
## converged

## Call:
## multinom(formula = multi_nom_pred_1, data = data)
##
## Coefficients:
##   (Intercept)   naccmmse   motsev1    motsev2   motsev3 disnsev1  disnsev2
## 1     8.63542 -0.5510550 0.7689109 0.09695091 -2.834951 1.176969 0.6506133
## 2    13.27451 -0.9088873 1.1356219 1.38572069 12.912661 1.764744 0.9984093
##     disnsev3   anxsev1  anxsev2   anxsev3   naccgds        bpsys        bpdias
## 1 -0.6265713 0.7515588 1.141708 0.9624614 0.1710346  0.002592345 -0.0009382702
## 2  0.9294681 0.9638322 1.463436 1.0174209 0.1478832 -0.013190999  0.0143807700
##          hrate        age       educ    female1     height       weight
## 1 0.0001524503 0.03561590 0.06088363 -0.6101105 0.05373321 -0.009192375
## 2 0.0044282064 0.02799159 0.14548914 -0.3349335 0.11889986 -0.012555806
##
## Std. Errors:
##   (Intercept)   naccmmse   motsev1   motsev2      motsev3  disnsev1  disnsev2
## 1  0.03114406 0.03115432 0.4221991 0.6323433 2.874570e-08 0.3077253 0.3972197
## 2  0.01832560 0.03699090 0.4615980 0.6248067 5.945549e-06 0.3453548 0.4543785
##    disnsev3   anxsev1   anxsev2   anxsev3    naccgds       bpsys      bpdias
## 1 0.9157339 0.1963128 0.2839218 0.5794171 0.02669504 0.003759083 0.006845571
## 2 0.8589818 0.2386128 0.3267408 0.6619793 0.03366729 0.005074659 0.009474489
##         hrate         age       educ   female1     height      weight
## 1 0.005478750 0.005559714 0.01981511 0.1392404 0.01531116 0.002089300
```

```
## 2 0.007256233 0.007387813 0.02609864 0.1852291 0.01965072 0.002899427
##
## Residual Deviance: 3223.788
## AIC: 3303.788
```

5.4.2 5 Fold CV for Multinomial Logistic Regression

```
###
# 5 fold CV
###

# Rows that will be split for 5 fold CV
index.mat = matrix(sample(1:n, n),nrow=5)

# Initialize empty dataframe to store predicted results
combined_resid_saturated <- data.frame(Test_Data = numeric(0),
                                       Predicted = numeric(0))
combined_resid_final <- data.frame(Test_Data = numeric(0),
                                   Predicted = numeric(0))


for(k in 1:5){
  train_data = data[-index.mat[k,],]
  test_data = data[index.mat[k,],]

  # Model 1
  model.1 <- multinom(multi_nom_pred_1, data = train_data)

  # Calculate probability pi
  probability_sat<- as.data.frame(predict(model.1,newdata = test_data,type="probs"))

  # Identify group with the highest probability
  max_col <- max.col(probability_sat) -1 # Subtract by 1 since group starts at 0
  probability_sat$pred_group <- max_col

  # Keep track of actual and predicted for each fold
  probability_track_i_sat <- data.frame(Test_Data = test_data$diagnosis,
```

```
                                        Predicted = probability_sat$pred_group)
  combined_resid_saturated <- rbind(combined_resid_saturated,
                                    probability_track_i_sat)

  # Model 2
  model.2 <- multinom(multi_nom_pred_2, data = train_data)

  # Calculate probability pi
  probability_final <- as.data.frame(predict(model.2,newdata = test_data,type="probs"))

  # Identify group with the highest probability
  max_col <- max.col(probability_final) -1 # Subtract by 1 since group starts at 0
  probability_final$pred_group <- max_col

  # Keep track of actual and predicted for each fold
  probability_track_i_final <- data.frame(Test_Data = test_data$diagnosis,
                                        Predicted = probability_final$pred_group)
  combined_resid_final <- rbind(combined_resid_final,
                                    probability_track_i_final)

}


dim(combined_resid_saturated)
dim(combined_resid_final)

#Accuracy for Saturated Model
mean(combined_resid_saturated$Test_Data == combined_resid_saturated$Predicted )

# Accuracy for Final Model
mean(combined_resid_final$Test_Data == combined_resid_final$Predicted )
```

6 Summary of Chapter 10 Reading

6.1 Introduction

Chapter 10 of Murphy's "Probabilistic Machine Learning: An Introduction" delves deep into the realm of logistic regression, a cornerstone in the world of statistical modeling and machine learning. This chapter introduces the foundational concepts, distinguishing between the binary and multinomial variants of the logistic regression algorithm. The text emphasizes its discriminative nature and goes into depth about how logistic regression predicts class probabilities based on input features. As the chapter progresses, Murphy expands on the intricacies of the logistic regression model, it's mathematical underpinnings, and most importantly the practical applications of these algorithms. This exploration offers a comprehensive insight into why logistic regression remains an indispensable tool for data scientists and statisticians alike in the realm of classifying binary and non-binary response variables.

The chapter starts by introducing this model in the context of classification tasks, where x is the input vector, y is the class label, and θ represents the model parameters. This method is usually applied in two situations:

1. Binary Logistic Regression (for two classes)
2. Multinomial Logistic Regression (for more than two classes)

An example of when binary logistic regression could be applied would be to predict whether or not will develop heart disease or not based on dietary factors and excercise levels. Another example could be to see if a borrower will default or not based on their level of income and total debt. Multinomial logistic regression could be used in an example with non-binary classes such as predicting to see if a patient has normal cognition, mild cognitive impairment, or dementia as shown earlier in this report. Another example would wanting to understand the factors that influence admissions decisions at a university and we have three possible response variable choices such as accepted, waitlisted, and rejected.

6.2 Binary Logistic Regression

The binary version of logistic regression corresponds to the model below:

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \text{Ber}(y|\sigma(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b)) \tag{1}$$

The equation employs the sigmoid function σ, which helps map any input into a value between 0 and 1, making it suitable for probability estimation. The weights, bias, and parameters are represented by $\mathbf{w}$, b, and θ respectively. The model calculates the probability of the positive class as:

$$p(y = 1|\boldsymbol{x}, \boldsymbol{\theta}) = \sigma(a) = \frac{1}{1 + e^{-a}} \tag{2}$$

where a is the log-odds or logit written as $log(\frac{1}{1+e^{-a}})$. The sigmoid function is integral to logistic regression. It's an S-shaped curve that can take any real-valued number and transform it into a value between 0 and 1. The relationship between the sigmoid function and its inverse, the logit, is crucial for understanding the transformation of data in the model.

Binary logistic regression models can have either liner classifiers what will define a linear plane known as a decision boundary or non-linear classifiers. So in essence, if the data can be linearly separated, then you would have a good linear classifier. When working with a nonlinear classifier, the decision boundary will be quadratic in nature, but can potentially be transformed in a linear decision boundary when the features are transformed from $\vec{x} = (x_1, x_2)$ to $\phi(\vec{x} = (x_1^2, x_2^2)$. However, it is important to note that as when the model becomes more complex as the parameter space increases, the risk of over fitting will increase.

6.2.1 Estimating Parameters for Logistic Regression

Logistic regression offers an elegant blend of simplicity and interpretability. The weights in the model provide insights into the importance of each feature. Positive weights increase the log-odds of the response (and thus increase the probability), and negative weights decrease the log-odds of the response (decreasing the probability). There are various ways to estimate the parameters of a logistic regression model such as applying a gradient based optimization algorithm. The first is to find the gradient of the objective function, negative log like-lihood (NLL), that solves

$$\nabla NLL(\mathbf{w}) = \mathbf{g(w)} = \mathbf{0} \tag{3}$$

With gradient-based optimization, the goal is to minimize (sometimes maximize) the objective function. The weights are iteratively adjusted until the parameters converge. Gradient-based optimization algorithms can converge to a stationary point where $\mathbf{g(w)=0}$. To ensure the stationary point is the actual global optimum, the next goal in estimating the parameters would be to ensure

that the NLL is convex in nature, which would require to find the Hessian of the NLL to prove that it is semi-definite.

An interesting approach to estimating the the parameters in logistic regression is to apply the perceptron algorithm that essentially starts with random weights. As the algorithm runs, it iteratively updates the weights whenever the prediction from the model makes a mistake. An advantage of using the perceptron method is that one does not need to compute probabilities, which can be useful if the label space of what we are trying to predict is very large. A downside to this algorithm however is that convergence only occurs when the data is linearly separable.

A third approach to estimate the parameters would be to use a classic second order optimization method which is in the form of

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{H}_t^{-1} \mathbf{g}_t \tag{4}$$

Above, w would be the weight, η is the step size, and $\mathbf{H}$ is the Hessian.

6.3 Multinomial Logistic Regression

When dealing with more than two class labels, multinomial logistic regression comes into play. Unlike the binary version, which uses the sigmoid function, the multinomial variant employs the softmax function as shown below

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = Cat(y|softmax(\boldsymbol{W}\boldsymbol{x} + b)) \tag{5}$$

where x is the input vector, y is the class label, and softmax is a function that is used to model the probabilities of multiple classes for a given input by taking a vector of real-valued numbers and transforms them into a probability distribution over multiple classes.

6.4 Estimating Paramters for Multinomial Logistic Regression

Similar as with logistic regression, one can compute the maximum likelihood estimate by minimizing the NLL again. In order to do this, one would need to solve again the gradient written in formula 3 above and one would need to make sure that the objective function is convex which would require finding the Hessian and applying a second-order optimization method. One down-

side to this is the calculation of the Hessian can be computationally expensive, so it is common to approximate it by using a quasi-newton method such as the BFGS technique.

6.5 Heirarchical Classification

It is not uncommon that one may want to investigate a set of labels that are in a hierarchy or taxonomy of some sort. For example, one might have to build an image classifier that predicts first what kind of pet one has. From there, one would have to identify the breed of the dog. The recommended approach to this is to first create a model for every binary output label for every possible node in a tree. Before training the model, one will ensure that the inner most label of this tree is brought up to the parent level (hypernyms), which would be the process of label smearing. At the parent node, one could have mammals for example. Then one level deeper, one could have dog and cat. The inner most layer could have the breed of the dog or the breed of the cat. Label smearing would mean that the inner most labels of the breed of the cat or dog would be propagated up to ensure that the image identifies that because the image was labeled as a Siamese breed of cat, it will also be labeled as a cat.

One issue with label smearing in hierarchical classification is that one could predict different classes such as "golden retriever", "cat", and "bird" all with probability 1 because the model does not capture that some labels are mutually exclusive by default and additional constraint must be added to prevent this. The constraint to be implemented would be a mutual exclusion between all label nodes which would be see as siblings. Going back to the example above, it would essentially partition the mammal probability as the sum of the probability of the image being a dog or a cat given the data. This helps prevent inccorrect hierarchical classifications.

6.6 Robust Logistic Regression

As with any dataset and data modeling techniques, outliers in the data can cause severe skewing of parameter estimates. In a classification task, these would be labeling errors which would cause labeling noise. To help reduce the impact of these outliers, robust logistic regression is utilized.

The first method here would be to take a logistic regression model and modify the likelihood so that each label y is generated uniformly at random with a probability π. Otherwise the probability is generated using the conditional model which in the binary scenario could be written as

$$p(y|\boldsymbol{x}) = \pi Ber(y|0.5) + (1-\pi)Ber(y|\sigma(\mathbf{w}^T\mathbf{x}) \tag{5}$$

Another method would be to approach robust logistic regression through bi-tempered loss. An observation that was far from the decision boundary but mislabeled would have an an adverse affect on the model if the loss function is convex. This would be fixed by replacing the cross entropy function with a "tempered" version which ultimately places a boundary to help ensure the loss of outliers is reduced.

6.7 Bayesian Logistic Regresion

Lastly, there is a Bayesian aspect of logistic regression that can be utilized to help capture uncertainty to a point estimate of the parameters provided by previous discussed techniques like the MLE. This uncertainty is captured by computing the posterior. Calculating this directly is not possible for a logistic regression model so an approximation must be used, such as the Laplace approximation. It is through this process where the posterior is approximated using a Gaussian. The posterior will tell one everything about the parameters of the model given the provided data. With machine learning, the goal is often to predict a response variable y given an input of a vector of x, which is where the posterior predictive comes into play as written below:

$$p(y|\mathbf{x}, D) = \int P(y|\mathbf{x}, \mathbf{w})p(\mathbf{w}|D)d\mathbf{w} \tag{6}$$

This can be complicated to compute at times, so it can be approximated using either the Monte Carlo method. With Monte Carlo, it's the simplest approach to approximate the integral found in (6). One draws S samples from the posterior $\mathbf{w}_s \sim p(\mathbf{w}|D)$ and then computes

$$p(y = 1|\mathbf{x}, D) \approx \frac{1}{S}\sum_{s=1}^{S} \sigma(\mathbf{w}_s^T \mathbf{x}) \tag{7}$$

This approach can be slow because one needs to draw S samples for each input $\mathbf{x}$.

Comparisons with Other Models

Compared to other machine learning models, logistic regression is less flexible but has the advantage of being more interpretable. For instance, while deep neural networks might achieve higher accuracy on certain tasks, they come at the cost of being black-box models. While logistic regression is linear, other models like decision trees, random forests, and neural networks can capture non-linear relationships. These models, however, might lack the interpretability that logistic regression offers. Thus, the choice of model often hinges on the specific requirements of a task.

Logistic regression has seen various extensions since its inception, such as ridge and lasso regularization, which help in feature selection and preventing overfitting. Additionally, techniques like polynomial regression can introduce non-linearity into the model.

Chapter 10 provided a comprehensive introduction to logistic regression, one of the foundational models in machine learning. While its simplicity is both a strength and a limitation, it remains a vital tool in the data scientist's toolkit, especially when interpretability and computational efficiency are paramount.