SPARSE REGRESSION OF TEXTUAL ANALYSIS

by

PHYLISICIA CARTER

BRENDAN AMES, COMMITTEE CHAIR
YUHUI CHEN
CALI DAVIS
HYUN-KYOUNG KWON
WEI ZHU

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2018

ABSTRACT

We consider sparse regression techniques as tools for classification of sentiment within Twitter posts. Analysis of Twitter usage suffers from several unique challenges. For example, the 140-character limit severely limits the amount of information contained in each post; this causes most tweets to contain an extremely small subset of the dictionary, presenting challenges for learning schemes based on dictionary usage. To remedy this undersampling issue, we propose usage of penalized regression. Here, we employ logistic regularization to avoid any degeneracy caused by the sparse usage of the dictionary in each tweet, while simultaneously learning which terms are most associated with each sentiment. Accelerated sparse discriminant analysis is also used to combat the issues of degeneracy and overfitting of the training data while providing dimension reduction. As illustrative examples, we employ sparse logistic regression to classify tweets based on the users' perception of a connection between vaccination and autism, and we examine the Twitter users' sentiment of the use of autonomous cars.

DEDICATION

I would like to dedicate this dissertation to my father, my mother, and my brother, Godfrey Sr., Phyllis, and Godfrey Jr., who have always supported and encouraged me every step of the way. Thank you for believing in me! I also dedicate this dissertation to all of my family and friends, who have continued to support me throughout my academic journey.

ACKNOWLEDGMENTS

It is with the utmost appreciation that I would like to thank the University of Alabama, the Department of Mathematics, and the Southern Regional Education Board. I would also like to acknowledge my dissertation committee, especially Dr. Brendan Ames, the committee chair.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Analysis of textual data has become an increasing popular area of research due to its usefulness in a variety of areas such as artificial intelligence, security, and marketing. Due to its worth in many industries, it is imperative that the results obtained be accurately classified and interpreted. James et al. [11, Chapter 4] refers to classification as "approaches for predicting qualitative responses". In this research, the aim is to accurately classify textual data based on the sentiments within the text. There has been research that has focused on classifying texts based on its subject matter, but there has been few works done that focus on accurately classifying the sentiment of text within the same subject matter [17, Section 1]. This textual analysis will be conducted through the use of linear regression, logistic regression, and sparse regression techniques. Some statistical approaches to classification use estimates of the probability of the testing data belonging to one category over the other category as methods of classification. In other words, the probability that an observation belongs to a particular qualitative response is predicted. Based on this probability, the observation is classified into one of the qualitative response categories. The separation of these categories is known as the decision boundary. The measure of accuracy in the prediction of such classifications can be computed by taking the error rate, defined to be the proportion of misclassified observations within the data. In this dissertation, we consider the application of several statistical methods of classification, including logistic regression, K-nearest neighbors, and linear discriminant analysis.

1.1 Logistic Regression

Logistic regression can be used to estimate the probability that the response variable can be be classified as one of two categories based on multiple predictors [11, Section 4.3]. This method of regression generally produces linear decision boundaries and is performed using the maximum likelihood to estimate the model coefficients, β_i. Since logistic regression utilizes the maximum likelihood estimate the model coefficients, the data does not have to follow Gaussian assumptions and can still perform well [11, Section 4.3]. Logistic regression will be discussed in more detail in Chapter 3.

1.2 K-Nearest Neighbors

K-nearest neighbors is an approach for classification that can be used when the parameters of the data do not follow any assumptions, and can be very dominant when the decision boundary is non-linear. The K-nearest neighbors approach (KNN) takes a value positive integer K and a test observation, x_0, and identifies K points in the training data that are nearest to x_0. Once the K points are identified, the conditional probability that the test observation falls into the category of one of these points is calculated. Bayes rule is then applied and the test observation is classified to the class with the largest probability. In order to acquire the smallest misclassification error, or error rate, for KNN, the choice of the integer K must be correctly chosen. Having a K that is too small can result in overfitting of the decision boundary, and having a K that is too large can causes the decision boundary to be less flexible and nearly linear. KNN also does not provide us with information as to which predictors are important to our model.

1.3 Linear Discriminant Analysis

Linear discriminant analysis (LDA) models the distribution of each predictor in each of the response classes and uses Bayes' Theorem to get estimates for the probability that each predictor belongs to a certain class or category. When the classes are well-separated, LDA is usually the method of choice due to the instability of logistic regression with well-separated classes. In order to use LDA, the data is assumed to have observations that have a Gaussian distribution with a common covariance matrix in each class [11, Section 4.5]. If the number of observations, n, is relatively small and the distribution of the predictors is approximately normal in each class, then the LDA model is more stable than the logistic regression model. Also, LDA is more popular when trying to classify data into more than 2 categories. The overall goal of LDA is to produce the smallest misclassification rate, regardless of the misclassification rate among the response classes. However, this misclassification rate for each response can be improved by evaluating the threshold of the model. Obtaining the best threshold will lower the misclassification rate of each response class, but it will also increase the overall misclassification rate of the model. LDA will be discussed in greater detail in Chapter 5.

1.4 Summary

In this paper, the methods of linear regression, logistic regression, linear discriminant analysis, sparse logistic regression, and sparse discriminant analysis will be discussed. A brief overview of each technique will be outline followed by results obtained from various data sources. With this research, sparse regression techniques will be used to analyze social media posts. In particular, 140-character limit Twitter posts will be analyzed using sparse versions of logistic regression and linear discriminant analysis.

The analysis of these Twitter posts will be analyzed to see which methods result in a lower misclassification rate and a balance among errors within in response class.

LINEAR REGRESSION

In this research, the data consists of n observations and p predictors or features. A useful tool in statistics for predicting quantitative responses is linear regression. Linear regression allows one to predict a quantitative response, Y_i, based on certain predictor variables, X_{ij}. Linear regression is conducted under the assumption that there is a linear, and possibly noisy, relationship between the predictors and the observations. In other words, the response Y_i is assumed to depend linearly on the predictor variables x_{ij}. Mathematically, this can be written as:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} + \epsilon_i,$$

where, $\beta_0, \beta_1, ..., \beta_p$ are unknown parameters to the training model. The first term, β_0, represents the intercept term of the model or the value of Y_i when all X_{ij}'s are zero, β_j is the average effect on Y_i of one unit increase in X_{ij} while holding all of the other predictors constant, X_{ij} represents the interaction between the i^{th} observation and the j^{th} predictor, and ϵ_i is the error term associated with the i^{th} term of the model since the true relationship between X and Y may not be linear. As previously stated, the data used is comprised of p predictor variables and n observations; therefore, X is a matrix of $n \times (p+1)$ dimensions with the rows being equal to the n observations and the columns being the p predictors plus a column of all ones to correspond to the intercept coefficient β_0. The response variable, Y, is an n x 1 vector, and b, which will be a vector of the calculated model parameters $\hat{\beta}_i$ is a $(p+1) \times 1$ vector.

The goal of linear regression is to estimate the coefficients of β_i so that the predicted line or hyperplane is as close to the original data as possible. The training data

is used to calculate $\beta_0, ..., \beta_p$, which yields the following prediction equation:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \hat{\beta}_2 X_{i2} + ... + \hat{\beta}_p X_{ip}.$$

Using a least squares approach will choose the parameters β_i which minimizes the sum of squared residuals. Here the residual sum on squares (RSS) is defined as

$$RSS = e_1 + e_2 + ... + e_n = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{i1} - \hat{\beta}_2 X_{i2} - ... - \hat{\beta}_p X_{ip})^2.$$

The values of $\hat{\beta}_i$ that minimizes RSS is given by the following two equations, known as normal equations:

$$\sum_{i=1}^{n} Y_i = n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} X_{ip}$$

$$\sum_{i=1}^{n} X_i Y_i = n\hat{\beta}_0 \sum_{i=1}^{n} X_i + \hat{\beta}_1 \sum_{i=1}^{n} X_{ip}^2$$

For simplicity, matrix notation will be used to express the method of solving for β_i. The normal equations are expressed in the following manner using matrix notation:

$$X^T X b = X^T Y$$

Solving for b will result in the following:

$$b = (X^T X)^{-1} X^T Y$$

Once the model parameters are calculated, the accuracy of the model can be determined by calculating the residual standard error (RSE), which approximates how much the response will differ from the actual regression line. This error can be calculated

in the following manner:

$$RSE = \sqrt{\frac{1}{n-2}RSS} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}.$$

A small value for RSE denotes that the predicted model fits the data well. In other words, if all of the predicted Y_i observations were on the fitted regression line, then RSE would have a value of 0. Another method for calculating the accuracy of the model is the R^2 statistic. This approach is sometimes used instead of RSE because RSE is measured in units of Y, and can sometimes be misleading as to what denotes a good value of RSE. However, the R^2 statistic is a way to measure the proportion of variability in Y that can be explained by X, and is therefore independent of the scale of Y. This R^2 statistic is a proportion of the variance explained:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

with $TSS = \sum (y_i - \bar{y})^2$. The total sum of squares (TSS) measures the total variance in the response Y before the regression is performed. Therefore, $TSS - RSS$ is the amount of variability in the response Y that is explained by performing regression. This value, $TSS - RSS$ is the sum of the square of the difference between the predicted values of Y_i and the average fitted value of Y. Since the R^2 statistic is a proportion it has values between 0 and 1. A value closer to 1 denotes that a large proportion of variability in Y is explained by the regression, and a value closer to 0 indicates that a small proportion of variability in Y is explained by the regression. The challenge with determining a good value for the R^2 statistic is that a decent value depends on the application that is being modeled.

2.1 Possible Complications with Linear Regression

Linear regression has several possible issues which include the relationship between the predictor and the response variable not being linear, a correlation of error terms, non-constant variance in error terms, outliers, high leverage points, and collinearity.

When applying linear regression to a data set, it is assumed that there is a linear relationship between the response variable and its predictors. However, this is not always the case, and the absence of this linear relationship could lead to all conclusions that were made based on the linear regression model to be false. Non-linearity can be identified by using residual plots. In simple linear regression, residual plots are the residuals, $e_i = y_i - \hat{y}_i$ versus the predictor, x_i. In multiple regression models, the residual plot is the residuals versus the predicted values, $\hat{y}_i$. When observing these residual plots, one is looking to see if there is any type of pattern between the plotted values. The absence of a pattern in the residual plot may indicate that the model is linear, while the presence of a pattern may indicate that the model is not linear.

Error terms can also play a large role in causing issues with linear regression. In particular, an assumption made about linear regression is that the error terms, $\epsilon_1, \epsilon_2, ..., \epsilon_n$, are all uncorrelated. This is not always true. Having a correlation among the error terms can cause the estimated standard errors to underestimate the true standard errors. This leads to narrower confidence intervals, which can cause one to assume that the model is performing better than it actually is. It also needs to hold true that the error terms of the linear model have a constant variance. In other words, it is assumed that error terms of the model do not increase or decrease based on the value of the response. One can identify non-constant variances by noticing a funnel shape in the residual plot.

Another element that can cause problems for linear regression is outliers. An outlier is a point for which y_i is far from the value predicted by the model. The presence of such a point can cause the computation of the confidence intervals, p-values,

and R^2 values for the model to drastically change. To recognize outliers, it is best to plot the studentized residuals, which are computed by dividing each residual, e_i, by its estimated standard error. Possible outliers are generally those studentized residuals with an absolute value greater than 3.

Closely related to outliers are high leverage points, which are observation with an unusual value for x_i. In contrast to outliers, the predictor value, x_i for these observations are large relative to the other observations. The leverage statistic is used to measure an observation's leverage. If the value of this statistic is large, then that is an indication of an observation with high leverage. The leverage statistic can be computed as follows:

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n} (x_{i'} - \bar{x})^2}$$

This statistic always has a value between $\frac{1}{n}$ and 1, and the average leverage statistic for all observations is $\frac{p+1}{n}$. If h_i is greater than the average leverage, then it may be a high leverage point.

An additional potential problem in linear regression is collinearity. This is when there are two or more predictors that are related. Collinearity can make it difficult to distinguish between the effects of the collinear predictors on the response variable. It also causes the estimates of the model parameters to be inaccurate, which makes the standard error for the model coefficients grow, and subsequently affects the t-statistic, which may lead to inaccuracies with the hypothesis testing. Collinearity can be detected by observing a correlation matrix of the predictors. Highly correlated predictors are identified by an element in the matrix with a large absolute value. This approach does not work if there are three or more variables that are collinear. This is referred to as multicollinearity. Multicollinearity can be identified by computed the variance inflation factor (VIF). The VIF is the ratio of the variance of $\hat{\beta}_j$ when the full model is fitted divided by the variance of $\hat{\beta}_j$, if fit on its own. Computing the VIF and getting a value

of 1 indicates that there is an absence of collinearity; however, if the VIF value is greater than 5 or 10, then there may be collinearity present. The VIF value can be calculated using the following:

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{x_j|x_{-j}}}$$

with $R^2_{x_j|x_{-j}}$ being the R^2 statistic from a regression of x_j onto all of the other predictors.

2.2 Regularization of Linear Regression

The potential problems can make the linear model difficult to interpret. To aid with the interpretability of the model, one can implement regularization, which shrinks the coefficients estimates of the model towards zero. Such shrinking of the coefficient estimates can significantly reduce their variance. The regularization methods we consider are ridge regression, the least absolute shrinkage and selection operator (LASSO), and elastic net.

2.2.1 Ridge Regression

As stated earlier, the least square approach produces estimates of the model parameters, $\beta_0, \beta_1, ..., \beta_p$, by minimizing the following:

$$RSS = \sum_{i=1}^{n}(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ip})^2$$

For ridge regression, the coefficients are estimated by minimizing the following:

$$\sum_{i=1}^{n}(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ip})^2 + \lambda \sum_{j=1}^{p}\beta_j^2,$$

where $\lambda \geq 0$ is a tuning parameter. The goal of ridge regression is to find coefficient estimates that fit the data well while making RSS small. In order to do this, there is a shrinkage penalty introduced in ridge regression. The shrinkage penalty is small when $\beta_0, \beta_1, ..., \beta_p$ are close to zero, and this causes it to have the effect of shrinking

β_j towards zero. The tuning parameter λ provides a control between RSS and the estimates of β_j. Whenever $\lambda = 0$, there is no effect on the shrinking of β_j and thus the least squares estimates are produced. As $\lambda \to \infty$, the effect of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero. Because ridge regression produces a different set of coefficient estimates, $\hat{\beta}_j^R$, for each λ, cross-validation, a technique discussed later, will be used to select the best λ.

As the value of λ increases, the flexibility of the ridge regression fit decreases, which leads to a decrease in variance but an increase in bias. Generally, when the relationship between the response variable and the predictors are linear, the least squares method will produce estimates with low bias and high variance [11, Section 6.2.1]. In other words, a small change in the training data could lead to a large change in the least squares coefficient estimates. In general, when the number of predictors, p, is almost as large as the number of observations, n, then the estimated parameters by the least squares method will be extremely variable. On the other hand, if $p > n$, then the least squares approach does not produce a unique solution; however, ridge regression tends to perform well by exchanging a small increase in bias for a large decrease in variance. Therefore, ridge regression works better in scenarios where the least squares estimates have high variance.

2.2.2 LASSO

One main issue with ridge regression is that it always includes all p predictors in the final model instead of selecting the best subset of p predictors. This is due to the fact that the $\lambda \sum B_j^2$ penalty will shrink all of the coefficients toward zero, but none of the coefficients will actually equal zero, unless $\lambda = \infty$. This is only a problem when it comes to trying to interpret the model when p is quite large. However, LASSO combats this disadvantage. The LASSO coefficients, $\hat{\beta}_j^L$, minimize

$$\sum_{i=1}^{n}(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ip})^2 + \lambda \sum_{j=1}^{p}|\beta_j|.$$

LASSO uses an l_1-norm, which forces some of the coefficient estimates to be equal to zero when the tuning parameter is sufficiently large. Therefore, LASSO performs variable selection, which results in models that are easier to interpret. Due to this variable selection, LASSO yields sparse models because it results in models that have only a subset of variables. Choosing the best λ to yield the best subset of model parameters is often done by using cross-validation [11, Section 6.2.2].

2.2.3 Cross-Validation

Cross-validation, in particular, k-fold cross-validation is the method in which the set of observations are randomly divided into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the model is fit on the remaining $k-1$ folds. The mean squared error (MSE) is then calculated on the observations of the held-out fold. This technique is repeated k times with a different group of observations serving as the validation set each time. The k-fold cross-validation estimate can be computed by averaging these values:

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i.$$

The ultimate goal of cross-validation for this research is to aid in selecting the appropriate tuning parameter for ridge regression, LASSO, and elastic net, which will be defined in the following section. In order to produce the best model with these regularization methods, one selects the tuning parameter, λ, that results in the smallest MSE [11, Section 5.1.3].

2.2.4 Elastic Net Penalty

The elastic net penalty, as described by Hastie in "Statistical Learning with Sparsity" [10, Section 4.2], is the combination of a squared l_2-penalty and the l_1-penalty. This method deals well with correlated groups of variables and has the tendency to select (or

not select) the correlated predictors together. The elastic net solves the convex problem

$$\min_{(\beta_0,\beta)\in\mathbb{R}\times\mathbb{R}^p} \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0 x_i^T\beta)^2 + \lambda\left[\frac{1}{2}(1-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1\right]$$

where $\alpha \in [0,1]$ is a parameter that can be varied. The penalty applied to an individual coefficient with no regard to the regularization weight $\lambda > 0$ is given by

$$\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|$$

If $\alpha = 1$, then the penalty becomes the l_1-norm or LASSO penalty, and if $\alpha = 0$, then the problem becomes the squared l_2-norm or the ridge penalty. By adding some component of the ridge penalty to the l_1-penalty, the elastic net provides controls for strong within-class correlations. For $\alpha < 1$ and $\lambda > 0$, the elastic net becomes strictly convex, and therefore, a unique solution exists regardless of the correlations in the X_j. Due to the elastic net problem being convex in the pair $(\beta_0, \beta) \in \mathbb{R}x\mathbb{R}^p$, there are several algorithms that can be used to solve the problem. Coordinate descent is an effective method with updates that are extensions of those for the LASSO problem. An unpenalized intercept is included in the model, and it can be dispensed with at the onset. The covariates x_{ij} are centered, and the optimal intercept becomes $\hat{\beta}_0 = \bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$. Once the optimal intercept, $\hat{\beta}_0$, is computed the optimal vector $\hat{\beta} = (\hat{\beta}_1, ..., \hat{\beta}_p)$ needs to be calculated. The coordinate descent update for the j^{th} coefficient has the form

$$\hat{\beta}_j = \frac{S_{\lambda\alpha}(\sum_{i=1}^{n} r_{ij}x_{ij})}{\sum_{i=1}^{n} x_{ij}^2 + \lambda(1-\alpha)}$$

with $S_\mu(z) := \text{sign}(z)(z-\mu)_+$ being the soft-thresholding operator, and $r_{ij} := y_i - \hat{\beta}_0 - \sum_{k\neq j} x_{ij}\hat{\beta}_k$ being the partial residual. The updates are continued until convergence is reached.

LOGISTIC REGRESSION

Logistic regression is a common technique used to classify data with a binary outcome, usually of 0 or 1, by modeling the probability that Y_i, the response variable, belongs to one of two categories based on the predictor or predictors, X, an $n \times (p+1)$ matrix as before. For simplicity, X is a matrix of all p predictors and β is the column vector for all model parameters in the logistic function. This logistic function is used to carry out logistic regression and is defined as the following:

$$p(X) = \frac{e^{\beta_0 + X_1\beta_1 + ... + X_p\beta_p}}{1 + e^{\beta_0 + X_1\beta_1 + ... + X_p\beta_p}} = \frac{e^{X\beta}}{1 + e^{X\beta}}.$$

In this function, $p(X)$ is the probability that a certain observation is in the specified category of Y, and the βs represent the model parameters, as in linear regression.. Because the numerator of the logistic function is an exponential function, it will always have a non-negative value, and the denominator is just the numerator plus one, so it will always be a positive number. Therefore, the logistic function will always have a value between 0 and 1 and a sigmoidal shape. After rearranging the logistic function and taking the *log* of both sides, an equation with the left-hand side known as the log-odds or logit equation is obtained:

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

This function is known as the log-odds or logit.

The first steps to performing logistic regression is to estimate the regression coefficients β. These unknown coefficients are estimated using the training data and the

maximum likelihood. In short, maximum likelihood will allow β to be calculated and put into the logistic function, which will yield a number close to 0 or 1. The estimates of β that maximize the likelihood function will be chosen. The likelihood function is defined as follows:

$$l(\beta) = \prod_{i=1}^{n} p(X_i)^{Y_i}(1 - p(X_i))^{1-Y_i}$$

The *log-likelihood* of this function is detailed below:

$$l(\beta) = \sum_{i=1}^{n} [Y_i \log(p(X_i) + (1 - Y_i) \log(1 - p(X_i)]$$

$$l(\beta) = \sum_{i=1}^{n} [Y_i\beta - \log(1 + e^{X_i\beta})].$$

Once the likelihood function is maximized and the coefficients are estimated, then the estimates are input into the logistic function, and the probability that the predictor belongs to the response variable Y is calculated. Maximizing this log-likelihood function is the same as minimizing the following function:

$$\min -\frac{1}{n} \sum_{i=1}^{n} [Y_i\beta - \log(1 + e^{X_i\beta})].$$

The default cutoff value for assigning predicts for the logistic function is 0.5. In other words, if $p(X_i) < 0.5$, then the predictor will be classified in the response class labeled 0, and if $p(X_i) \geq 0.5$, then the predictor will be said to belong to the response class labeled as 1. However, since the data may not be perfectly balanced, the threshold or cutoff value may need to be adjusted. For instance, most of the predictors that belong to the response class classified as 0 may have a probability of 0.3 or less. Hence the threshold would need to be adjusted to get the most accurate logistic model of the data.

Another way to improve the logistic model is to apply regularization techniques

similar to those used for linear regression. In order for this logistic model to yield sparse results and to increase interpretability of the model, the l_1 penalty ($\|l_1\|$) is implemented in the following manner:

$$\min -\frac{1}{n}\sum_{i=1}^{n}[Y_i\beta - \log(1 + e^{X_i\beta})] + \lambda\|\beta\|_1.$$

GLMNET

Glmnet is a package in R that uses the penalized maximum likelihood to fit a generalized linear model [7]. Glmnet is a coordinate descent method for fitting generalized linear models (GLM) with elastic-net (NET) regularization. This package was used to analyze the data in this research. The regularization path from the glmnet package is computed for the LASSO and elastic net penalty. This algorithm yields sparse data, which aids with interpreting the model, and is extremely fast. This package can fit several models that include linear, logistic, multinomial, poisson, and Cox regression models. The following technique is from the glmnet vignette by Hastie and Qian [8]. Glmnet solves the following negative log-likelihood problem

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} -\left[\frac{1}{n}\sum_{i=1}^{n} y_i(\beta_0 + x_i^T\beta) - \log(1+e^{\beta_0+x_i^T\beta})\right] + \lambda\left[\frac{1}{2}(1-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1\right]$$

over a grid of values of λ. The α controls the elastic net penalty in the above problem. If $\alpha = 1$, then the LASSO penalty is performed, and if $\alpha = 0$, then ridge regression is performed. The overall strength of the penalty is controlled by the tuning parameter λ.

When performing ridge regression, the ridge penalty will shrink the coefficients of correlated predictors towards each other. LASSO, on the other hand, selects one of these predictors and discards the others. The elastic net penalty mixes these two approaches. For instance, assume that there are correlated predictors in a group, with $\alpha = 0.5$, the elastic net will select the groups in or out together.

Cyclical coordinate descent is used within the glmnet algorithm. This technique optimizes the objective function over each model parameter while keeping the other parameters fixed, and is repeated until convergence is reached. As illustrated in [7,

Section 3], this procedure is done in the following manner. Consider the log-likelihood of the logistic regression function:

$$l(\beta) = \sum_{i=1}^{n} [Y_i \beta - \log(1 + e^{X_i \beta})],$$

which is a concave function of parameters. Forming a quadratic approximation to the log-likelihood, which is a Taylor expansion about the current estimates $\tilde{\beta}_0, \tilde{\beta}$, yields the following:

$$l_Q(\beta_0, \beta) = -\frac{1}{2n} \sum_{i=1}^{n} w_i (z_i - \beta_0 - X_i^T \beta)^2 + C(\tilde{\beta}_0, \tilde{\beta})^2,$$

with $z_i = \tilde{\beta}_0 + X_i^T \tilde{\beta} + \frac{Y_i - \tilde{p}(X_i}{\tilde{p}(X_i)(1-\tilde{p}(X_i))}$, $w_i = \tilde{p}(X_i)(1 - \tilde{p}(X_i))$, and $\tilde{p}(X_i)$ is calculated at the current parameters. The Newton update is calculated by minimizing l_Q. In similar fashion, for each value of λ, an outer loop which computes the quadratic approximation l_Q about the current parameter $(\tilde{\beta}_0, \tilde{\beta})$ is created. Next, the coordinate descent method is used to solved the penalized weighted least-squares problem:

$$\min_{(\beta_0, \beta) \in \mathbb{R}^{p+1}} \left\{ - l_Q(\beta_0, \beta) + \lambda P_\alpha(\beta) \right\}.$$

This produces the following nested loops: As λ gets smaller, the quadratic approximation l_Q is updated using the current parameters $(\tilde{\beta}_0, \tilde{\beta})$, and the coordinate descent algorithm on the penalized weighted least-squares problem is being run. It should be noted that when $p >> n$, λ cannot be run all the way to zero since the saturated logistic regression fit is undefined because the parameters must go towards $\pm\infty$ in order to yield probabilities of 0 or 1. Therefore, the default λ sequence goes to $\lambda_{\min} = \epsilon \lambda_{\max} > 0$. Also, the Newton algorithm does not always converge without step-size optimization, and this code does not implement any checks for divergence because it would slow down the process. The quadratic approximations are very accurate because there is a closed form expression for starting solutions, and each subsequent solution is warm-stated from the previous

close-by solution.

Cross-validation with the glmnet package is executed using the cv.glmnet function. Utilizing this command yields the plot of the cross-validation curve, which displays upper and lower standard deviation curves along the λ sequence. Two selected lambdas are displayed by two vertical dotted lines. The two lambdas shown are lambda.min, which is the value of λ that produces the minimum mean cross-validated error, and lambda.1se, which gives the most regularized model such that the error is within one standard error of the minimum. Once cross-validation is done on the training data, predictions can be made on the testing data with the λ that best suits one's preference.

LINEAR DISCRIMINANT ANALYSIS

Whenever one is attempting to classify data into more than one class, the method of choice is usually Linear Discriminant Analysis (LDA). LDA can be considered using three different approaches: Bayes' Theorem, Fisher's Discriminant, and optimal scoring. Throughout this chapter, these three methods will be explored along with the regularization of each method.

5.1 Bayes' Theorem Method for Linear Discriminant Analysis

This section is derived from the Bayes' Theorem approach for LDA in "An Introduction to Statistical Learning with Applications in R" [11, Section 4.4]. LDA using the Bayes' Theorem method models the distribution of the predictors X separately in each of the response classes and then uses Bayes' Theorem to turn these into estimates for $P(Y = k|X = x)$. Bayes' theorem states

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}.$$

Letting $p_k(X) = P(Y = k|X = x)$ be the posterior probability that an observation $X = x$ belongs to the k^{th} class, then one can input π_k and $f_k(X)$ into Bayes' theorem. The overall prior probability that a randomly chosen observation is originally in the k^{th} class is denoted as π_k, and is calculated by computing the fraction of the training observations that belong to the k^{th} class.

The number of predictors will tell how $f_k(x)$ will be calculated. If there is only one predictor, then $f_k(x)$ will be assumed to be Gaussian and have a normal density

function of

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2}.$$

The mean and variance parameters for the k^{th} class are represented by μ_k and σ_k^2, respectively. These estimates can be calculated by taking the average of all the training observations from the k^{th} class and by taking the weighted average of the sample variances for each of the K classes. Once $\hat{\mu}_k$ and $\hat{\sigma}^2$ has been estimated, then these values along with $\hat{\pi_k}$ are plugged into the LDA classifier. The LDA classifier is of the following form:

$$\hat{\delta}_k(x) = x\frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{2\mu}_k}{\hat{\sigma}^2} + \log \hat{\pi}_k.$$

The main goal of the LDA classifier is to assign an observation $X = x$ to the class in which $\hat{\delta_k}(x)$ has the highest value. If there is more than one predictor, then the predictors can be denoted as $X = (X_1, X_2, ..., X_p)$, and the predictors are assumed to have a multivariate Gaussian distribution with a class-specific mean vector and a common covariance matrix. The multivariate Gaussian density is defined as:

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)).$$

Just as with one predictor, the values of $\mu_1, ..., \mu_k$, $\pi_1, ..., \pi_k$, and Σ are plugged into the LDA classifier, and the observation $X = x$ is assigned to the class for which $\hat{\delta_k}(x)$ is the largest. The LDA classifier is described as the follows:

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log(\pi_k).$$

5.2 Fisher's Method of Linear Discriminant Analysis

Per the procedure from [9, Chapter 4], Fisher's method of LDA focuses on finding the linear combination of predictors so that the ratio of the between-class variance and the within-class variance is maximized. This would ensure that there would be maximum separation between the centers of the data while simultaneously minimizing the variation within each class of data linearly projected to a space spanned by the discriminant vectors. Mathematically, this is represented by letting B denote the between-class covariance matrix and W denoting the within-class covariance matrix. The between-class covariance matrix B measures the distance between the classes, and the within-class covariance matrix measures how far away the centers of the classes are from one another. Fisher's approach is to find the value of β which maximizes the following:

$$\frac{\beta^T B \beta}{\beta^T W \beta}.$$

This problem is solved by computing the eigenvector that corresponds to the largest eigenvalue of $W^{-1}B$. This vector will be a linear discriminant β_1. The next direction β_2, orthogonal in W to β_1 is computed by solving for the value of β which maximizes $\frac{\beta_2^T B \beta_2}{\beta_2^T W \beta_2}$, which gives the eigenvector corresponding to the second largest eigenvalue. This process continues until all betas have been found. There will be $K-1$ nonzero eigenvectors of $W^{-1}B$. If there are 2 classes that we are trying to classify data into, then the discriminant function can be written as $S^{-1}(\bar{x_1} - \bar{x_2})$, where S^{-1} is the inverse covariance matrix of the data and is multiplied by the difference between the mean vectors of the predictors for each class. A new sample, u, is projected onto the discriminant function as $uS^{-1}(\bar{x_1} - \bar{x_2})$. This will yield a discriminant score. The new sample will be classified into Class 1 if the sample is closer to the Class 1 mean than to the Group 2 mean in the projection:

$$|\beta^T(u - \bar{x_1})| - |\beta^T(u - \bar{x_2})| < 0$$

The solution for LDA consists of an inverted covariance matrix, and a unique solution only exists when the matrix is invertible. Therefore, the data must contain more observations than predictors, and the predictors must be independent. If there are more predictors than observations, then regularization methods can be used with LDA. [12, Section 12.3]

5.3 Penalization for Bayes' Theorem and Fisher's Approach for Linear Discriminant Analysis

Applying sparse penalization techniques to LDA will vary depending on the approach of LDA that is used. These techniques are followed from Section 8.4 in "Statistical Learning with Sparsity" [10, Section 8.4]. When performing LDA using Bayes' Theorem, we will use the nearest centroid rule, which is defined later. In very high dimensions, Σ will usually be of diagonal form since the predictors are assumed to be uncorrelated. With this assumption, one gets diagonal linear discriminant analysis. Letting $G(x) = \max_{\{k\in 1,..,K\}} \delta_k(x)$ and $\hat{\sigma}_j^2$ be the pooled within-class variance for the j^{th} predictor, the classification rule becomes

$$\hat{G}(x) = \arg\min_{l=1,...,K} \left\{ \sum_{j=1}^{p} \frac{(x_j - \hat{\mu}_{jl})^2}{\hat{\sigma}_j^2} - \log(\hat{\pi}_k) \right\}.$$

This is known as the nearest centroid rule. The nearest centroid rule uses all p predictors, but there may only be a subset of the predictors that are informative. Therefore, the model needs to reparametrized and a sparsity-inducing penalty needs to be imposed. In order to so, the mean vector for class k needs to be represented as $\mu_k = \bar{x} + \alpha_k$, where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the overall mean vector and $\alpha_k \in \mathbb{R}^p$, $k = 1, ..., K$ is the contrast for class k. This satisfies the constraint $\sum_{k=1}^{K} \alpha_k = 0$. The l_1-regularization criterion is then

optimized:

$$\text{minimize}_{\alpha_k \in \mathbb{R}^p, k=1,...,K} \left\{ \frac{1}{2N} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} \frac{(x_{ij} - \bar{x}_j - \alpha_{jk})^2}{\hat{\sigma}_j^2} + \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} \frac{\sqrt{N_k}}{\hat{\sigma}_j} |\alpha_{jk}| \right\}$$

subject to $\sum_{k=1}^{K} \alpha_{jk} = 0$ for $j = 1, ..., p$ and with C_k being the subset of indices i for which $g_i = k$ and $N_k = |C_k|$ is the total number of class-k samples. The solutions of α_k amount to simple soft-thresholding of particular class-wise contrasts as defined by Hastie et al. The steps are as follows, as outlined in "Statistical Learning with Sparsity" [10, Section 8.4]: The contrasts are defined as $d_{jk} = \frac{\tilde{x}_{jk} - \bar{x}_j}{m_k \alpha_j}$, where $\tilde{x}_{jk} = \frac{1}{n_k} \sum_{i \in C_k} x_{ij}$, $\bar{x}_j$ is he j^{th} component of the global mean $\bar{x}$, and $m_k^2 = \frac{1}{N_k} - \frac{1}{N}$. The soft-thresholding operator is then applied

$$d'_{jk} = S_\lambda(d_{jk}) = \text{sign}(d_{jk})(|d_{jk}| - \lambda)_+$$

and then the transformation is reversed to yield the shrunken centroid estimates of $\hat{\mu}'_{jk} = \bar{x}_j + m_k \sigma_j d'_{jk}$. These shrunken centroids are used for the estimates for μ_{jk} in the nearest centroid rule. If the contrast, d'_{jk}, is equal to zero by the soft-thresholding for each of the k classes, then that predictor is not a participant in the nearest centroid rule. Hence, the nearest shrunken centroid method performs feature selection. Also, a predictor may have $d_{jk} = 0$ for some classes, but not for all, and would therefore only play a role in those classes. When utilizing Fisher's LDA, the goal is to produce low-dimensional projections of the data while preserving the class separation. Sparsity using Fisher's Linear Discriminant Analysis, as proposed by Witten and Tibshirani, is as follows:

$$\text{maximize}_{\beta \in \mathbb{R}^p} \{\beta^T B \beta - \sum_{j=1}^{p} \hat{\sigma}_j |\beta_j|\}$$

subject to $\beta^T \tilde{W} \beta \leq 1$, where $\hat{\sigma}_j^2$ is the j^{th} diagonal element of W, and $\tilde{W}$ is a positive definite estimate for W. This yields a first sparse discriminant vector $\hat{\beta}_1$ with the level of sparsity to be determined by the choice of λ. Further components can be found by

first removing the current solution from B before solving Fisher's regularization LDA.

5.4 Optimal Scoring Method for Linear Discriminant Analysis

The goal of the optimal scoring approach of LDA is to recast the problem in terms of a multivariate linear regression, where the codes for the output classes are chosen optimally. Let Y be an indicator matrix of $n \times K$ dimension with n being the number of observations and K being the number of classes. The data is assumed to have been centered with mean of zero. The entries of the indicator matrix, Y, is as follows:

$$y_{ik} = \begin{cases} 1 & \text{if observation } i \text{ belongs to class } k \\ 0 & \text{otherwise} \end{cases}$$

[1, Section 2]. The optimal scoring approach to LDA will produce a sequence of discriminant vectors and scoring vectors. Suppose that the first k-1 discriminant vectors $\beta_1, ..., \beta_{k-1}$ and the scoring vectors $\theta_1, ..., \theta_{k-1}$ have been computed. The k^{th} discriminant vector β_k and the scoring vector θ_k is calculated by solving the following optimal scoring problem:

$$(\beta_k, \theta_k) = \arg \min_{\beta_k \in \mathbb{R}^p, \theta_k \in \mathbb{R}^K} \left\{ \frac{1}{n} \|Y\theta_k - X\beta_k\|_2^2 \right\}$$

$$\text{s.t. } \theta_k^T Y^T Y \theta_k = 1 \text{ and } \theta_k^T Y^T Y \theta_j = 0 \text{ for all } j = 1, 2, ..., k-1.$$

Details on the deriving of this optimal scoring problem can be found in Hastie et al., 1994. The optimal solution β_k of this problem is proportional to the Fisher linear discriminant criterion [10].

5.4.1 Regularization of Optimal Scoring Method for Linear Discriminant Analysis

Sparse discriminant vectors can be obtained by regularizing the optimal scoring problem with the generalized elastic-net penalty. This results in the modified optimiza-

tion problem:

$$(\beta_k, \theta_k) = \arg \min_{\beta_k \in \mathbb{R}^p, \theta_k \in \mathbb{R}^K} \left\{ \frac{1}{n} \|Y\theta_k - X\beta_k\|_2^2 + \lambda \|\beta_k\|_1 \right\}$$

$$\text{s.t. } \theta_k^T Y^T Y \theta_k = 1 \text{ and } \theta_k^T Y^T Y \theta_j = 0 \text{ for all } j = 1, 2, ..., k-1.$$

The l_1-penalty with a nonnegative regularization weight λ and a quadratic penalty defined by a positive semidefinite matrix Ω, equivalent to the elastic net penalty in the special case that $\Omega = \gamma I$ have been added. If the regularization weight λ on the l_1-penalty is sufficiently large, then the discriminant vectors will be sparse, and if $\lambda = 0$, then the minimizing criterion is the same as the penalized discriminant analysis proposal of Hastie et al. Despite the criterion being nonconvex, a local optimum can still be calculated using the elastic net to solve for β. If $p >> n$ and the predictors are not structured, then letting $\Omega = 0$ results in the problem being solved with the soft-thresholding algorithm for penalized matrix decomposition. If the problem has spatial or temporal structure, then the matrix Ω can be chosen to encourage spatial or temporal smoothness of the solution. With this case, the optimal scoring method is ideal since the quadratic term can be absorbed into the quadratic loss. Or the matrix Ω can be a diagonal matrix, and thus optimal scoring is convenient. Implementing an l_1-penalty and an l_2-penalty, which produces the elastic net penalty results in a modified optimization problem. As with the optimal scoring problem, the first k-1 discriminant vectors and the scoring vectors have already be calculated. The calculation of the k^{th} sparse discriminant vector β_k and scoring vector θ_k is done by solving the following optimal scoring problem:

$$(\beta_k, \theta_k) = \arg \min_{\beta_k \in \mathbb{R}^p, \theta_k \in \mathbb{R}^K} \left\{ \frac{1}{n} \|Y\theta_k - X\beta_k\|_2^2 + \gamma \beta_k^T \Omega \beta_k + \lambda \|\beta_k\|_1 \right\}$$

$$\text{s.t. } \theta_k^T Y^T Y \theta_k = 1 \text{ and } \theta_k^T Y^T Y \theta_j = 0 \text{ for all } j = 1, 2, ..., k-1,$$

with Y being an indicator matrix of n dimension, and γ and λ being nonnegative tuning parameters. The nonconvex spherical constraints make this optimization problem nonconvex, and therefore, a globally optimal solution using iterative methods cannot be found. Hence, Clemmensen et al., [4] proposed that a block coordinate descent method be used to iteratively approximate solutions. In particular, let's assume that an estimate (β^t, θ^t) of (β_k, θ_k) has been found. In order to update θ^t, fix $\beta{=}\beta_k^t$ and solve the original optimization problem for optimal scoring. This will produce the following problem:

$$\theta_k^t = \arg\min_{\theta_k \in \mathbb{R}^K} \|Y\theta_k - X\beta_k^t\|_2^2$$

$$\text{s.t. } \theta_k^T Y^T Y \theta_k = 1 \text{ and } \theta_k^T Y^T Y \theta_j = 0 \text{ for all } j = 1, 2, ..., k-1.$$

The above optimization problem is nonconvex and can be solved. Once θ^{t+1} has been updated, then β^{t+1} can be computed by solving the following:

$$\beta_k^{t+1} = \arg\min_{\beta_k \in \mathbb{R}^p} \left\{ \frac{1}{n}\|Y\theta_k^{t+1} - X\beta_k\|_2^2 + \gamma\beta_k^T\Omega\beta_k + \lambda\|\beta_k\|_1 \right\}.$$

5.5 Advantages to Regularizing Linear Discriminant Analysis

As previously stated, utilizing the sparsity criterion for LDA allows one to develop a more interpretable model that reduces overfitting on the training data. LDA often suffers when the number of predictors p is far greater than the number of observations n and when linear boundaries cannot separate the K classes [4, Section 1]. Hence, using sparsity to identify the best subset of predictors that will accurately model the data is one way to combat some of LDA's issues.

5.6 Accelerated Sparse Discriminant Analysis

Atkins et al., [1] proposed a collection of algorithms that will solve the elastic net problem using proximal operators. These proposed algorithms require fewer computational resources compared to elastic net. The following proposed algorithms will be discussed in this section: Proximal Gradient, Accelerated Proximal Gradient, and Alternating Direction.

5.6.1 Proximal Gradient Method

The proximal operator $prox_f$: $\mathbb{R}^p \to \mathbb{R}^p$ of f, which is a given convex function $\mathbb{R}^p \to \mathbb{R}$, is defined as:

$$\text{prox}_f(y) = \arg\min_{x \in \mathbb{R}^p} \left\{ f(x) + \frac{1}{2}\|x - y\|^2 \right\}.$$

This produces a point that balances the competing objectives of being near y while simultaneously minimizing f. This technique of using the proximal operator allows one to solve this optimization problem for the minimization of nonsmooth functions. For further background on this use of proximal operators, see Atkins et al., 2017. Considering

$$\beta_k^{t+1} = \arg\min_{\beta_k \in \mathbb{R}^p} \left\{ \frac{1}{n}\|Y\theta_k^{t+1} - X\beta_k\|_2^2 + \gamma\beta_k^T\Omega\beta_k + \lambda\|\beta_k\|_1 \right\},$$

expanding $\|Y\theta_k^{t+1} - X\beta_k\|_2^2$ and dropping the constant term will show that the above equation is equivalent to minimizing

$$f(\beta) = \frac{1}{2}\beta^T 2(X^TX + \gamma\Omega)\beta + (-2X^TY\theta^{t+1})^T\beta + \lambda\|\beta\|_1$$

$$f(\beta) = \frac{1}{2}\beta^T A\beta + d^T\beta + \lambda\|\beta\|_1,$$

with $A = 2(X^TX + \gamma\Omega)$ and $d = -2X^TY\theta^{t+1}$. Following some of the proximal operator's properties, f can be written as $f(\beta) = f_1(\beta) + f_2(\beta)$, with $f_1(\beta) = \frac{1}{2}\beta^T A\beta$ and $f_2(\beta) =$

$\lambda\|\beta\|_1$. The function for $f(\beta)$ will have a unique minimizer given that the penalty matrix Ω is positive definite, which makes f strongly convex. The proximal operator of the $f_2(\beta)$ is

$$\text{prox}_{\lambda\|\cdot\|_1}(y) = \text{sign}(y)\max\{|y| - \lambda e, 0\} =: S_\lambda(y).$$

The proximal operator $S_\lambda = \text{prox}_{\lambda\|\cdot\|_1}$ is sometimes referred to as the soft thresholding operator and sign: $\mathbb{R}^p \to \mathbb{R}^p$ and max: $\mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^p$ are element-wise sign and maximum mappings defined by:

$$[\text{sign}(y)]_i = \text{sign}(y_i) = \begin{cases} 1, & \text{if } y_i > 0 \\ 0, & \text{if } y_i = 0 \\ -1, & \text{if } y_i < 0 \end{cases}$$

and $[\max(x, y)]_i = \max(x_i, y_i)$. The proximal gradient method can now be applied to update β^t in the following manner:

$$\beta^{t+1} = \text{sign}(p^t)\max\left\{|p^t| - \lambda\alpha_t e, 0\right\},$$

with

$$p^t = \beta^t - \alpha_t \nabla f(\beta^t) = \beta^t - \alpha_t(A\beta^t + d),$$

and e and 0 represents a vector comprised of all ones and a vector comprised of all zeros, respectively.

5.6.2 Accelerated Proximal Gradient Method

Atkins et al., [1] proposes the use of momentum terms to accelerate convergence of the iterates. This is done by modifying the fast iterative soft thresholding algorithm (FISTA) [2, Section 4]. The goal of this technique is to accelerate the convergence of the iterates by taking the proximal gradient step from an extrapolation of the last two iterates. The updates for this accelerated proximal gradient method takes the following

form:

$$y^{t+1} = x^t + \omega_t(x^t - x^{t-1})$$

$$x^{t+1} = \text{prox}_{\alpha g}(y^{t+1} - \alpha \nabla f(y^{t+1})),$$

with $\omega_t \in [0, 1)$ is an extrapolation parameter. A standard choice for this parameter is $\frac{t}{t+3}$. This modification of the proximal gradient method yields a sequence of iterates that converge to the optimal solution at a quicker rate.

5.6.3 Alternating Direction Method of Multipliers

The Alternating Direction Method of Multipliers (ADMM) solves problems with the following form

$$\min_{x \in \mathbb{R}^p, y \in \mathbb{R}^m} \Big\{ f(x) + g(y) : Ax + By = c \Big\},$$

by utilizing an approximate dual gradient ascent, where $f : \mathbb{R}^p \to \mathbb{R}, g : \mathbb{R}^m \to \mathbb{R}, A \in \mathbb{R}^{r \times p}, B \in \mathbb{R}^r,$ and $c \in \mathbb{R}^r$. By splitting the decision variable $\beta \in \mathbb{R}^p$ as two new variables $x, y \in \mathbb{R}^p$ with a linear coupling constraint of $x = y$ allows

$$\min_{\beta \in \mathbb{R}^p} f(\beta) = \min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2} \beta^T 2(X^T X + \gamma \Omega)\beta + (-2X^T Y \theta^{t+1})^T \beta + \lambda \|\beta\|_1 \right\}$$

$$f(\beta) = \frac{1}{2} \beta^T A \beta + d^T \beta + \lambda \|\beta\|_1$$

to be rewritten as

$$\min_{x, y \in \mathbb{R}^p} \Big\{ \frac{1}{2} x^T A x - x^T d + \lambda \|y\|_1 : x - y = 0 \Big\}.$$

This rewritten form allows ADMM to be used to generate a sequence of iterates using approximate dual gradient ascent steps. First, consider the iterates (x^t, y^t, z^t) after t steps of the algorithm. The update for x is given by the following

$$x^{t+1} = \arg\min_{x \in \mathbb{R}^p} L_\mu(x, y^t, z^t) = \arg\min_{x \in \mathbb{R}^p} \frac{1}{2} x^T (\mu I + A) x - x^T (d + \mu y^t - z^t),$$

where $L_\mu(x, y, z)$ represents the augmented Lagrangian and $\mu > 0$ is a penalty parameter controlling the emphasis on enforcing feasibility of the primal iterates x and y. The augmented Lagrangian is

$$L_\mu(x, y, z) = \frac{1}{2}x^T Ax - x^T d + \lambda\|y\|_1 + z^T(x - y) + \frac{1}{2}\mu\|x - y\|^2$$

for all $x, y \in \mathbb{R}^p$. After the application of the the first order necessary and sufficient conditions for optimality, x^{t+1} must satisfy the following:

$$(\mu I + A)x^{t+1} = d + \mu y^t - z^t.$$

Taking the Cholesky decomposition of $\mu I + A = BB^T$ and solve for x^{t+1} by solving the following triangular systems

$$BB^T x^{t+1} = d + \mu y^T - z^T.$$

With the generalized elastic net matrix Ω being diagonal, the Sherman-Morrison-Woodbury formula can be applied to more efficiently solve the linear system. More details on this procedure is available in Atkins et al., 2017, Section 2.1. Next y can be updated by

$$y^{t+1} = \arg\min_{y\in\mathbb{R}^p} L_\mu(x^{t+1}, y, z^t) = \arg\min_{y\in\mathbb{R}^p} \lambda\|y\|_1 + \frac{1}{2}\mu\|y - x^{t+1} - \frac{z^t}{\mu}\|.$$

Hence, y^{t+1} is updated as the value of the soft thresholding operator of the l_1-penalty at $\frac{z^t}{\mu} + x^{t+1}$:

$$y^{t+1} = S_\lambda(x^{t+1} + \frac{z^t}{\mu}) = \text{sign}(x^{t+1} + \frac{z^t}{\mu}) \max\left\{|x^{t+1} + \frac{z^t}{\mu}| - \lambda e, 0\right\}.$$

And lastly, z can be updated using the approximate dual ascent step by

$$z^{t+1} = z^t + \mu(x^{t+1} - y^{t+1}).$$

Under certain strong convexity assumptions of f and g and rank assumptions on A and B, it is known that ADMM creates a sequence of iterates that converge linearly to an optimal solution of

$$\min_{x \in \mathbb{R}^p, y \in \mathbb{R}^m} \Big\{ f(x) + g(y) : Ax + By = c \Big\}.$$

All of these assumptions are satisfied through the optimization problem of

$$\min_{x, y \in \mathbb{R}^p} \Big\{ \frac{1}{2} x^T A x - x^T d + \lambda \|y\|_1 : x - y = 0 \Big\},$$

with Ω being positive definite. Thus the iterates x^t, y^t, z^t converges to a minimizer of $f(\beta)$. Hence, $x^t - y^t \to 0$ and $f(x^t), f(y^t)$ converge linearly to the minimum value of f [1, Section 2].

RESULTS

With this research, we consider the sparse regression techniques as tools for classification of sentiment within Twitter posts.

6.1 Twitter Analysis for Autism Sentiments

As an illustrative example, we employ sparse regression techniques to classify tweets based on the users' perception of a connection between vaccinations and autism spectrum disorder. In other words, we are tasked with correctly classifying whether or not a user believes that autism spectrum disorder is caused by vaccinations based on the content of their tweets. The data for this research consists of two sets of tweets that were manually labeled in the following scheme: pro-vaccination (1), anti-vaccination (2), neutral (3), unsure (4), unrelated (5), and unidentified (6) based on their content by two different individuals [19]. Those tweets which had a label other than pro-vaccination (1) or anti-vaccination (2) were discarded and not considered in this research. The two sets of aforementioned data originally consisted of the same 2000 tweets, but were reduced during the two independent manual labeling processes. The training data, which consists of one set of tweets labeled by one individual, has 1543 observations/tweets, and the testing data, which is the set of tweets labeled by the other individual, consists of 1827 observations/tweets. From both data sets, stop words such as: is, was, he, she, it, the, etc, were removed along with all punctuation. Each data set was then converted into a document term matrix using only the dictionary from the training data. The document term matrices are comprised of rows, which are the documents or tweets, and columns, which are the terms or predictors from all of the tweets in the training data.

The total amount of predictors or features in both data sets is 5281. Hence, the training data is a 1543×5281 data matrix, and the testing data is a 1827×5281 data matrix. The approaches of analysis conducted in this research include linear regression, logistic regression, sparse logistic regression, sparse logistic regression with an optimal threshold, and accelerated sparse discriminant analysis.

6.1.1 Linear Regression Results

The first method performed on this data was linear regression. Because this data has multiple predictor variables, the linear regression model had the following form

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} + \epsilon_i$$

with Y_i being the quantitative response with labels of 1 or 2, for pro-vaccination and anti-vaccination sentiments, respectively, and $X_{i1}, X_{i2}, ..., X_{ip}$ being the terms in the dictionary or predictors. The unknown model parameters β for the training data was computed using the lm command in R, which fits linear models and carries out linear regression. The linear model obtained was then used to predict the response values, Y_i, for the testing data. Once these values were predicted, they were rounded to either 1 for pro-vaccination sentiment or 2 for anti-vaccination sentiment, which were the given labels. This rounding process was carried out in the standard rounding process, other than the fact that all of the values below 0 were treated as having a value of 1, and all of the values above 2 were treated as having a value of 2. The misclassification rate for this linear model is 52.9%, which is worse than making an educated guess for the labels of each tweet. The large misclassification rate could be contributed to by the arbitrary labeling of 1 and 2 for the response variable. More importantly, this model further indicates that the relationship between the predictor variable and the response variable is not a linear relationship.

6.1.2 Logistic Regression and Sparse Logistic Regression Results

Logistic regression was the next technique that was implemented on this textual data. This approach seems more natural, as we are trying to accurately classify one's feelings about vaccinations and autism. In other words, we are attempting to categorize the tweets into one of two categories; therefore, logistic regression seems to be a more natural fit for this type of analysis. Once again, there are multiple predictors within this data, so the logistic regression model will take on the following form

$$p(X) = \frac{e^{\beta_0+X_1\beta_1+...+X_p\beta_p}}{1+e^{\beta_0+X_1\beta_1+...+X_p\beta_p}} = \frac{e^{X\beta}}{1+e^{X\beta}}$$

with X being the n x p document term matrix and β_i being the unknown model parameters. The logistic regression aims to predict the probability that each predictor belongs to a particular response class. Prior to running logistic regression in R on the training data, the response variable Y_i, which was originally 1 or 2, was relabeled to 0 and 1, respectively, since these are the usual response variables for logistic regression. To carry out logistic regression, the glmnet function in R was used. The model was trained on the training data, and the coefficient parameters computed were used to predict the probability that each predictor belonged to one of two classes for the testing data. Just as described in the logistic regression section of this paper, the following maximum likelihood problem was solved

$$\min\left\{ -\frac{1}{n}\sum_{i=1}^{n}\left[Y_i\beta - \log(1+e^{X_i\beta})\right]\right\}$$

Due to the number of features being far larger than the number of observations, the model produced was very degenerate, and therefore yielded a large misclassification rate of 66.3%. This large misclassification rate is assumed to mainly be caused by overfitting of the logistic model. Thus, the model was regularized using the l_1-penalty or LASSO and the l_2-penalty or ridge regression, separately. The sparse logistic regression problem

for both LASSO and ridge are indicated below, respectively.

$$\min\left\{ -\frac{1}{n}\sum_{i=1}^{n}\left[Y_i\beta - \log(1+e^{X_i\beta})\right] + \lambda\|\beta\|_1 \right\}$$

$$\min\sum_{i=1}^{n}\left\{(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ip})^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right\};$$

100-fold cross-validation was performed using the glmnet function in R, with lambda.min being the chosen λ, which yields the minimum cross-validation error.

As before with the standard logistic regression, the models obtained from both ridge regression and LASSO were then used to make predictions on the testing data. Both ridge regression and LASSO performed fairly well, but ridge uses all of the 5281 features, which makes interpreting the model more difficult. On the other hand, LASSO had a slightly lower misclassification rate and due to its variable selection, this model is more interpretable. The ridge regression model produced the following truth table, which had a misclassification rate of 0.3766, and the LASSO model produced the truth table listed below with a misclassification rate of 0.3027. It should be noted that all the columns of the truth tables in this chapter represent the actual classifications and the rows represent the predicted classifications.

Table 6.1: Truth Table for Ridge Regression

	0	**1**
0	264	72
1	459	1032

Table 6.2: Truth Table for LASSO

	0	**1**
0	246	76
1	477	1028

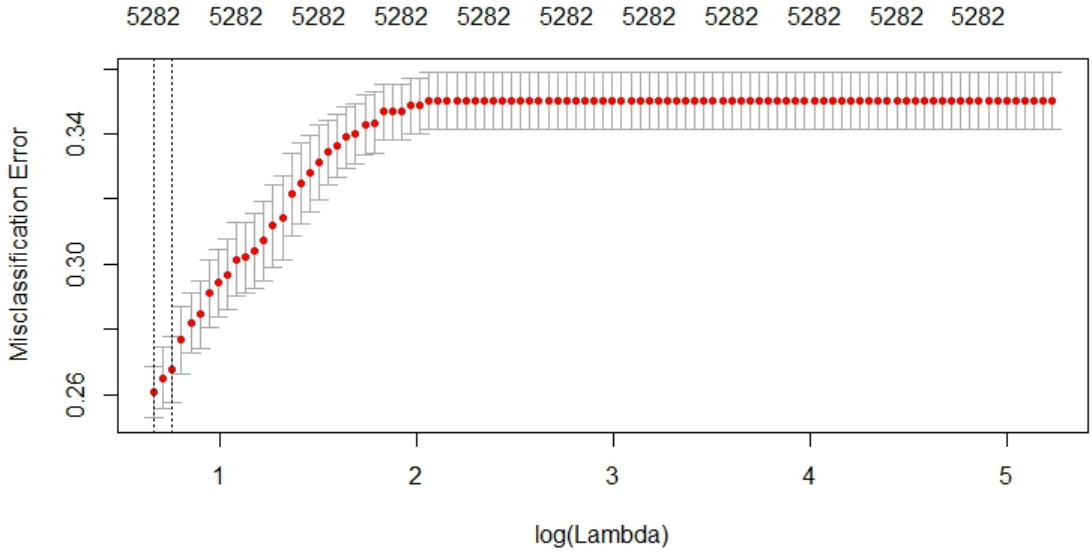

Figure 6.1: Plot Cross-Validation for Ridge Regression

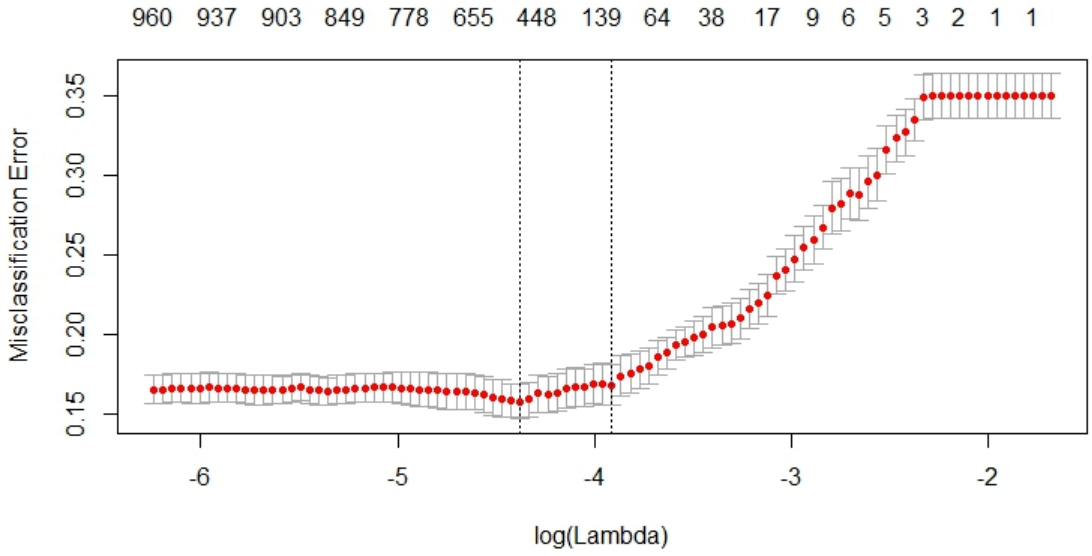

Figure 6.2: Plot for Cross-Validation for LASSO

Despite having a fairly low misclassification rate for both the ridge regression and LASSO, the regularizations cause the misclassification rate for the response classes to be imbalanced. For the ridge regression, response class 0 has a misclassification rate of 0.6349 and response class 1 has a misclassification rate of 0.0652. LASSO produces a misclassification rate of 0.6598 for response class 0 and a misclassification rate of 0.0688 for response class 1.

Applying the ridge regression and LASSO models back to the training data yields in-sample results. The in-sample misclassification rate for ridge regression is 0.0032, and the in-sample misclassification rate for LASSO is 0.0093. The truth table for both in-sample ridge regression and LASSO are below. Ridge regression yields a misclassification rate of 0.0093 for response class 0, and a misclassification rate of 0 for response class 1. LASSO produces a misclassification rate of 0 for response class 0 and a misclassification rate of 0.0784 for response class 1.

Table 6.3: Truth Table for Ridge Regression (In-Sample)

	0	**1**
0	535	0
1	5	1003

Table 6.4: Truth Table for LASSO (In-Sample)

	0	**1**
0	535	79
1	0	929

Since the implementation of l_1-penalty uses variable selection, a feature list is produced to select which features are most closely associated with each response class. The feature lists with the top ten terms most associated with each sentiment are below.

As outlined above, the sparse logistic regression with the l_1-penalty yields a sparse

Table 6.5: Feature List for Pro-Vaccination Sentiment

Term	Coefficient
httptco2xdyqkpf1g	-3.77E+00
cnncom	-3.78E+00
idiot	-3.84E+00
debate	-3.95E+00
mouth	-3.95E+00
due	-4.00E+00
antivaxx	-4.05E+00
diagnosi	-4.95E+00
skleffman	-5.72E+00
httptcoxwg26frn	-6.22E+00

Table 6.6: Feature List for Anti-Vaccination Sentiment

Term	Coefficient
induc	8.14E+00
autismvaccinescauseautismdrwakefieldhttptco2o39ynuv5u	7.46E+00
amish	6.43E+00
httptco8e5ldg6m	5.76E+00
autismreally	5.39E+00
cdcwhistleblow	4.58E+00
equal	4.18E+00
illuminati	4.17E+00
byte	4.11E+00
cost	4.00E+00

model with most model parameters equal to zero. However, due to degeneracy, there are many solutions and thus there may be a better model to improve the misclassification rate of the testing data. The following table summarizes all of the misclassification rates for the standard logistic regression.

Table 6.7: Logistic Regression Results

	Ridge Regression	LASSO
Misclassification Rate	0.2906	0.3027
	In-Sample Results	
Misclassification Rate	0.0032	0.0093

6.1.3 Sparse Logistic Regression with Optimal Threshold Results

Due to the imbalance in errors among the response classes of both ridge regression and LASSO of the logistic regression, the method of using the optimal threshold with logistic regression was performed in hopes of balancing out the number of errors in each class. In other words, the sparse logistic models did a poor job at accurately identifying the observations that should have been labeled as 0, but it did a fairly good job at identifying observations that were labeled as 1. The general idea with the optimal threshold technique is to gain an overall lower misclassification rate, while also producing a lower misclassification rate among each response class. The optimal threshold for both LASSO and ridge regression was conducted using the optimalCutoff function in R [18, Section 3.1]. The optimal threshold and misclassification rate for both the ridge regression and LASSO penalty is displayed below.

Table 6.8: Sparse Logistic Regression with Optimal Threshold Results

	Ridge Regression	LASSO
	Optimal Threshold	Optimal Threshold
Misclassification Rate	0.1834	0.2731

Table 6.9: Truth Table for Ridge Regression with Optimal Threshold

	0	**1**
0	499	111
1	224	993

Table 6.10: Truth Table for LASSO with Optimal Threshold

	0	**1**
0	366	142
1	357	962

The in-sample results for the optimal threshold for both ridge regression and LASSO are below.

Table 6.11: In-Sample Sparse Logistic Regression with Optimal Threshold Results

	Ridge Regression	LASSO
	Optimal Threshold	Optimal Threshold
Misclassification Rate	0.0032	0.1005

Table 6.12: Truth Table for Ridge Regression with Optimal Threshold (In-Sample)

	0	**1**
0	537	2
1	3	1001

Table 6.13: Truth Table for LASSO with Optimal Threshold (In-Sample)

	0	**1**
0	467	82
1	73	921

6.1.3.1 Sparse Logistic Regression with Mean Threshold Results

Since the optimalCutoff function in R utilizes the actual response variables, and these are not always known in real-world applications, the mean of the predicted values of the testing data was used as the threshold. This yielded the following results:

The misclassification rate for the Ridge regression using the mean as the threshold is 0.1872, and the misclassification rate for the LASSO using the mean as the threshold

Table 6.14: Truth Table for Ridge Regression with Mean Threshold

	0	**1**
0	575	194
1	148	910

Table 6.15: Truth Table for LASSO with Mean Threshold

	0	**1**
0	367	137
1	356	967

is 0.2698. These results are very comparable to the results that were produced using the optimal threshold technique. For ridge regression, the misclassification rate for class 0 and class 1 is 0.2047 and 0.2132, respectively. The LASSO produces the following misclassification rates for class 0 and class 1 respectively: 0.4924 and 0.1241.

Utilizing the mean of the data as the threshold also yielded the following in-sample results:

Table 6.16: In-Sample Sparse Logistic Regression with Mean Threshold Results

	Ridge Regression	LASSO
	Mean Threshold	Mean Threshold
Misclassification Rate	0.0078	0.0752

Table 6.17: Truth Table for Ridge Regression with Mean Threshold (In-Sample)

	0	**1**
0	540	12
1	0	991

The following table summarizes the sparse logistic regression results for both optimal and mean thresholds.

Table 6.18: Truth Table for LASSO with Mean Threshold (In-Sample)

	0	**1**
0	535	111
1	5	892

Table 6.19: Sparse Logistic Regression with Threshold Results

	Ridge Regression		LASSO	
	Opt Thresh	Mean Thresh	Opt Thresh	Mean Thresh
M.C. Rate	0.1834	0.1872	0.2731	0.2698
	In-Sample Results			
M.C. Rate	0.0032	0.0078	0.1005	0.0752

6.1.4 Accelerated Sparse Discriminant Analysis Results

We use the package accSDA in R to perform dimension reduction in a faster method. The data was uploaded into R, and a cross-validation scheme utilizing both λ and γ were performed.

To begin, the training data was split into training observations and validation observations using 10-folds. The ASDA method was trained on this data using a range of λs and γs. The λs chosen were $0.01, 0.001,$ and 0.00001, and the chosen γs were $0.001, 0.0001,$ and 0.000001. These values were chosen after performing a trial run using ASDA, in which a larger range of λ and γ were used. Once running this trial run, the values of λ and γ close to the chosen values produced the lowest misclassification rate. The misclassification rates for each λ, γ pair was calculated, and the pair with the lowest misclassification rate was used in the final model. The final ASDA model was then run on the testing data to produce the following results. A misclassification rate of 0.2852 was yielded, and the following truth table was produced:

This model produces a large misclassification rate of 0.6321 for the classification of response class 0, and therefore this model can be improved by tuning the threshold. The misclassification rate of response class 1 is 0.0580. It can be seen that there is

Table 6.20: Truth Table for ASDA

	0	**1**
0	266	64
1	457	1040

an extreme imbalance of misclassification with this model. Therefore, even though the overall misclassification rate of the model is low, the number of misclassifications within response class 0 is very large. The goal for performing ASDA is to get results that surpass those of the above techniques that have been used. Hence, this technique should lower the misclassification rate of the entire data set and for each response class. Therefore, the next logical step towards making these improvements is to apply the optimal threshold component to this ASDA technique.

6.1.5 Accelerated Sparse Discriminant Analysis with Optimal Threshold Results

Just as with the above ASDA technique, cross validation was performed using the same λs and γs as before. Then the optimalCutoff function in R was utilized to obtain the optimal threshold based on the predicted scores of the training set of the training data. The λ and γ pair that produced the least amount of total misclassifications were chosen and used for the model along with an optimal threshold that was computed using the predicted scores for the testing data. This final ASDA model was run on the testing data and yielded the following results:

Table 6.21: Truth Table for ASDA with Optimal Threshold

	0	**1**
0	450	192
1	273	912

The overall misclassification rate for this model is 0.2545, which is lower than the above ASDA model. Response class 0 has a misclassification rate of 0.3776, and response class 1 has a misclassification rate of 0.1739. The misclassification rates of these response

classes are more balanced than the previous results. The in-sample results are as below. The overall misclassification rate for this in-sample model is 0.1426 with response class 0

Table 6.22: Truth Table for ASDA with Optimal Threshold (In-Sample)

	0	**1**
0	434	114
1	106	889

having a misclassification rate of 0.1963, and response class 1 having a misclassification rate of 0.1137. The in-sample results outperform the out-of-sample results, as should be the case, since the model is re-run on the training data. However, these in-sample results have a large misclassification rate considering that the model was applied to the data in which is was trained on. To improve these results, the choices of λ and γ will be explored more closely in the next subsection.

6.1.5.1 Accelerated Sparse Discriminant Analysis with Mean Threshold Results

Once again since the exact labels of the data may not be known, the model was adjusted to use the mean of the scores of the data as the threshold. The procedure was the same as with the optimal threshold for ASDA technique except rather than using the optimalCutoff command in R, the mean scores of the data was used. This yielded the following results: The overall misclassification rate for this model is 0.2616, which is slightly

Table 6.23: Truth Table for ASDA with Mean Threshold

	0	**1**
0	477	232
1	246	872

larger than the misclassification rate with the optimal threshold utilized. Response class 0 has a misclassification rate of 0.3402, and response class 1 has a misclassification rate of 0.2101. The in-sample results for ASDA with the mean threshold are as follows: The overall in-sample misclassification rate for this model is 0.1925, which is also larger than

Table 6.24: Truth Table for ASDA with Mean Threshold (In-Sample)

	0	**1**
0	520	277
1	20	726

the rate for the ASDA model using the optimal threshold. Response class 0 and response class 1 had misclassification rates of 0.0370 and 0.2762, respectively. Both of these rates are also larger than those using the optimal threshold. Improvements to these models are made in the following subsection by extending the range of the values of λ and γ in the ASDA cross-validation scheme.

6.1.6 Accelerated Sparse Discriminant Analysis with Extended Range of λ and γ Results

Extending the range of λ and γ combined with the use of the optimal threshold yields the results below. The values of λ are $0.01, 0.001, 0.00001, 0.0000001, 0.02, 0.002, 0.03,$ 0.003, and 0.04. The values of γ are $0.001, 0.0001, 0.000001, 0.02,$ and 0.04. The results using these values of λ and γ are summarized as follows:

Table 6.25: Truth Table for ASDA with Optimal Threshold

	0	**1**
0	425	189
1	289	915

The overall misclassification rate for this model is 0.2666 with response class 0 and response class 1 having misclassification rates of 0.4122 and 0.1712, respectively. The number of misclassifications in this model are more than the number of misclassifications using the aforementioned values of γ and λ, which allots for the increase in the misclassification rates. It can be seen that there is also an imbalance in the number of errors for response class 0 and response class 1. The in-sample results for this model are below:

Table 6.26: Truth Table for ASDA with Optimal Threshold (In-Sample)

	0	**1**
0	367	92
1	173	911

The misclassification rate for this model is 0.1717 and response class 0 has a misclassification rate of 0.3204, and response class 1 has a misclassification rate of 0.0917. The total number of misclassifications for this in-sample data is fairly large; however, the the misclassification rate for response class 1 is extremely low.

6.1.6.1 Accelerated Sparse Discriminant Analysis with Mean Threshold Results

Just as with the above models, this model was modified to use the mean of the scores of the data as the threshold. Doing so yielded the following results: The

Table 6.27: Truth Table for ASDA with Mean Threshold

	0	**1**
0	477	232
1	246	872

misclassification rate for this model is 0.2616 with response class 0 and response class 1 having misclassification rates of 0.3402 and 0.2101, respectively. This model produced the slightly lower misclassification rates overall and for each of the response classes than the above model with the optimal threshold. It should also be noted that the errors between the response classes are closer than the errors between the response classes when using the optimal threshold. The in-sample results for this model is below. The

Table 6.28: Truth Table for ASDA with Mean Threshold (In-Sample)

	0	**1**
0	434	114
1	106	889

in-sample misclassification rate is 0.1925 with response class 0 and response class 1 having misclassification rates of 0.1963 and 0.1137, respectively. Although the overall misclassification rate for this model is larger than the in-sample rate using the optimal threshold, the misclassification rate for the response classes are more balanced.

With the extension in the range of values of λ and γ for ASDA with the optimal and mean thresholds, utilizing the mean thresholds yielded more balanced response class misclassifications than the optimal threshold. Also, the overall misclassification rates when the mean scores were used as the threshold was comparable to the rates when the optimal threshold was used.

6.1.7 Accelerated Sparse Discriminant Analysis with Ridge Regression Results

Lastly, since ridge regression produced relatively low misclassification rates for logistic regression models, it was implemented into ASDA with the same parameters of using the optimal threshold and the mean threshold. Applying ridge regression to ASDA with the optimal threshold yielded the results below.

Table 6.29: Truth Table for ASDA with Ridge Regression and Optimal Threshold

	0	**1**
0	440	181
1	283	923

The misclassification rate for this model is 0.2540, which is lower than the misclassification rate of the standard ASDA with the optimal threshold model. Response class 0 and response class 1 have misclassification rates of 0.3914 and 0.1639, respectively. The response class misclassification rates are slightly more imbalanced than the standard ASDA with optimal threshold model. However, response class 1 does have a lower rate than response class 1 of the standard ASDA with optimal threshold model. The in-sample results for ASDA with ridge regression and optimal threshold is below.

This in-sample model has a misclassification rate of 0.1432 with response class

Table 6.30: Truth Table for ASDA with Ridge Regression and Optimal Threshold (In-Sample)

	0	**1**
0	428	109
1	112	894

0 and response class 1 having misclassification rates of 0.2071 and 0.1087, respectively. Once again the imbalance of errors between the response classes is slightly larger, but response class 1 has a lower misclassification rate than the standard in-sample ASDA with optimal threshold model.

6.1.7.1 Accelerated Discriminant Analysis with Ridge Regression and Mean Threshold Results

Following the same procedure as before, the mean of the scores was used as the threshold for the data in place of the optimal threshold. The following results were obtained.

Table 6.31: Truth Table for ASDA with Ridge Regression and Mean Threshold

	0	**1**
0	478	230
1	245	874

This model produced a misclassification rate of 0.2600, which is lower than the misclassification rate yielded by the standard ASDA with mean threshold model. Response class 0 and response class 1 had misclassification rates of 0.3389 and 0.2083, respectively. Both of these response class misclassification rates are also lower than those of the standard ASDA with mean threshold model. However, the imbalance of errors among each response class is roughly the same as that produced by the standard ASDA model with mean threshold.

The in-sample results for the ASDA with ridge regression and mean threshold

model are below.

Table 6.32: Truth Table for ASDA with Ridge Regression and Mean Threshold (In-Sample)

	0	**1**
0	521	277
1	19	726

This in-sample model yielded a misclassification rate of 0.1918 with response class 0 and response class 1 having a misclassification rate of 0.0352 and 0.2762, respectively. Although the overall misclassification rate for this in-sample model is lower than the misclassification rate for the standard in-sample ASDA with mean threshold model, the response class errors are more unbalanced. In fact, this model did better at predicting the response class 0 than it did at predicting response class 1, which has not been the case for all of the other models.

The following tables summarize the results for ASDA with optimal and mean thresholds and with the extended values of λ and γ. These tables also include the in-sample results for each of the models.

Table 6.33: Accelerated Sparse Discriminant Analysis with Threshold Results

	Ridge Regression		LASSO	
	Opt Thresh	Mean Thresh	Opt Thresh	Mean Thresh
Misclassification Rate	0.2540	0.2600	0.2545	0.2616
		In-Sample Results		
Misclassification Rate	0.1432	0.1918	0.1426	0.1925

Table 6.34: Accelerated Sparse Discriminant Analysis with Threshold and 10 λs and 5 γs Results

	Optimal Threshold	Mean Threshold
Misclassification Rate	0.2666	0.2616
	In-Sample Results	
Misclassification Rate	0.1717	0.1426

6.1.8 Analysis of Autism Data with 80/20-Partition of Data Results

In this subsection, the autism data was partitioned into a training set and a validation set. The training set consisted of 80% of the original autism training data, and the validation set was comprised of the remaining 20% of the training data. For sparse logistic regression, the ridge regression and LASSO were obtained with and without an optimal threshold. For the optimal thresholds, there were two thresholds that were used within this method with the intention of improving the model by lowering the misclassification rates and balancing the number of misclassification within each response class. The first threshold optimizes for the minimum misclassification error of the model, and the second optimal threshold yields the maximum Youden's Index. Youden's Index tells how well a model has performed through the use of sensitivity and specificity. The formula for Youden's Index is sensitivity + specificity - 1, or

$$\text{Youden's Index} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} + \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}} - 1.$$

Both of these optimal thresholds were used in both the sparse logistic regression models and the ASDA models. In the following sub-subsections the results obtained from these models will be presented.

6.1.8.1 Sparse Logistic Regression with 80/20-Split Results

Sparse logistic regression using only the original training data with 80% of the data being the training set and the remaining 20% of the data being the validation set was conducted. This yields ridge regression and LASSO models. With these regularization models, the implementation of the optimal threshold optimizing for both the minimum misclassification error and the maximum of Youden's Index was used. The results are as follows. The sparse logistic regression results with no optimal threshold are below.

Table 6.35: Truth Table for Ridge Regression

	0	**1**
0	27	1
1	85	196

The misclassification rate for this model is 0.2783 with response class 0 and response class 1 having a misclassification rate of 0.7589 and 0.0051, respectively.

Table 6.36: Truth Table for LASSO

	0	**1**
0	79	15
1	33	182

The misclassification rate for this LASSO model is 0.1553 with response class 0 and response class 1 having a misclassification rate of 0.2946 and 0.0761, respectively.

The in-sample results for ridge regression and LASSO with no optimal threshold are below.

Table 6.37: Truth Table for Ridge Regression (In-Sample)

	0	**1**
0	353	0
1	75	806

The misclassification rate for this model is 0.0608 with response class 0 and response class 1 having a misclassification rate of 0.1752 and 0, respectively.

Table 6.38: Truth Table for LASSO (In-Sample)

	0	**1**
0	346	22
1	82	784

The misclassification rate for this in-sample LASSO model is 0.0843 with response class 0 and response class 1 having a misclassification rate of 0.1916 and 0.0273, respectively.

6.1.8.2 Sparse Logistic Regression with Optimal Threshold Results

For the optimal threshold, as stated above, two of them will be utilized. One optimal threshold will optimize for the minimum number of misclassifications for the entire model, and the second optimal threshold will optimize for the maximum value of Youden's Index. Including the optimal threshold which minimizes the number of misclassifications yielded the following results.

Table 6.39: Truth Table for Ridge Regression with Optimal Threshold for Misclassification

	0	**1**
0	106	31
1	6	166

The misclassification rate for this ridge regression model is 0.1197 with response class 0 and response class 1 having a misclassification rate of 0.0536 and 0.1574, respectively.

Table 6.40: Truth Table for LASSO with Optimal Threshold for Misclassification

	0	**1**
0	91	17
1	21	180

The misclassification rate for this LASSO model is 0.1230 with response class 0 and response class 1 having a misclassification rate of 0.1875 and 0.0863, respectively.

The results for sparse logistic regression using the optimal threshold that maximizes Youden's Index are below.

Table 6.41: Truth Table for Ridge Regression with Optimal Threshold for Youden's Index

	0	**1**
0	106	31
1	6	166

The misclassification rate for this ridge regression model is 0.1197 with response class 0 and response class 1 having a misclassification rate of 0.0536 and 0.1574, respectively.

Table 6.42: Truth Table for LASSO with Optimal Threshold for Youden's Index

	0	**1**
0	91	17
1	21	180

The misclassification rate for this LASSO model is 0.1230 with response class 0 and response class 1 having a misclassification rate of 0.1875 and 0.0863, respectively.

The in-sample results for both of these models are included below.

Table 6.43: Truth Table for Ridge Regression with Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	425	1
1	3	805

The in-sample misclassification rate for this model is 0.0032 with response class 0 and response class 1 having a misclassification rate of 0.0070 and 0.0012, respectively.

Table 6.44: Truth Table for LASSO with Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	424	59
1	4	744

The misclassification rate for this in-sample LASSO model is 0.0511 with response class 0 and response class 1 having a misclassification rate of 0.0093 and 0.0073, respectively.

The results for the in-sample sparse logistic regression models using the optimal threshold that maximizes Youden's Index is below.

Table 6.45: Truth Table for Ridge Regression with Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	426	2
1	2	804

The misclassification rate for this in-sample ridge regression model is 0.0032 with response class 0 and response class 1 having a misclassification rate of 0.0047 and 0.0025, respectively.

Table 6.46: Truth Table for LASSO with Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	424	59
1	4	744

The misclassification rate for this in-sample LASSO model is 0.0511 with response class 0 and response class 1 having a misclassification rate of 0.0093 and 0.0073, respectively.

6.1.8.3 Sparse Logistic Regression with Mean Threshold Results

Using the mean scores of the data as the threshold for sparse logistic regression models yielded the results below. Since the mean scores of the data are being used as the threshold, there is no way to optimize for the smallest misclassification rate or the maximum Youden's Index. Thus, there is only one set of out-of-sample and in-sample results.

Table 6.47: Truth Table for Ridge Regression with Mean Threshold

	0	**1**
0	104	31
1	8	166

The misclassification rate for this model is 0.1262 with response class 0 and response class 1 having a misclassification rate of 0.0714 and 0.1574, respectively.

Table 6.48: Truth Table for LASSO with Mean Threshold

	0	**1**
0	95	33
1	17	164

The misclassification rate for this model is 0.1618 with response class 0 and response class 1 having a misclassification rate of 0.1518 and 0.1675, respectively.

The in-sample results for both of these sparse logistic models with the mean threshold are included below.

Table 6.49: Truth Table for Ridge Regression with Mean Threshold (In-Sample)

	0	**1**
0	426	6
1	2	800

The misclassification rate for this model is 0.0065 with response class 0 and response class 1 having a misclassification rate of 0.0047 and 0.0074, respectively.

Table 6.50: Truth Table for LASSO with Mean Threshold (In-Sample)

	0	**1**
0	424	114
1	4	692

The misclassification rate for this model is 0.0956 with response class 0 and response class 1 having a misclassification rate of 0.0093 and 0.1414, respectively.

6.1.8.4 Accelerated Sparse Discriminant Analysis with Optimal Threshold Results

In this sub-subsection, the results of ASDA with the implementation of an optimal threshold with 80% of the original training data being the training set and the remaining 20% of the training data being the validation set is detailed. The results for ASDA using the optimal threshold that minimizes the number of misclassifications are below.

Table 6.51: Truth Table for ASDA with Optimal Threshold for Misclassification

	0	**1**
0	76	21
1	25	187

The misclassification rate for this model is 0.1489 with response class 0 and response class 1 having a misclassification rate of 0.2475 and 0.1010, respectively.

The results for ASDA using the optimal threshold that maximizes Youden's Index are below.

Table 6.52: Truth Table for ASDA with Optimal Threshold for Youden's Index

	0	**1**
0	90	37
1	11	171

The misclassification rate for this model is 0.1553 with response class 0 and response class 1 having a misclassification rate of 0.1089 and 0.1779, respectively.

The in-sample results for both of these ASDA models with optimal thresholds are included below.

Table 6.53: Truth Table for ASDA with Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	381	107
1	58	688

The misclassification rate for this in-sample model is 0.1337 with response class 0 and response class 1 having a misclassification rate of 0.1321 and 0.1346, respectively.

The in-sample results for ASDA using the optimal threshold that maximizes Youden's Index are below.

Table 6.54: Truth Table for ASDA with Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	385	113
1	54	682

The misclassification rate for this in-sample model is 0.1353 with response class 0 and response class 1 having a misclassification rate of 0.1230 and 0.1421, respectively.

6.1.8.5 Accelerated Discriminant Analysis with Mean Threshold Results

The results of ASDA with the implementation of the mean scores of the data as the optimal threshold with 80% of the original training data being the training set and the remaining 20% of the training data being the validation set is detailed. As before, there were two optimal thresholds that were used within this ASDA method. Since the mean scores of the data are being used as the threshold, there is no way to optimize for the smallest misclassification rate or the maximum Youden's Index. Thus, there is only one set of out-of-sample and in-sample results.

Table 6.55: Truth Table for ASDA with Mean Threshold

	0	**1**
0	93	64
1	8	144

The misclassification rate for this ASDA model is 0.2330 with response class 0 and response class 1 having a misclassification rate of 0.0259 and 0.3077, respectively. The in-sample results for this ADSA with mean threshold model is included below.

Table 6.56: Truth Table for ASDA with Mean Threshold (In-Sample)

	0	**1**
0	429	212
1	10	583

The misclassification rate for this model is 0.1799 with response class 0 and response class 1 having a misclassification rate of 0.0081 and 0.2667, respectively.

6.1.8.6 Accelerated Discriminant Analysis with Extended Range of λ and γ Results

Extending the range of λ and γ to match the range of those used above and performing ASDA yielded the results below. The results from the ASDA model that optimizes the threshold for misclassification are below.

Table 6.57: Truth Table for ASDA with Optimal Threshold for Misclassification

	0	**1**
0	68	16
1	38	187

The misclassification rate for this model is 0.1748 with response class 0 and response class 1 having a misclassification rate of 0.3585 and 0.0788, respectively.

The results from the ASDA model that optimizes the threshold for the maximum Youden's Index are below.

Table 6.58: Truth Table for ASDA with Optimal Threshold for Youden's Index

	0	**1**
0	98	55
1	8	148

The misclassification rate for this model is 0.2039 with response class 0 and response class 1 having a misclassification rate of 0.0755 and 0.2709, respectively.

The in-sample results for both of these ASDA with optimal thresholds models are below.

Table 6.59: Truth Table for ASDA with Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	379	111
1	55	689

The in-sample misclassification rate is 0.1345 with response class 0 and response class 1 having a misclassification rate of 0.0446 and 0.1388, respectively.

The misclassification rate for this model is 0.1345 with response class 0 and response class 1 having a misclassification rate of 0.0446 and 0.1388, respectively.

Table 6.60: Truth Table for ASDA with Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	379	111
1	55	689

6.1.8.7 Accelerated Sparse Discriminant Analysis with Mean Threshold Results

Utilizing the mean scores of the data as the threshold yielded the results below. Just as before, since the mean scores of the data is being used as the threshold, then there is no way to optimize for minimum misclassification or maximize for Youden's Index.

Table 6.61: Truth Table for ASDA with Mean Threshold

	0	**1**
0	98	60
1	8	143

The misclassification rate for this model is 0.2201 with response class 0 and response class 1 having a misclassification rate of 0.0755 and 0.2956, respectively.

The in-sample results for the ASDA model with the mean serving as the threshold are below.

Table 6.62: Truth Table for ASDA with Mean Threshold (In-Sample)

	0	**1**
0	421	213
1	13	587

The misclassification rate for this model is 0.1831 with response class 0 and response class 1 having a misclassification rate of 0.0105 and 0.2663, respectively.

6.1.8.8 Accelerated Discriminant Analysis with Ridge Regression and Optimal Threshold Results

ASDA with ridge regression was also conducted on the autism sentiment data with the two optimal thresholds (one for minimizing the misclassifications and one for maximizing Youden's Index) along with the mean scores of the data as a threshold. The results for each are as follows.

Utilizing ASDA with ridge regression with an optimal threshold that minimizes the number of misclassifications yielded the results below.

Table 6.63: Truth Table for ASDA with Ridge Regression and Optimal Threshold for Misclassification

	0	**1**
0	83	27
1	20	179

The misclassification rate for this model is 0.1521 with response class 0 and response class 1 having a misclassification rate of 0.1942 and 0.1311, respectively.

Utilizing ASDA with ridge regression with an optimal threshold that maximizes Youden's Index yielded the results below.

Table 6.64: Truth Table for ASDA with Ridge Regression and Optimal Threshold for Youden's Index

	0	**1**
0	99	54
1	4	152

The misclassification rate for this model is 0.1877 with response class 0 and response class 1 having a misclassification rate of 0.0388 and 0.2621, respectively.

The in-sample results for each of these models are below. Optimizing the threshold for the minimum misclassifications yielded the following in-sample results.

Table 6.65: Truth Table for ASDA with Ridge Regression and Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	346	84
1	91	713

The in-sample misclassification rate for this model is 0.1418 with response class 0 and response class 1 having a misclassification rate of 0.2082 and 0.1054, respectively. Utilizing ASDA with ridge regression with an optimal threshold that maximizes Youden's Index yielded the in-sample results below.

Table 6.66: Truth Table for ASDA with Ridge Regression and Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	405	154
1	32	643

This in-sample model has a misclassification rate of 0.1507 with response class 0 and response class 1 having a misclassification rate of 0.0732 and 0.1942, respectively.

6.1.8.9 Accelerated Discriminant Analysis with Ridge Regression and Mean Threshold Results

The results of ASDA with ridge regression and using the mean scores of the data as the threshold are below.

Table 6.67: Truth Table for ASDA with Ridge Regression and Mean Threshold

	0	**1**
0	100	63
1	3	143

The misclassification rate for this model is 0.2136 with response class 0 and response class 1 having a misclassification rate of 0.0291 and 0.3058, respectively.

The in-sample for ASDA with ridge regression using the mean scores as the threshold results are below.

Table 6.68: Truth Table for ASDA with Ridge Regression and Mean Threshold (In-Sample)

	0	**1**
0	419	213
1	18	584

The in-sample misclassification rate is 0.1872 with response class 0 and response class 1 having a misclassification rate of 0.0412 and 0.2673, respectively.

6.2 Twitter Analysis for Autonomous Car Sentiments

The final illustrative example used in the research was the analysis of Twitter posts dealing with the sentiment of autonomous cars. In this data, users expressed their opinions as to whether or not they agreed with having autonomous cars in operation. The data originally consisted of 7156 tweets or observations which were manually labeled on a scale of 1 to 5 with 1 having an anti-autonomous car sentiment and 5 having a pro-autonomous car sentiment. Each observation labeled as "not relevant" or with a label of 2, 3, or 4 were discarded in this research. This left the data set to contain 567 observations with 2084 predictor variables or features. This data was also partitioned with 80% of the data being used as the training set, and the remaining 20% of the data

being used as the validation set. Following the same procedure as the previous two analyses, the stop words and all punctuation was removed from the data. Then both the training set and the validation set were converted into document term matrices using the dictionary from both the training and validation sets. Once again the rows in the document term matrices represent the observations or tweets, and the columns represent the terms or predictor variables. Thus, the training set is a data matrix of 453×2084, and the validation set is a data matrix of 114×2084. Sparse logistic regression with an optimal and mean threshold and accelerated discriminant analysis were conducted on this data.

6.2.1 Sparse Logistic Regression Results

Conducting the same procedures as used with the above analyses on the autism data and the global warming data, sparse logistic regression with 100-fold cross-validation was performed to obtain a ridge regression and LASSO model. This data has already been partitioned so that 80% of the training data is used as the training set, and the remaining 20% of the training data is used as the validation set. The results of these models are below.

Table 6.69: Truth Table for Ridge Regression

	0	**1**
0	0	0
1	18	96

This ridge regression model has a misclassification rate of 0.1579 with response class 0 and response class 1 misclassification rate having a misclassification rate of 1 and 0, respectively.

This LASSO model has a misclassification rate of 0.1579 with response class 0 and response class 1 misclassification rate having a misclassification rate of 1 and 0, respectively.

Table 6.70: Truth Table for LASSO

	0	**1**
0	0	0
1	18	96

The in-sample results for these ridge regression and LASSO models are below.

Table 6.71: Truth Table for Ridge Regression (In-Sample)

	0	**1**
0	11	0
1	80	362

This in-sample ridge regression model has a misclassification rate of 0.1766 with response class 0 and response class 1 misclassification rate having a misclassification rate of 0.8791 and 0, respectively.

Table 6.72: Truth Table for LASSO (In-Sample)

	0	**1**
0	14	0
1	77	362

This in-sample LASSO model has a misclassification rate of 0.1700 with response class 0 and response class 1 misclassification rate having a misclassification rate of 0.8462 and 0, respectively.

6.2.2 Sparse Logistic Regression with Optimal Threshold Results

Just as with the autism sentiment data set, sparse logistic regression with two optimal thresholds, one for minimizing the number of misclassifications, and the other for maximizing Youden's Index, were conducted. The results are below. The threshold that minimizes the number of misclassifications in the model yielded the results that follow.

Table 6.73: Truth Table for Ridge Regression with Optimal Threshold for Misclassification

	0	**1**
0	7	4
1	11	92

The misclassification rate for this model is 0.1316 with response class 0 and response class 1 having a misclassification rate of 0.6111 and 0.0417, respectively.

Table 6.74: Truth Table for LASSO with Optimal Threshold for Misclassification

	0	**1**
0	8	4
1	10	92

The misclassification rate for this model is 0.1228 with response class 0 and response class 1 having a misclassification rate of 0.5556 and 0.0417, respectively.

The results for sparse logistic regression using the optimal threshold that maximizes Youden's Index is below.

Table 6.75: Truth Table for Ridge Regression with Optimal Threshold for Youden's Index

	0	**1**
0	12	10
1	6	86

The misclassification rate for this model is 0.1404 with response class 0 and response class 1 having a misclassification rate of 0.3333 and 0.1042, respectively.

Table 6.76: Truth Table for LASSO with Optimal Threshold for Youden's Index

	0	**1**
0	8	4
1	10	92

The misclassification rate for this model is 0.1228 with response class 0 and response class 1 having a misclassification rate of 0.5556 and 0.0417, respectively.

The in-sample results for both of these models are included below. The results for the threshold used to minimize the number of misclassifications is below.

Table 6.77: Truth Table for Ridge Regression with Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	90	0
1	1	362

The misclassification rate for this model is 0.0022 with response class 0 and response class 1 having a misclassification rate of 0.0110 and 0, respectively.

Table 6.78: Truth Table for LASSO with Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	63	7
1	28	355

The misclassification rate for this model is 0.0773 with response class 0 and response class 1 having a misclassification rate of 0.3077 and 0.0193, respectively.

The results for the in-sample sparse logistic regression using the optimal threshold that maximizes Youden's Index is below.

Table 6.79: Truth Table for Ridge Regression with Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	90	0
1	1	362

The misclassification rate for this model is 0.0022 with response class 0 and response class 1 having a misclassification rate of 0.0110 and 0, respectively.

Table 6.80: Truth Table for LASSO with Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	70	21
1	21	341

The misclassification rate for this model is 0.0927 with response class 0 and response class 1 having a misclassification rate of 0.2308 and 0.0580, respectively.

6.2.2.1 Sparse Logistic Regression with Mean Threshold Results

Using the mean scores of the data as the threshold for sparse logistic regression yielded the following results. Since the mean scores of the data are being used as the threshold, there is no way to optimize for the smallest misclassification rate or the maximum Youden's Index. Thus, there is only one set of out-of-sample and in-sample results.

Table 6.81: Truth Table for Ridge Regression with Mean Threshold

	0	**1**
0	14	27
1	4	69

The misclassification rate for this model is 0.2719 with response class 0 and response class 1 having a misclassification rate of 0.2222 and 0.2813, respectively.

Table 6.82: Truth Table for LASSO with Mean Threshold

	0	**1**
0	9	15
1	9	81

The misclassification rate for this model is 0.2105 with response class 0 and response class 1 having a misclassification rate of 0.5000 and 0.1563, respectively.

The in-sample results for both of these models are included below.

Table 6.83: Truth Table for Ridge Regression with Mean Threshold (In-Sample)

	0	**1**
0	91	4
1	0	358

The misclassification rate for this model is 0.0088 with response class 0 and response class 1 having a misclassification rate of 0 and 0.0110, respectively.

Table 6.84: Truth Table for LASSO with Mean Threshold (In-Sample)

	0	**1**
0	70	21
1	21	341

The misclassification rate for this model is 0.0927 with response class 0 and response class 1 having a misclassification rate of 0.2308 and 0.0580, respectively.

6.2.3 Accelerated Discriminant Analysis Results

ASDA was also performed on this self-driving data with the goal of improving the overall misclassification rate of the models and to balance the number of misclassifications occurring in each of the response classes. ASDA with an optimal threshold, ASDA with the mean scores of the data being used as the threshold, ASDA with an extended range of λ and γ, and ASDA with ridge regression implemented were all utilized for this data. The results for each method is detailed in the following subsections.

6.2.3.1 Accelerated Discriminant Analysis with Optimal Threshold Results

In this sub-subsection, the results of ASDA with the implementation of an optimal threshold with 80% of the data being the training set and the remaining 20% of the data being the validation set is detailed. There were two optimal thresholds that were used within this ASDA method. The first threshold optimizes for the minimum misclassification error of the model. The second optimal threshold yields the maximum Youden's Index. The results for ASDA using the optimal threshold that minimizes the number of misclassifications is below.

Table 6.85: Truth Table for ASDA with Optimal Threshold for Misclassification

	0	**1**
0	5	2
1	13	94

The misclassification rate for this model is 0.1316 with response class 0 and re-

sponse class 1 having a misclassification rate of 0.7222 and 0.0208, respectively.

The results for ASDA using the optimal threshold that maximizes Youden's Index is below.

Table 6.86: Truth Table for ASDA with Optimal Threshold for Youden's Index

	0	**1**
0	13	27
1	5	69

The misclassification rate for this model is 0.2807 with response class 0 and response class 1 having a misclassification rate of 0.2778 and 0.2813, respectively.

The in-sample results for both of these models are included below.

Table 6.87: Truth Table for ASDA with Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	59	9
1	32	353

The misclassification rate for this model is 0.0905 with response class 0 and response class 1 having a misclassification rate of 0.3516 and 0.0249, respectively.

The results for ASDA using the optimal threshold that maximizes Youden's Index is below.

Table 6.88: Truth Table for ASDA with Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	72	48
1	19	314

The misclassification rate for this model is 0.1479 with response class 0 and response class 1 having a misclassification rate of 0.2088 and 0.1326, respectively.

6.2.3.2 Accelerated Discriminant Analysis with Mean Threshold Results

The results of ASDA with the implementation of the mean scores of the data as the optimal threshold with 80% of the data being the training set and the remaining 20% of the data being the validation set is detailed below. As before, there were two optimal thresholds that were used within this ASDA method, one to minimize the number of misclassifications and another to maximize Youden's Index. Since the mean scores of the data are being used as the threshold, there is no way to optimize for the smallest misclassification rate or the maximum Youden's Index. Thus, there is only one set of out-of-sample and in-sample results.

Table 6.89: Truth Table for ASDA with Mean Threshold

	0	**1**
0	13	35
1	5	61

The misclassification rate for this model is 0.3509 with response class 0 and response class 1 having a misclassification rate of 0.2778 and 0.3646, respectively.

The in-sample results for both of these models are included below.

Table 6.90: Truth Table for ASDA with Mean Threshold (In-Sample)

	0	**1**
0	81	97
1	10	265

The misclassification rate for this model is 0.2362 with response class 0 and response class 1 having a misclassification rate of 0.1099 and 0.2680, respectively.

6.2.3.3 Accelerated Discriminant Analysis with Extended Range of λ and γ Results

Extending the range of λ and γ to match the range of those used for the autism data and performing ASDA with optimal thresholds that minimize the number of misclassifications for the model and that maximizes Youden's Index yielded the results

below. The following results are for the extended ASDA with an optimal threshold that optimizes for the minimum number of misclassifications.

Table 6.91: Truth Table for ASDA with Optimal Threshold for Misclassification

	0	**1**
0	0	1
1	19	94

The misclassification rate for this model is 0.1754 with response class 0 and response class 1 having a misclassification rate of 1 and 0.0105, respectively.

The results for the extended ASDA model which an optimal threshold that maximizes Youden's Index are below.

Table 6.92: Truth Table for ASDA with Optimal Threshold for Youden's Index

	0	**1**
0	0	1
1	19	94

The misclassification rate for this model is 0.1754 with response class 0 and response class 1 having a misclassification rate of 1 and 0.0105, respectively.

The in-sample results for the both optimal thresholds are below.

Table 6.93: Truth Table for ASDA with Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	0	1
1	90	362

The misclassification rate for this model is 0.2009 with response class 0 and response class 1 having a misclassification rate of 1 and 0.0028, respectively.

Table 6.94: Truth Table for ASDA with Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	0	1
1	90	362

The misclassification rate for this model is 0.2009 with response class 0 and response class 1 having a misclassification rate of 1 and 0.0028, respectively.

6.2.3.4 Extended ASDA with Mean Threshold Results

Utilizing the mean scores as the threshold yielded the results below.

Table 6.95: Truth Table for ASDA with Mean Threshold

	0	**1**
0	15	38
1	5	56

The misclassification rate for this model is 0.3772 with response class 0 and response class 1 having a misclassification rate of 0.2500 and 0.4043, respectively.

The in-sample results for the mean threshold are below.

The misclassification rate for this model is 0.1700 with response class 0 and response class 1 having a misclassification rate of 0 and 0.2115, respectively.

Table 6.96: Truth Table for ASDA with Mean Threshold (In-Sample)

	0	**1**
0	89	77
1	0	287

6.2.3.5 Accelerated Discriminant Analysis with Ridge Regression and Optimal Threshold Results

ASDA with ridge regression was also conducted on the autonomous cars sentiment data with two optimal thresholds and the mean scores of the data as a threshold. The results for each are as follows. Utilizing ASDA with ridge regression with an optimal threshold that minimizes the number of misclassifications yielded the results below.

Table 6.97: Truth Table for ASDA with Ridge Regression and Optimal Threshold for Misclassification

	0	**1**
0	0	1
1	21	92

The misclassification rate for this model is 0.1930 with response class 0 and response class 1 having a misclassification rate of 1 and 0.0108, respectively.

Utilizing ASDA with ridge regression with an optimal threshold that maximizes Youden's Index yielded the results below.

Table 6.98: Truth Table for ASDA with Ridge Regression and Optimal Threshold for Youden's Index

	0	**1**
0	0	1
1	21	92

The misclassification rate for this model is 0.1930 with response class 0 and response class 1 having a misclassification rate of 1 and 0.0108, respectively.

The in-sample results for each of the optimal thresholds are below.

Table 6.99: Truth Table for ASDA with Ridge Regression and Optimal Threshold for Misclassification (In-Sample)

	0	**1**
0	0	1
1	88	364

The in-sample misclassification rate for this model is 0.1965 with response class 0 and response class 1 having a misclassification rate of 1 and 0.0027, respectively.

Utilizing ASDA with ridge regression with an optimal threshold that maximizes Youden's Index yielded the results below.

Table 6.100: Truth Table for ASDA with Ridge Regression and Optimal Threshold for Youden's Index (In-Sample)

	0	**1**
0	0	1
1	88	364

This in-sample model has a misclassification rate of 0.1965 with response class 0 and response class 1 having a misclassification rate of 1 and 0.0027, respectively.

6.2.3.6 Accelerated Discriminant Analysis with Ridge Regression and Mean Threshold Results

The results of ASDA with ridge regression and the mean scores representing the threshold are below.

Table 6.101: Truth Table for ASDA with Ridge Regression and Mean Threshold

	0	**1**
0	13	23
1	4	74

The misclassification rate for this model is 0.2368 with response class 0 and response class 1 having a misclassification rate of 0.2353 and 0.2371, respectively.

The in-sample results of this ASDA model are below.

Table 6.102: Truth Table for ASDA with Ridge Regression and Mean Threshold (In-Sample)

	0	**1**
0	92	55
1	0	306

The in-sample misclassification rate is 0.1214 with response class 0 and response class 1 having a misclassification rate of 0 and 0.1524, respectively.

6.3 Summary of Results for Each Model

The following tables provide the misclassification rates for each analysis technique. Both out-of-sample misclassification rates and in-sample misclassification rates are in the following tables. This table shows the misclassification rates for all of the out-of-sample models.

This next table shows all of the in-sample misclassification rates for each of the models discussed.

Table 6.103: Misclassification Rates for All Models

	Autism	Autonomous Cars
Ridge Regression	0.3766	
LASSO	0.3027	
Ridge Regression with Optimal Threshold	0.1834	
Ridge Regression with Mean Threshold	0.1872	
LASSO with Optimal Threshold	0.2731	
LASSO with Mean Threshold	0.2698	
ASDA with Optimal Threshold	0.2545	
ASDA with Mean Threshold	0.2616	
Extended ASDA with Optimal Threshold	0.2666	
Extended ASDA with Mean Threshold	0.2616	
ASDA with Ridge Regression and Optimal Threshold	0.2540	
ASDA with Ridge Regression and Mean Threshold	0.2600	
	80/20 Partitioning of Data	
Ridge Regression	0.2783	0.1579
LASSO	0.1553	0.1579
Ridge Regression with Optimal Threshold for Misclassification	0.1197	0.1316
LASSO with Optimal Threshold for Misclassification	0.1230	0.1228
Ridge Regression with Optimal Threshold for Youden's Index	0.1197	0.1404
LASSO with Optimal Threshold for Youden's Index	0.1230	0.1228
Ridge Regression with Mean Threshold	0.1262	0.2719
LASSO with Mean Threshold	0.1618	0.2105
ASDA with Optimal Threshold for Misclassification	0.1489	0.1316
ASDA with Optimal Threshold for Youden's Index	0.1553	0.2807
ASDA with Mean Threshold	0.2330	0.3509
Extended ASDA with Optimal Threshold for Misclassification	0.1748	0.1754
Extended ASDA with Optimal Threshold for Youden's Index	0.2039	0.1754
Extended ASDA with Mean Threshold	0.2201	0.3772
ASDA with Ridge Regression and Optimal Threshold for Misclassification	0.1521	0.1930
ASDA with Ridge Regression and Optimal Threshold for Youden's Index	0.1877	0.1930
ASDA with Ridge Regression and Mean Threshold	0.2136	0.2368

Table 6.104: Misclassification Rates for All In-Sample Models

	Autism	Autonomous Cars
Ridge Regression	0.3766	
LASSO	0.3027	
Ridge Regression with Optimal Threshold	0.0032	
Ridge Regression with Mean Threshold	0.0078	
LASSO with Optimal Threshold	0.1005	
LASSO with Mean Threshold	0.0752	
ASDA with Optimal Threshold	0.1426	
ASDA with Mean Threshold	0.1925	
Extended ASDA with Optimal Threshold	0.1717	
Extended ASDA with Mean Threshold	0.1426	
ASDA with Ridge Regression and Optimal Threshold	0.1432	
ASDA with Ridge Regression and Mean Threshold	0.1918	
	80/20 Partitioning of Data	
Ridge Regression	0.0608	0.1766
LASSO	0.0843	0.1700
Ridge Regression with Optimal Threshold for Misclassification	0.0032	0.0022
LASSO with Optimal Threshold for Misclassification	0.0511	0.0773
Ridge Regression with Optimal Threshold for Youden's Index	0.0032	0.0022
LASSO with Optimal Threshold for Youden's Index	0.0511	0.0927
Ridge Regression with Mean Threshold	0.0065	0.0088
LASSO with Mean Threshold	0.0956	0.0927
ASDA with Optimal Threshold for Misclassification	0.1337	0.0905
ASDA with Optimal Threshold for Youden's Index	0.1353	0.1479
ASDA with Mean Threshold	0.1799	0.2362
Extended ASDA with Optimal Threshold for Misclassification	0.1345	0.2009
Extended ASDA with Optimal Threshold for Youden's Index	0.1345	0.2009
Extended ASDA with Mean Threshold	0.1831	0.1700
ASDA with Ridge Regression and Optimal Threshold for Misclassification	0.1418	0.1965
ASDA with Ridge Regression and Optimal Threshold for Youden's Index	0.1507	0.1965
ASDA with Ridge Regression and Mean Threshold	0.1872	0.1214

CONCLUSIONS

The overall goal for this research was to accurately classify textual data into one of two categories based on the sentiment that was expressed by Twitter users. Each data set had far more predictor variables or features than observations, which causes linear regression and logistic regression to overfit the data due to degeneracy.

In order to combat this issue, regularization was added to the logistic regression model in the form of ridge regression and LASSO. Due to the misclassification rate not being as low as expected and the threshold or cutoff for classifying the predicted probabilities into the appropriate response classes not appearing to be 0.5, which is the standard threshold for logistic regression, the threshold was adjusted. This threshold is the value at which the data is classified into one of the two categories. In other words, if the probability of the data is below this threshold, then it will be placed into response class 0, and if it is above this threshold, then it will be classified into response class 1. As previously stated, R was used to conduct this research, and it has an optimalCutoff command that is part of the Information Value package, which computes the optimal threshold based on the data. This optimalCutoff command allows the user to choose the optimal threshold that will optimize the model for the minimum number of misclassifications or for the maximum Youden's Index. However, this command uses the actual labels of data to compute the optimal threshold. In most cases the actual labels of the observations may not be known, and therefore, the mean of the predicted scores of the data was used as the threshold. This is what is referred to as the mean threshold within this research. This technique provided results comparable to those of the optimal threshold, and should therefore be used if the actual labels of the observations are not

known.

Linear discriminant analysis (LDA) can also be used to classify data into two response categories. LDA is especially useful when working with higher dimension data, as LDA performs dimension reduction on the data and projects it onto one of two response classes. Accelerated Sparse Discriminant Analysis (ASDA), a regularized form of LDA, was used to aid in the classification of this textual data. As with the above techniques, the goal is to accurately classify as many of the observations as possible, while also maintaining a small misclassification rate for each response class. Just as with sparse logistic regression, the optimal threshold, the mean threshold, ridge regression, and LASSO were incorporated into the ASDA component of this analysis.

Lastly, in an attempt to receive even lower misclassification rates for the models and each response class, the training data was partitioned into training and validation sets, and sparse logistic regression and ASDA were conducted on these partitions to improve the models.

7.1 Autism Sentiments Results

For the data about autism and vaccination sentiments, the training and testing data was comprised of the same set of tweets manually labeled by two individuals. One individual's labeling of the tweets were used as the training data, and the other individual's labeling of the tweets were used as the testing data. This preparation of the data poses potential issues with the classification of the tweets. Some of the misclassifications within the models of the data could be due to not having the same tweets share the same labels. For instance, one person could have labeled the first tweet as having an anti-vaccination sentiment, while the other person could have labeled the same tweet as having a pro-vaccination sentiment. Due to the contrary actual labeling of this tweet, this would cause some errors in the machine learning process. Hence, there would be an increase in the number of misclassifications, and thus raising the misclassification rate

of the model. Another issue with this data set was that there far more anti-vaccination sentiments (1) than there were pro-vaccination sentiments (0). The training data, which contained a total of 1543 observations, had a total of 1003 of those actually labeled with an anti-vaccination sentiment or class 1 observations. This leaves only 540 actual class 0 observations, which is roughly half of the class 1 observations. The testing data, which was comprised of 1827 tweets had a total of 1104 anti-vaccination sentiment tweets or class 1 observations. Hence, there were only 723 class 0 observations, which is a little more than half of the class 1 observations. Due to this, the machine learning aspect of this research does a fairly decent job at classifying those tweets that belong to the anti-vaccination sentiment category, response class 1, but it does a poorer job at classifying the tweets with a pro-vaccination sentiment, response class 0. Since there are more anti-vaccination tweets in both the training and testing data sets, if the machine is not sure how to classify the tweets, then the tweets are usually classified into the anti-vaccination sentiment category, which is usually accurate since there are more tweets sharing this sentiment. Due to the large number of anti-vaccination sentiment tweets, the analysis techniques perform quite well with this class of data. On the contrary, the analysis performs relatively poorly with pro-vaccination sentiment tweets because they are being labeled as anti-vaccination sentiment tweets due to the imbalance in the data.

For the autism data, the lowest rate of misclassification was observed when the training data was partitioned with 80% being the training set and the remaining 20% of the data being the validation set and sparse logistic regression using the l_2-penalty with an optimal threshold was performed. This technique yielded a misclassification rate of 0.1197. The same number of misclassifications were calculated regardless if the optimal threshold was optimized for the minimum number of misclassifications for the entire model or the maximum of Youden's Index. Recall that ridge regression uses all of the features in the model, thus this model may not be too interpretable, but utilizing the l_1-penalty with the same partitioning of the data and an optimal threshold

produced comparable results. The LASSO model with an optimal threshold yielded a misclassification rate of 0.1230, regardless of how the threshold was optimized. This model has a slightly larger misclassification rate, but it is a more interpretable model. Also, by only using the training data and partitioning it into training and validation sets, the possibility of having contrary labeling of tweets is no longer an issue. Partitioning this data in the above manner and performing the same sparse techniques produced fewer errors than the models that were not partitioned and consisted of the 1543 observation training set and 1827 observation testing set. This verifies that there were some issues with contrary labeling of the two sets of data by the individuals.

Of the accelerated sparse discriminant analysis methods, ASDA with ridge regression and an optimal threshold for misclassification performed the best on the autism data that was manually labeled by two individuals. It should be noted that this optimal threshold was optimized to produce the lowest number of misclassifications for the entire model. This model yielded a misclassification rate of 0.2540 with response class 0 and response class 1 having a misclassification rate of 0.3914 and 0.1639, respectively. The accelerated sparse discriminant analysis model that performed the best on the autism data that was partitioned with an 80/20-split was the model that implemented an optimal threshold that minimized the amount of misclassifications for the model. This technique yielded a misclassification rate 0.1489 with response class 0 and response class 1 having misclassification rates of 0.2475 and 0.1010, respectively.

As mentioned above, the fact that the partitioned set of the autism data outperformed the data with both training and testing data sets implies our concerns about the potential issues with contrary labeling were valid. Also, the methods which utilized the optimal thresholds produced smaller misclassification rates than the same method with no optimal threshold employed. However, using the mean of the scores of the data as the threshold performs more poorly, as to be expected, since the optimal thresholds are computed using the actual labels of the observations. Using the means as the threshold

also performs more poorly due to the imbalance of the data. Since there are far more tweets in response class 1 than tweets in response class 0, then the mean of the scores of the data will be closer to 1, which will make the threshold higher, and hence increases the amount of tweets that will be classified as response class 1. And as expected, due to the imbalance of tweets in each response class, response class 1 has the lower misclassification rates in most cases for the autism data.

7.2 Autonomous Car Sentiments Results

With the data about autonomous car sentiments, users were asked to express their opinions as to whether or not they agreed with autonomous cars being used. Those who shared the sentiment of having self-driving cars were classified into response category 1, and those who expressed the sentiment of not agreeing with having self-driving cars were classified into response class 0. This data contained a total of 567 pro-autonomous car sentiments and anti-autonomous car sentiments. Of these 567 observations, 458 of those had a pro-autonomous car sentiment, which left only 109 observations to have an anti-autonomous car sentiment. Just as with the autism data set, this set of data is extremely unbalanced which could lead to potential problems when trying to accurately classify the tweets. In fact, much like the previous set of data, response class 1 is predicted better than response class 0 due to the large imbalance of the data.

Because the autonomous car data only consisted of one set of tweets, the data was partitioned into an 80/20-split with 80% of the data being used as the training set and the remaining 20% of the data being used as the validation set. The autonomous car data performed extremely well with sparse logistic regression with the l_2-regularization and an optimal threshold to optimize the minimum number of misclassifications for the model. This technique yielded a misclassification rate of 0.1316 with response class 0 and response class 1 having a misclassification rate of 0.6111 and 0.0417, respectively. It can be seen that this model did great at identifying and labeling the observations that

belonged to response class 1, but not so well with identifying observations that belonged to response class 0. The sparse logistic regression with the l_1-regularization penalty and an optimal threshold to optimize the minimum number of misclassifications for the model performed even better with a misclassification rate of 0.1228, but the misclassification rates within the response classes were also extremely unbalanced. Taking into account that the actual response classes of the observations were used in computing this optimal threshold, the mean value of the predicted scores of the data was replaced as the threshold and performed with the implementation of the l_1-penalty and l_2-penalty for sparse logistic regression. The LASSO model with a mean threshold performed better with a misclassification rate of 0.2105 compared to ridge regression with a mean threshold, which had a misclassification rate of 0.2719. This highlights the significance of adjusting the threshold, as the models with an optimal threshold outperformed the models with the standard threshold of 0.5. On the other hand, using the mean values of the predicted scores of the data did poorer in terms of misclassification rates. This could be due to the heavy imbalance of response class 1 observations. Because of the majority of the data has scores greater than 0.5 when the mean of these values are used as the threshold, this forces the model to have a larger threshold than necessary, which leads to larger misclassification rates.

The accelerated sparse discriminant analysis with an optimal threshold that minimized the number of misclassifications for the entire model did the best with this self-driving car data. The method yielded a misclassification rate of 0.1316 with response class 0 and response class 1 having a misclassification rate of 0.7222 and 0.0208, respectively. Once again there is an imbalance in the accuracy of classification among the response classes; therefore, the accelerated sparse discriminant analysis model with an extended range of γ and λ and an optimal threshold for Youden's Index may be the better model since is has a more balanced misclassification rate among the classes. This model yielded a misclassification rate of 0.1754 with response class 0 and response class

1 having a misclassification rate of 1 and 0.0105, respectively. However, recall that the optimal threshold calculated uses the actual labels of the observations, and this information may not be known in real-world situations. Thus, having the mean values of the scores of the data represent the threshold for the accelerated discriminant analysis method produced a misclassification rate of 0.3509 with response class 0 and response class 1 having a misclassification rate of 0.2778 and 0.3646, respectively. Despite the potential issues that this data posed, some version of both sparse logistic regression and accelerated discriminant analysis performed well.

7.3 Overall Results

The two data sets used in this research both pose potential issues due to the nature of the data. For both sets of data, each tweet or observation only contains a very small subset of the dictionary of terms or the predictor variables due to the 140-character limit per tweet. This means that some tweets could be classified based on a single term, which makes the classification of each tweet more difficult.

Also, with both sets of data, there were far more predictor variables than observations, which hindered the performance of linear regression and logistic regression due to degeneracy. These two regression techniques suffered from overfitting the data, which caused the number of misclassified response class labels to outnumber the amount of correct labels. Therefore, sparse logistic regression with an l_1-penalty and l_2-penalty were utilized in an attempt to more accurately classify the observations in the data. The implementation of the l_1-penalty, or LASSO, provides some important characteristics, as it uses variable selection by shrinking some of the model coefficients to zero when modeling the data, which makes the model easier to interpret. The l_1-penalty also produces a feature list, which will tell which terms in the dictionary of the data most closely relates to each sentiment. This helps with the interpretability of the model. However, this penalty sometimes has a larger misclassification rate than ridge regression, which

uses the l_2-penalty. With ridge regression, all of the model coefficients are used, which can make the interpretation of the model more difficult.

Because this research focuses on correctly labeling or classifying as many of the observations as possible, while also providing a balance of the number of misclassifications within the response classes, sparse logistic regression with an optimal threshold was employed. This technique allowed the threshold or cutoff value in logistic regression to be optimized for either providing the minimum number of misclassifications for the model or providing a maximum value for Youden's Index, which ensures that there is a balance in the number of misclassifications among the response classes. Since the optimal threshold uses the actual labels of the data to provide such results, the mean values of the predicted scores of the data was also utilized as the threshold to compare with the optimal threshold results. In both sets of data, using the mean scores as the threshold did not do as well as using one of the two optimal thresholds, but this is expected since the data was unbalanced and since the optimal threshold is computed using the actual response classes of the tweets.

As an improvement due to the high dimensions in the two data sets, a sparse version of linear discriminant analysis, accelerated sparse discriminant analysis, was used to model the data. Linear discriminant analysis performs dimension reduction on the data and projects it onto the two response classes. However, this technique can have issues when there are far more predictors than observations, which is the case with the two data sets that are used in this dissertation. Thus, accelerated sparse discriminant analysis was used to combat some of these potential issues. In this research, accelerated sparse discriminant analysis was used to develop a model that is easier to interpret and reduce overfitting of the training data. Along with this method of analysis, an l_2-penalty and an optimal (and mean) threshold were implemented to provide better results.

Although the analyses presented in this research did quite well, it should be mentioned that there are still some potential issues when analyzing social media posts,

specifically Twitter posts. When performing this type of textual analysis on Twitter posts, it is imperative to note that the tone of the text poses a potential issue. The machine may not be able to detect such tones such as sarcasm, but manual labeling of the tweets would detect such and this could lead to errors in the analysis. Also, Twitter has a command called "Retweet", which allows users to copy another user's tweet and add or delete content from it. This could be an issue since two users who may have differing sentiments may have a Twitter posts that only differ by one word. Since the dictionary of the terms in the data is such a small subset of terms, then each term is very significant in classifying each tweet. Thus, having the same tweet with the exception of one word could lead to errors in classifying the data.

7.4 Future Work

Improving the overall classification rate for the models can possibly be achieved by incorporating the following recommendations. To begin with, when performing accelerated sparse discriminant analysis for the autonomous car data, the same λ,γ-pairs were used that were used for the autism data. To get better results for the ASDA methods, training λ,γ-pairs specific for this data may improve the classification of the data. Also, since ridge regression and LASSO performed well, incorporating an elastic-net penalty to each method may lower the misclassification rates for each model since the elastic-net penalty is a combination of ridge regression and LASSO.

To provide better analysis of textual data of this form, specifically Twitter posts, employing a method of allowing the document term matrix to consist of phrases or n-grams rather than single terms will be considered. This will aid in captivating the sentiment of the tweet or the text. For instance, with the document term matrix only consisting of single terms, the techniques only acknowledges if the terms appear in the observation; however, the placement of these terms are not considered. In the case of the words "do" and "not", the machine cannot distinguish if these two words are placed side

by side to create the phrase "do not", or if these two words are just present in the text. Also, allowing the document term matrix to to be binary rather than contain frequencies of each term may aid in the classification processes. It may be more significant to indicate whether or not a term appears, rather than the number of times that the term appears in each observation.

Another improvement that will be considered for future work on this classification research is to find better ways to fine tune the optimal threshold that does not involve using the actual classification labels, as well as applying an ordinal version of ASDA that will add the order of sentiment as a feature in addition to the terms. To combat the issue of the imbalance of data, resampling will be employed with each set of data.

Each of these plans will be incorporated in an effort to create a better balance between the misclassification errors of each response class while achieving the smallest overall number of errors for the models. To confirm the validity of the results, these techniques will be applied to more data sets to ensure that the results are comparable.

REFERENCES

[1] Atkins, S., Gudmundur, E., Ames, B., & Clemmensen, L. Proximal methods for sparse optimal scoring and discriminant analysis.

[2] Beck, A. & Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1): 183-202, 2009.

[3] Chapter 12: Logistic Regression. (n.d.). Retrieved from `http://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf`

[4] Clemmensen, L., Hastie, T., Witten, D., & Erboll, B. Sparse discriminant analysis. *Technometrics*, 53(4), 2011.

[5] Differences between L1 and L2 as Loss Function and Regularization. (2013, December 18). Retrieved from `http://www.chioka.in/differences-between-l1-and-l2-as-loss-function-and-regularization/`

[6] Friedman, J., Hastie, T., Simon, N., & Tibshirani, R. (n.d.). Glmnet Package (Version 2.0-10) [Program documentation].

[7] Friedman, J., Hastie, T., & Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. 2009.

[8] Hastie, T., & Qian, J. (2016, September 13). Glmnet Vignette. Retrieved from `https://web.stanford.edu/~hastie/glmnet/glmnet_beta.html#log`

[9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

[10] Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical Learning with sparsity: The Lasso and Generalizations*. Boca Raton: CRC Press, Taylor Francis Group.

[11] James, G., Witten, D., & Hastie, T. (2013). *An Introduction to Statistical Learning with Applications in R*. New York: Springer.

[12] Kuhn, M. & Johnson, K. (2013). *Applied Predictive Modeling*. New York: Springer

[13] Kutner, M.H., Nachtsheim, C., & Neter, J. (2004). *Applied Linear Regression Models*. Boston: McGraw-Hill/Irwin.

[14] Lecture 15: Linear Discriminant Analysis. (n.d.). Retrieved from `https://www.doc.ic.ac.uk/~dfg/ProbabilisticInference/old_IDAPILecture15.pdf`

[15] Lesson 10.3: Linear Discriminant Analysis. (n.d.). Retrieved from `https://onlinecourses.science.psu.edu/stat505/node/94/`

[16] Linear Discriminant Analysis. (n.d.). Retrieved from `http://www.saedsayad.com/lda.htm`

[17] Pang, B., & Lee, L. Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the conference on empirical methods in natural language processing*. 2002.

[18] Prabhakaran, S. 2016, October 29). InformationValue: Performance Analysis and Companion Functions for Binary Classification Models. [Program documentation].

[19] Tomeny, T., Vargo, C., & El-Toukhy, S. Geographic and Demographic Correlates of Autism-Related Anti-Vaccine Beliefs on Twitter. 2009.

[20] Welling, M. (n.d.). Fisher Linear Discriminant Analysis. Retrieved from `https://www.ics.uci.edu/~welling/teaching/273ASpring09/Fisher-LDA.pdf`.