Journal of Behavioral Data Science, 2025, *5* (1), 103–147.
DOI: https://doi.org/10.35566/jbds/bain

A Tutorial on Supervised Machine Learning Variable Selection Methods in Classification for the Social and Health Sciences in R

Catherine M. Bain[1 [0000−0002−2767−6882]], Dingjing Shi[1,2 [0000−0002−5652−3818]], Yaser M. Banad[3 [0000−0001−7339−810X]], Lauren E. Ethridge[1,4 [0000−0003−0601−6911]], Jordan E. Norris[1 [0000−0002−4438−3416]], and Jordan E. Loeffelman[1 [0000−0002−0269−7708]]

[1] Department of Psychology, University of Oklahoma, Norman, OK, USA
cbain1@ou.edu
[2] School of Psychology, Georgia Institute of Technology, Atlanta, GA, USA
[3] School of Electrical and Computer Engineering, University of Oklahoma, Norman, OK, USA
[4] Department of Pediatrics, Section on Developmental and Behavioral Pediatrics, University of Oklahoma Health Sciences Center, Oklahoma City, OK, USA

Abstract. With the increasing availability of large datasets in the behavioral and health sciences, the need for efficient and effective variable selection techniques has grown. While traditional methods like stepwise regression remain prevalent, numerous advanced techniques are available but underutilized in these fields. This tutorial aims to increase awareness and understanding of five variable selection methods available in the popular statistical software R: LASSO, Elastic Net, a penalized SVM classifier, random forest, and the genetic algorithm. Using a recent survey-based assessment dataset on misophonia diagnosis, we provide step-by-step guidance on variables selections and implementation of each method in the context of classification. We discuss the strengths, weaknesses, and performance of each technique, emphasizing the importance of selecting appropriate performance metrics. The associated code and data implemented in this tutorial are available on Open Science Framework and provide an interactive learning experience. We encourage social and health science researchers to adopt these advanced variable selection methods, leading to more robust, interpretable, and impactful models. This paper is written with the assumption that individuals have at least a basic understanding of R.

Keywords: Machine learning · Variable selection · Big data · R · Data classification

1 Introduction

In the behavioral and health sciences, selecting the right variables for a model is crucial for understanding human behavior's complexity. Researchers strive to uncover how personality traits influence treatment engagement, how symptoms manifest in disorders, and how to accurately classify individuals into meaningful groups for diagnosis or intervention. They not only want to understand how these aspects (i.e., variables) are related to each other and to overarching constructs but may also want to use the variables to classify individuals into groups (e.g., diagnosing clinical disorders, determining participant compliance, etc.). The accuracy of these classifications or predictions is greatly influenced by which variables a researcher uses to create the classifications. For example, if a researcher is interested in diagnosing someone with depression, the accuracy of the diagnosis would suffer if relying solely on the presence of a depressed mood. However, if they use a variety of variables like depressed mood, loss of interest in activities, hours slept, and change in appetite or weight, their classification would be more accurate.

Researchers must carefully construct their classification models to understand variable interrelationships while maximizing predictive accuracy. Variable selection techniques can help researchers to identify and select informative variables to build these models. The use of variable selection techniques can lead to more accurate predictions, reduce the computational cost of creating the model, and improve the parsimony of the model by eliminating redundant and irrelevant variables. For example, variable selection techniques have been used to build models pertaining to identifying exposure-outcome associations (Lenters, Vermeulen, & Portengen, 2018) as well as predicting mortality rates (Amene, Hanson, Zahn, Wild, & Döpfer, 2016; Bourdès et al., 2010), psychological strain in teachers (Wettstein et al., 2023), and nomophobia (Luo, Ren, Li, & Liu, 2021).

Behavioral researchers often turn to stepwise regression to perform variable selection. An APA PsychINFO database search for the term "stepwise regression" returned 222 peer-reviewed articles published in the last 3 years using stepwise regression for variable selection. Stepwise regression, however, has many severe limitations and statistical experts do not recommend it (Smith, 2018; Thompson, 1995; Whittingham, Stephens, Bradbury, & Freckleton, 2006). These limitations include the inability to distinguish signal (i.e., true predictor variables) from noise (Derksen & Keselman, 1992; Kok, Choi, Oh, & Choi, 2021; Whittingham et al., 2006; Wiegand, 2010), underestimation of p-values, and failure to replicate (Smith, 2018; Thompson, 1995). As such, many alternative variable selection algorithms have been proposed in the literature, but behavioral researchers have been slow to adopt these new methods in place of more traditional methods (Serang, Jacobucci, Brimhall, & Grimm, 2017; Shi, Shi, & Fairchild, 2023). One potential reason for this delay may be the disconnect between methodological and applied behavioral researchers, as much methodological research is often inaccessible for applied researchers at first (e.g., complex techniques, lack of published code, or no tutorials). An APA PsychINFO database search for the term "variable selection" returned 253 papers published

in quantitative methods journals in the last 20 years, indicating that methodological researchers are dedicated to developing better approaches to variable selection than stepwise regression. Of these publications, however, only one is a tutorial (Gunn, Hayati Rezvan, Fernández, & Comulada, 2023).

Given the clear gap in the popularity of variable selection methodological research and the lack of tutorials on how to apply them, the field would benefit greatly from additional tutorials on variable selection techniques with demonstrations of how to apply them to psychological datasets. The following groups would benefit, specifically, from this tutorial. First, behavioral and health science researchers who are working with big data or looking to further enhance their understanding of advanced variable selection techniques to build more robust and interpretable models. Second, graduate students and early career researchers who are new to machine learning and variable selection methods and seek practical guidance on applying these techniques in their own research. Third, those who may be teaching courses on data analysis, machine learning, or statistics who are looking for comprehensive examples to illustrate advanced techniques to their students. By following this tutorial, readers will gain practical knowledge on implementing five advanced variable selection methods in R, insights into the strengths and weaknesses of each method, helping researchers to choose the most appropriate technique for their specific research question, and access to the associated code on Open Science Framework, providing an interactive learning experience. We encourage social and health sciences researchers to adopt these advanced methods, leading to more robust, interpretable models.

Specifically, the goal of this paper is to provide a tutorial on five variable selection techniques freely available to researchers in R. We will introduce the Least Absolute Shrinkage and Selection Operator (LASSO), Elastic Net, a version of the genetic algorithm (GA), and implementations of Support Vector Machines (SVMs) and Random Forest that have been adapted to perform variable selection. The manuscript is organized as follows. The first section illustrates the importance of variable selection in machine learning and explains why each of the five methods was selected. Then, a motivating example pertaining to the diagnosis of misophonia is provided. The dataset was collected from a psychology research pool and represents an excellent example of a dataset available to many behavioral and health researchers (Norris, Kimball, Nemri, & Ethridge, 2022). Within this example, there are three major sections. The first discusses methods using a logistic regression model (i.e., LASSO, EN, and the GA), the second discusses SVM, and the third pertains to random forest. Each technique is introduced, the code necessary to implement each technique is provided, and each technique's associated strengths and weaknesses are discussed. This paper is written with the assumption that individuals have at least a basic understanding of R.

1.1 Variable Selection in Machine Learning

Objectives of Variable Selection Variable selection is a fundamental step in the process of building robust and efficient machine learning models, and its

importance cannot be overstated (Chowdhury & Turin, 2020; Guyon & Elisseeff, 2003). It serves as a critical mechanism for optimizing model performance and ensuring its reliability across various tasks and datasets. The goal of variable selection (also known as feature selection in machine learning literature) is to identify the most informative (i.e., best) subset of variables for a given task. The criteria for defining "best" vary depending on the researcher's objectives, as highlighted by Huang (2015). Highlights of Huang's discussion argue that there are two main objectives of variable selection: (1) to improve the accuracy of the model, and (2) to determine the relevance of the variables in the model so as to better guide researchers' hypothesis generation.

Types of Variable Selection In the field of machine learning, variable selection techniques are often classified into one of three categories, initially discussed in the seminal paper by Guyon and Elisseeff (2003): filter methods, wrapper methods, and embedded methods .

Filter methods (e.g., χ^2, Euclidean distance, or the i-test) are often used as a pre-processing step, but they can be used as a stand-alone variable selection method. These techniques choose variables (or features) before building any model to measure the construct of interest. For example, a filter could select items based on a particular feature relevance score, a variable's correlation with the constructs of interest, or the variable's amount of variance. Most often, significance testing is used as a filter method to determine variable selection (e.g., a variable would need to correlate significantly, as determined by a p-value, with the outcome variable). However, these significance tests occur in a univariate fashion (i.e., one variable is tested at a time), which ignores possible interaction effects or covariance among variables. No filter methods are presented in this tutorial, as past research indicates they provide inferior results and miss important information as the selection is separate from model estimation (Blum & Langley, 1997; Guyon & Elisseeff, 2003; Kohavi, 1996), but we include a brief overview to provide the reader with a full picture of the types of variable selection methods that exist.

Wrapper methods improve upon filter methods by accounting for a variable's ability to measure the construct of interest. Each wrapper method operates under a specific algorithmic ideology from machine learning (e.g., stepwise regression techniques operate as greedy algorithms, choosing the variable that will optimize the selected criteria at each step). Wrapper methods are flexible in that they are not constrained to any one type of model (e.g., regression, structural equation modeling, etc.) but rather can be "wrapped" around the researcher's chosen model. The wrapper method explained in this tutorial is the genetic algorithm, which we have wrapped around a logistic regression model for classification purposes. More details about the genetic algorithm will be provided in a later section of this paper.

Embedded methods are similar to wrapper methods in how well a set of variables predicts the given construct of interest. Embedded methods differ from wrapper methods in that they perform variable selection while simultaneously

estimating the prediction model (Guyon & Elisseeff, 2003). Although this often results in higher efficiency than wrapper methods, embedded methods are constrained to one type of model. The embedded methods discussed in this tutorial are LASSO and Elastic Net which use a logistic regression classification model (Engebretsen & Bohlin, 2019), Elastic SCAD SVM which uses an SVM classifier (Becker, Toedt, Lichter, & Benner, 2011), and Boruta which uses a random forest classifier (Kursa & Rudnicki, 2010).

Variable Selection Importance Variable selection is advantageous with any model (e.g., regression, structural equation modeling, etc.) because, as mentioned previously, it leads to more accurate predictions, reduces the computational cost of the model, and improves the parsimony of the model by eliminating redundant and irrelevant variables. However, there are additional advantages to variable selection when paired with machine learning models. First, variable selection helps manage dimensionality problems (i.e., when a dataset contains more predictors than observations). Over the years, technology such as the invention of online data collection platforms like Prolific or the creation of mobile health apps has allowed researchers to collect more complex data from increasingly larger samples. As datasets grow in both size and complexity, the number of variables may also increase, leading to computational inefficiencies and reduced model interpretability (Barceló, Monet, Pérez, & Subercaseaux, 2020). By carefully selecting relevant variables, we can effectively reduce the dimensionality of the data, thereby streamlining the computational process and facilitating easier interpretation of the model (Jia, Sun, Lian, & Hou, 2022)

Moreover, the variable selection process enables models to achieve higher accuracy and better generalization capabilities. For example, van Vuuren et al. (2021) found that LASSO created a model that was able to classify students as at risk for suicide with a higher accuracy than simple inclusion rules (i.e., predicting based on history of suicide alone). Pratik, Nayak, Prasath, and Swarnkar (2022) utilized Elastic Net to select variables that were able to predict smoking addiction in young adults with higher accuracy than previous research. By focusing on the most informative variables, the model can discern meaningful patterns within the data, leading to more precise predictions and improved performance on unseen or new data. This selective approach prevents the model from being overwhelmed by noise or irrelevant information, allowing it to focus on capturing the underlying relationships that drive the outcome of interest. For example, researchers found that applying Elastic Net regularization to classifiers based on clinical notes reduced the number of features selected by more than a thousandfold, making these classifiers more easily interpretable and maintaining performance (Marafino, John Boscardin, & Adams Dudley, 2015).

Furthermore, the inclusion of irrelevant variables in the modeling process can introduce bias and adversely affect the estimation of model parameters. Additionally, extraneous variables may introduce noise or confounding factors, leading to skewed parameter estimates and potentially misleading conclusions (Kerkhoff & Nussbeck, 2019). By excluding such variables through proper selec-

tion techniques, we can ensure that the model's estimates remain unbiased and reflects the true underlying relationships in the data, increasing the ecological validity of study results and models produced.

Lastly, a well-selected set of variables enhances the model's predictive performance and contributes to its stability and reliability (Arjomandi-Nezhad, Guo, Pal, & Varagnolo, 2023; Fox et al., 2017). Models built on a carefully chosen subset of variables are less susceptible to overfitting, where the model simply memorizes the data rather than learning meaningful patterns. Avoiding overfitting leads to more robust models that generalize better and are less prone to erratic behavior or unexpected deviations, which may lead to harmful classifications (e.g., classifying an individual as having a particular disorder when they do not; Cateni, Colla, & Vannucci, 2010; Heinze, Wallisch, & Dunkler, 2018).

Put simply, variable selection is indispensable in the realm of machine learning. It serves as a cornerstone for improving computational efficiency, enhancing model accuracy and generalization, reducing bias in parameter estimation, and fostering the stability and reliability of the resulting models. As such, behavioral and health researchers must employ rigorous techniques and considerations during the variable selection process to ensure the models' and conclusions' effectiveness and generalizability.

Applications of Variable Selection Methods Understanding the appropriate contexts for applying different variable selection methods is crucial for researchers to make informed decisions. Below we outline scenarios where each of the five methods discussed in this tutorial – LASSO, Elastic Net, genetic algorithm (GA), support vector machines (SVM), and random forest – can be most effectively utilized.

LASSO is particularly effective for datasets with a large number of predictors, especially when many predictors are thought to be irrelevant or redundant (Tibshirani, 1996). It is often used in clinical research for identifying key biomarkers from extensive genetic data or in psychological students for selecting significant psychological traits that predict mental health outcomes (Chu et al., 2024; Wettstein et al., 2023). However, LASSO is constrained by degrees of freedom requirements, so, if researchers' data contains more predictors than observations, this approach would be infeasible.

Elastic Net is best suited for datasets with highly correlated predictors. It combines the strengths of both LASSO and Ridge regression, which makes it most suitable for complex datasets with multicollinearity. This method is applied in epidemiology to study the impact of multiple, correlated environmental exposures on health outcomes and in social sciences to analyze survey data where multiple questions pertaining to a given latent construct are often correlated (Han & Dawson, 2021; Pratik et al., 2022).

The genetic algorithm is ideal for complex optimization problems where traditional methods may fail to find the global optimum. It is flexible and can be adapted to various types of models and data structures. If researchers believe there may be strong interactions between variables, this approach may be most

appropriate. In fact, GA has been used in the behavioral and health sciences to explore variable selection when interactions between numerous behavioral variables are present, or hypothesized to be present, in the data (Adams, Bello, & Dumancas, 2015; Basarkod, Sahdra, & Ciarrochi, 2018; Gan & Learmonth, 2016; Moore et al., 2017; Yukselturk, Ozekes, & Türel, 2014).

SVMs are highly effective for classification problems with high-dimensional (where there are more predictors than observations) data. They are robust to overfitting, especially when an advanced kernel function (discussed in more detail later) are used. They are often used in medical diagnosis for classifying patients based on medical imaging data (Becker, Werft, Toedt, Lichter, & Benner, 2009; Fernandez, Caballero, Fernandez, & Sarai, 2011) and in classification studies such as predicting dementia (Battineni, Chintalapudi, & Amenta, 2019).

Random forest performs particularly well when data have a mix of variable types or complex interactions. It handles large datasets well, provides measures of variable importance, and is less prone to overfitting than some other approaches due to the ensemble approach. Random forest has been applied to educational psychology to assess student related outcomes (Alamri et al., 2021; El Haouij et al., 2018; Tan, Main, & Darolia, 2021). Within the health sciences, researchers have used random forest to predict cases of COVID-19, predict risk for adverse health effective, and identify longitudinal predictors of health (Cafri, Li, Paxton, & Fan, 2018; Iwendi et al., 2020; Loef et al., 2022).

Our Chosen Variable Selection Techniques Researchers have a variety of variable selection methods available to them, and many are freely available to researchers in R packages. Perhaps the most widely applicable and easy-to-use R package for variable selection is the relatively new *FSinR* package (Aragón-Royón, Jiménez-Vílchez, Arauzo-Azofra, & Benítez, 2020), which contains a large number of filter and wrapper methods widely used in the literature for both classification and regression models that are available in the R caret package (Kuchirko, Bennet, Halim, Costanzo, & Ruble, 2021). A short, non-exhaustive list of other easy-to-use R packages for variable selection is cited here for the reader's convenience (Calcagno & Mazancourt, 2010; Genuer, Poggi, & Tuleau-Malot, 2010; Kursa & Rudnicki, 2010; Strobl, Malley, & Tutz, 2009; Trevino & Falciani, 2006; Wehrens & Franceschi, 2012).

The five techniques utilized in this paper were chosen for a variety of reasons. First and foremost, LASSO and Elastic Net are arguably the most popular modern variable selection techniques within the behavioral sciences. The implementations used in this tutorial come from the *glmnet* R package (Friedman, Hastie, & Tibshirani, 2010; Tay, Narasimhan, & Hastie, 2023). Social psychology researchers have used such techniques to create better environments that promote prosocial environments for children (Chu et al., 2024), and health researchers have used them to model the progression of Alzheimer's disease (Liu, Cao, Gonçalves, Zhao, & Banerjee, 2018). Implementations of SVM and random forest were chosen because of their strength as classification algorithms and because they can handle more complex data types (e.g., mixed variable types or

non-linearly separable). The SVM implementation comes from the *penalizedSVM* package (Becker et al., 2009) while the random forest implementation comes from the *Boruta* package (Kursa & Rudnicki, 2010). Lastly, the GA was chosen (1) to introduce the reader to the concept of metaheuristic approaches to variable selection and (2) because it has been shown to outperform more common methods like LASSO and Elastic Net across a variety of different data conditions (Bain, Shi, Boness, & Loeffelman, 2023). The GA implementation comes from the *GA* package (Scrucca, 2013, 2017). Note that while this paper includes core code snippets, the accompanying Open Science Framework (OSF) repository provides the complete code and data necessary to replicate all analyses. The repository link is provided in the data availability section.

A Motivating Example This tutorial uses the assessment of misophonia as an example through which we illustrate each technique. Individuals with misophonia experience strong, negative, emotional responses to specific sounds (i.e., triggers Wu, Lewin, Murphy, & Storch, 2014). The original data sample consisted of undergraduate students ($N = 343$) at a large southwestern university. Participants were predominately white (76.7%), female (69.7%), and students (96.5%) ranging from ages 18 to 36 ($M = 18.96$, $SD = 1.7$). The dataset contains 106 independent variables related to both direct characteristics of misophonia and related characteristics, as well as one self-report binary diagnosis variable. It is available to the reader on the accompanying OSF repository linked in the availability of data and materials section of this paper. Since misophonia is still not fully understood (i.e., formal diagnostic criteria have not been set, and researchers are still trying to determine the most important symptoms), this dataset is an illustrative example of variable selection. Some symptoms may be unimportant for, or not predictive of, a true misophonia diagnosis. One should note that this dataset does not contain any missing data, as it was handled *a priori* using listwise deletion. In addition, one should note that the group sizes are unbalanced (16.5% diagnosed, 83.5% not). This presents additional complexity and is one reason why we have chosen to evaluate the methods using both accuracy and F-score. For more information on the larger previously published dataset from which this data was selected and the background on misophonia, see the work of Norris et al. (2022).

The Importance of Cross Validation. Model overfitting is a common problem for implementing variable selection techniques (see Figure 1). If a model is built too closely to the specifications of a specific dataset (i.e., it is not robust to changes in the data), it is considered overfit. Alternatively, a model can be underfitted where it is built in such a way that it is too generalizable and does not create accurate or meaningful predictions. Researchers need to be cautious of overfitting and underfitting to ensure that they build models that can accurately generalize to new data while making meaningful and accurate predictions.

Cross-validation is one common way to help researchers increase generalizability in a meaningful way (i.e., protect against overfitting). In cross-validation, the model is built on (or, in the case of this tutorial, variables are selected from)

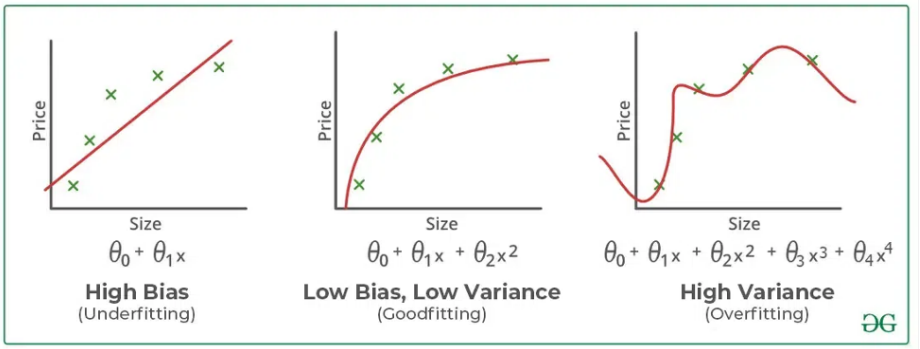


Figure 1. The leftmost graph illustrates an underfit model on a small amount of data. The middle figure illustrates a fit that balances both bias and variance leading to good fit. The rightmost graph illustrates an overfit model. Figure obtained from Geeks for Geeks (*ML | Underfitting and Overfitting*, 2017)

a different set of data than it is evaluated. Although this can occur through the collection of two different datasets, this is typically done by dividing one dataset into parts. One can do this division in many ways, and this paper implements holdout cross-validation, which occurs when one splits the data into two sets (test and training sets) before conducting any analyses. Typically, 70% of the data is used for the training set in holdout cross-validation, and the remaining 30% is used for the test dataset. The code for how we performed holdout cross-validation can be found in the companion code on OSF. For additional information on the importance of cross-validation and alternative approaches to cross-validation, see the helpful tutorials cited here (Ghojogh & Crowley, 2023; Song, Tang, & Wee, 2021).

2 Methods

2.1 Logistic Regression Models

Logistic regression is a widely used statistical model for binary classification problems and models the probability that a given observation (e.g., a set of participants' responses to a given questionnaire), belongs to a particular category. The equation for logistic regression is :

$$P(Y = 1|\mathbf{X}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m)}} \tag{1}$$

Here, $P(Y = 1|\mathbf{X})$ represents the probability that the participant belongs in class 1 given their response matrix ($\mathbf{X}$). The intercept term (β_0), is the value of the log-odds when all predictor variables are zero. The coefficients ($\beta_1, \ldots, \beta_m$) associated with each of the predictor variables ($x_1, \ldots, x_{\Updownarrow}$) represent the change in the log-odds of the dependent variable for a one-unit change in the corresponding predictor variable for a total of m predictors.

Regularization Techniques Two of the techniques discussed in this paper, LASSO and Elastic Net, are regularization techniques. Regularization is a common method used to combat issues of overfitting found in models estimated with maximum likelihood estimation (like logistic regression). Each regularization technique works to combat overfitting by intentionally introducing a small amount of bias into the model such that a generic regularization function, within the context of classification, takes the following form:

$$L^{Reg}(\boldsymbol{\beta}) = L^{logistic}(\boldsymbol{\beta}) - \lambda \mathscr{P}(\boldsymbol{\beta}) \tag{2}$$

where L^{Reg} is the penalized optimization function, $L^{logistic}$ is the negative log likelihood, λ is a regularization parameter (i.e., a tuning parameter), and $\mathscr{P}$ is a penalty function that will vary across the regularization technique. The goal of regularization is to find the optimal balance between bias (generalizability of the model) and variance (specific model fit Helwig, 2017). The magnitude of the lambda (λ) penalty determines this balance. A larger lambda will lead to a sparser and more generalizable model. One popular technique utilized to determine the value of the lambda parameter is cross-validation. As mentioned above, cross-validation occurs when the data is split into multiple subsets, the model is developed (i.e. trained) on a subset, and evaluated (i.e., validated) on another. This process is iterative, allowing for the selection of the lambda penalty that minimizes prediction error across different subsets.

One optimal model, in the context of this paper, is one that produces the most accurate classifications. Accuracy can be calculated using the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where TP is the number of individuals who were correctly classified as having a diagnosis of misophonia, TN is the number of individuals who were correctly classified as not having a diagnosis of misophonia, FP is the number of individuals who were classified as having a diagnosis but did not truly have a diagnosis in the labeled data, and FN are the number of individuals who were incorrectly classified as not having a diagnosis when a diagnosis was present in the labeled data. It is worth noting that accuracy may not be the best optimization criteria given the unbalanced nature of the data (i.e., the number of observations in class 0 is much larger than the number in class 1). In practice, researchers may want to use a weighted accuracy or an F-score in their own research, depending on the relative importance of a false positive versus a false negative. For example, a clinician attempting to predict suicide attempts may prioritize a false positive (i.e., saying the individual is likely to attempt suicide when they do not actually attempt) over a false negative (i.e., saying the individual will not attempt when they actually will). Non-weighted accuracy was included for ease of explanation. However, we will also evaluate each model in terms of an F-score to illustrate the differences between these metrics. The equation for calculating an F-score is seen below.

$$F1 = \frac{TP}{TP + .5(FP + FN)} \tag{4}$$

The F1 score is a measure of a model's ability to balance precision (accuracy of positive predictions) and recall (correct identification of positive instances). The equation provided modifies the traditional F1 score by scaling the sum of false positives (FP) and false negatives (FN) by 0.5, reducing their weight in the final score. This adjustment can be useful when false positives and false negatives are not equally important or should be penalized less.

LASSO. LASSO (Tibshirani, 1996) is one of the penalized regression techniques that perform variable selection. LASSO can handle data with multicollinearity, be applied to various types of data (e.g., continuous, categorical, mixed type), and is adaptable to sparse data (i.e., multiple predictors have zero or near-zero coefficients; Foucart, Tadmor, & Zhong, 2023; Mendez-Civieta, Aguilera-Morillo, & Lillo, 2021). The parameter estimates (i.e., the $\boldsymbol{\beta}$ coefficients) for LASSO can be obtained by maximizing the penalized log-likelihood function:

$$L^{LASSO}(\boldsymbol{\beta}) = \sum_{i=1}^{n}[y_i \mathbf{x}_i \boldsymbol{\beta} - log(1 + e^{\mathbf{x}_i \boldsymbol{\beta}})] - \lambda \sum_{j=1}^{m} |\beta_j| \tag{5}$$

where $L^{LASSO}(\boldsymbol{\beta})$ is is the loss function and is comprised of two summations. The first summation represents the logistic regression log likelihood and n is the number of observations in the data, y_i represents the actual binary outcome of the i-th observation, $\mathbf{x}_i$ is the vector of predictor variables for the i-th observation, $\boldsymbol{\beta}$ is the vector of coefficients (including the intercept term), and $log(1 + e^{\mathbf{x}_i \boldsymbol{\beta}})$ is the log of the logistic function denominator, which ensures that the probabilities are correctly bounded between 0 and 1. The second summation is the LASSO penalty (or the ℓ_1 regularization term) which adds a penalty proportional to the absolute value of the coefficients and m is the number of predictors in the initial model. Here λ is the regularization hyperparameter that controls the degree of shrinkage such that larger values lead to the selection of fewer variables and $\sum_{j=1}^{m} |\beta_j|$ is the sum of the absolute values of the coefficients for all predictor variables (note that the summation begins at 1, indicating that the intercept, β_0, is excluded from regularization and must be included in the final. For a more detailed discussion of LASSO, see Tibshirani's (1996) paper . Regularization techniques are useful for variable selection because they add a penalty for large coefficients, effectively shrinking less important variables towards zero and thus eliminating them from the model. This helps in improving model interpretability and preventing overfitting, particularly in scenarios with a large number of predictors or multicollinearity.

As with all methods, researchers may be interested in the recommended sample size LASSO. One conservative estimate suggests that researchers should have 10 observations per candidate variable (e.g., with 10 variables, a researcher would need 100 observations; Peduzzi, Concato, Feinstein, & Holford, 1995; Peduzzi, Concato, Kemper, Holford, & Feinstein, 1996). However, this recommendation is made more generally for regression, and thus, does not generalize as specifically to regularization techniques where not all variables are included in the final model. Recent simulation studies have investigated the performance of LASSO in

small sample sizes (e.g., 50 – 100 participants) and found that methods perform well (Bain et al., 2023; Kirpich et al., 2018; Wen et al., 2019).

To utilize LASSO for variable selection, we use the `cv.glmnet()` function from the *glmnet* package in R (Friedman et al., 2010). More information on the hyperparameters of the function can be found in Table 1. This function determines the magnitude of lambda through a k-fold cross-validation approach.

```
lasso.model <- cv.glmnet(x = predTrain,
    y = outcomeTrain, type.measure = "class",
    alpha=1, family="binomial", nfolds = 10)
```

Through this model, we can obtain the chosen lambda value. To obtain a full list of all evaluated lambda values, use `lasso.model$lambda`. One can also plot the k-fold cross-validation procedure to obtain λ using `plot(lasso.model)` (Figure 2). There are two lambda values that are particularly of interest. The first can be obtained with `lasso.model$lambda.min`. This lambda value is responsible for producing the model with minimal cross-validated error. The second can be obtained with `lasso.model$lambda.1se`, or the 1se rule. This lambda value is responsible for producing the model that has a cross-validated error within one standard error of the minimum. There are advantages to each. Breiman and colleagues (2017) as well as Chen and Yang (2021) suggest that researchers should use the 1se rule to select lambda to reduce the instability of the model while maintaining a parsimonious model. However, this gain in stability comes with a loss in accuracy (an increase in misclassification error of one standard error). In addition, some research has shown that the 1se rule performs poorly in regression (Chen & Yang, 2021) as opposed to a classification tree, so we used the value that minimized cross-validation error (lambda min). To obtain our lambda min value, specify `lasso.model$lambda.min`. Using this specified lambda value, we can build a LASSO model using the `glmnet()` function with the following code, which will produce the coefficients as seen in Table 2.

```
lasso.model.min <- glmnet(x = predTrain,
    y = outcomeTrain, alpha=1,
    family="binomial",
    lambda = lasso.model$lambda.min)
```

Out of the original 106 predictor variables, only 16 were selected via LASSO, thus a sparse model has been obtained. It is important to examine what variables were selected by the model to ensure that they are theoretically justified. Ideally researchers would make this decision about all variables, however, for the sake of space within this paper, we have chosen to only examine two of the 16 selected items. One selected item, MQ4 reads: "In comparison to other people, I am sensitive to the sound of people making nasal sounds." As nasal and throat sounds are often thought to be triggers for those with misophonia, this item makes theoretical sense to be a predictor of the diagnosis. For another selected item, S5_7, participants were asked, "Please rate your typical reaction to the following stimuli, if produced by another person: Throat clearing." This item is

Table 1. Hyperparameters of the `cv.glmnet()` function and their corresponding definitions.

Parameter	Description
x	A matrix of predictor (or input) variables.
y	The vector containing the response (or outcome) variable.
type.measure	The optimization measure to be used within the internal cross-validation procedure. By setting this to "class" misclassification error is optimized.
alpha	The Elastic Net mixing hyperparameter. Because the same function is used to implement ridge, LASSO, and Elastic Net, the value for alpha determines which regularization technique is run. Alpha is constrained between 0 and 1, with a value of 0 implementing ridge regression, 1 implementing LASSO regression, and anything in between implementing an Elastic Net regression.
family	The type of regression to be implemented. By setting this hyperparameter to "binomial" an MLE regression is implemented.
nfolds	The number of partitions implemented in the internal k-fold cross-validation.

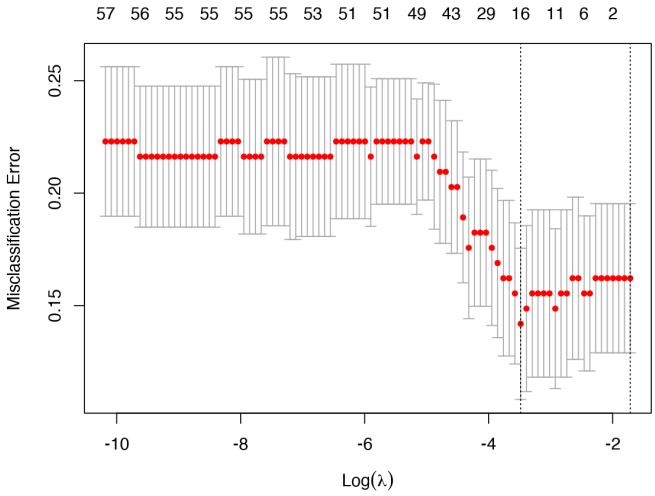


Figure 2. Cross-validated estimate of the mean squared prediction error for LASSO as a function of the log λ. The upper axis indicates the number of non-zero coefficients in the regression model at the given log λ. The dashed vertical line illustrates the location of the CV minimum and the one standard error rule locations for λ.

theoretically justifiable for the same reason as above, reactions to throat sounds are a symptom of the disorder.

The coefficient estimates obtained through a LASSO approach are biased by the nature of the algorithm (Yarkoni & Westfall, 2017), and thus research

Table 2. A table containing the variables selected by the LASSO model and their corresponding estimated coefficients. The table also includes the full item for that particular variable. If items share a common stem, we have grouped them together.

Variable	Coefficient	Full Item
(Intercept)	-5.741	
MQ4	0.154	In comparison to other people, I am sensitive to the sound of people making nasal sounds (e.g., inhale, exhale, sniffing, etc.).
Once you are aware of the sound(s), because of the sound(s), how often do you:		
MQ11	0.039	Cover your ears?
MQ12	0.137	Feel anxious or distressed?
MQ13	0.112	Become sad or depressed?
MQ17	0.045	Become physically aggressive?
Please rate your typical reaction to the following stimuli, if produced by another person:		
S5_7	0.087	Throat clearing
S5_24	-0.032	Car engine
S5_25	0.318	Clock ticking
S5_31	0.159	Pacing
S5_32	0.024	Nail biting
S5_35	0.123	Strong smells
S5_36	0.089	Seeing someone chew gum
Please indicate your level of agreement to the following statements:		
S5_56	0.124	I can feel physical pain if I cannot avoid a sound.
S5_57	0.419	Sometimes in response to sounds I feel rage that is difficult to control.
S5_75	0.252	Some sounds have caused me to use violence towards myself or others.
S5_78	-0.018	It does not matter who is making the sounds, my reactions are the same.

recommends recalculating them using a standard regression before interpreting the coefficients of the model. To do that, one could use the following code.

```
selected <- trainDat %>% select(MQDX, MQ4, MQ11,
    MQ12, MQ13,MQ17, S5_7,S5_24, S5_25, S5_31,
    S5_32, S5_35, S5_32,S5_35, S5_36, S5_56,
    S5_57, S5_75, S5_78)
logistic.model <- glm(MQDX ~ .,
    family=binomial(link = "logit"),
    data = selected)
```

In this code, we first use the `select()` function from the *dplyr* package to select only the variables with non-zero coefficients in the lasso.model.min as well as our outcome variable, MQDX (Wickham et al., 2023). We then use these variables to build a standard logistic regression model using the `glm()` function.

A comparison of the biased coefficients obtained from the LASSO model and the corrected coefficients obtained in the standard logistic model can be seen in Table 3. Obtaining the predicted classification prior to calculating accuracy is crucial. Accuracy values (Equation 3) are then determined using the coefficients estimated from both the LASSO model (incorrectly biased) and the logistic model. The following code can be used to obtain the accuracy values from the logistic model as well as the F-score from the model. Note that the F-score is obtained using the `F1_Score()` function from the *MLmetrics* package (Yan, 2024).

Table 3. A table containing the variables selected by the LASSO model and the coefficient estimates obtained directly from the LASSO model as well as the re-estimated (non-biased) coefficients obtained by creating a typical logistic model using the selected variables.

Variable	LASSO Estimate	Logistic Estimate
(Intercept)	-5.741	-8.802
MQ4	0.154	0.361
MQ11	0.039	0.480
MQ12	0.137	0.760
MQ13	0.112	-0.193
MQ17	0.045	0.143
S5_7	0.087	-0.057
S5_24	-0.032	-1.154
S5_25	0.318	1.051
S5_31	0.159	0.538
S5_32	0.024	0.245
S5_35	0.123	0.110
S5_36	0.089	0.285
S5_56	0.124	0.387
S5_57	0.419	0.867
S5_75	0.252	0.186
S5_78	-0.018	-0.600

```
pp.logistic <- predict(logistic.model,
    data.frame(predTest),
    type = "response")
pc.logistic <- ifelse(pp.logistic > .5, 1, 0)
a.logistic <- mean(outcomeTest == pc.logistic)
```

```
f1.logistic <- F1_Score(pc.logistic, outcomeTest)
```

In the first line of code, using the `predict()` function, the `logistic.model` object and our `predTest` data (reminder that this is the holdout sample created during cross-validation earlier) we can create our predictions. By specifying `type = "response"`, the function will return predicted probabilities. In our second line, the predicted probabilities are transformed into predicted classes such that if the probability of them belonging to class 1 is at least 0.5, they are assigned to class 1 otherwise class 0. The third line calculates accuracy. The value obtained using the coefficient estimates from the LASSO model is an accuracy score of 0.86. The value obtained using the coefficient estimates from the logistic model is 0.89. The F-score for both the LASSO model and the logistic model is 0.92. Note that the accuracy changes across models, but the F-score remains the same. This indicates that the models likely differ only in their true negative results, as that measure is not included in the calculation of the F-score.

Despite the strong performance of LASSO on this data, LASSO does have limitations (Algamal & Lee, 2015). First, it is unable to select more variables than there are observations. Second, LASSO will select a single variable in the presence of multicollinearity regardless of that variable's predictive capacity. Zou and Hastie (2005) proposed a new regularization technique called Elastic Net to combat these first two limitations.

Elastic Net. Elastic Net differs from LASSO through the use of an additional penalty to the regression equation. Elastic Net implements both the ℓ_1 penalty, or the LASSO penalty, and the ℓ_2 penalty, or the ridge penalty, to the regression equation. With the inclusion of both penalties, the optimization function for Elastic Net is as follows:

$$L^{ElasticNet}(\boldsymbol{\beta}) = \sum_{i=1}^{n}[y_i \mathbf{x}_i \boldsymbol{\beta} - log(1 + e^{\mathbf{x}_i \boldsymbol{\beta}}] - \lambda_1 \sum_{j=1}^{m} \beta_j^2 - \lambda_2 \sum_{j=1}^{m} |\beta_j| \quad (6)$$

The first summation represents the log likelihood and is exactly the same as was seen in Equation 4. The second summation is new to the reader as it is the ridge penalty, which adds a penalty proportional to the squared value of the coefficients (Hoerl & Kennard, 1970). Here λ_1 is the regularization hyperparameter that controls the degree of shrinkage such that larger values lead to the selection of fewer variables. The third summation is the LASSO penalty, which only differs from Equation 4 in that we now use λ_2 (instead of just λ) to denote the regularization hyperparameter that controls the degree of shrinkage from the LASSO penalty. The values for λ_1 and λ_2 can be equal or can be set to different values to allow differential application of the penalties. By incorporating the ridge penalty, Elastic Net can select multiple correlated variables while removing irrelevant ones (Algamal & Lee, 2015). For more on the ridge penalty, see work by McDonald (2009). This makes Elastic Net more suitable than LASSO for datasets with highly correlated predictors, such as dummy-coded variables.

Sample size considerations should also be made when researchers are considering using Elastic Net. The recommendations are similar to those for LASSO in

that conservative estimates suggests that researchers should have 10 observations per candidate variable similar to LASSO (Peduzzi et al., 1995, 1996). However, this recommendation comes from the general regression literature, and thus, may not hold with regularization. Recent simulation studies have investigated the performance of Elastic Net in small sample sizes (e.g., 50 – 100 participants) and found that methods perform well (Bain et al., 2023; Kirpich et al., 2018; Wen et al., 2019).

We can obtain our `lambda.min` value using the `cv.glmnet()` function, just as we did for LASSO. However, we change the value for alpha from alpha = 1 to alpha = 0.5. We can then use this value to build our final Elastic Net model (the second piece of code below). We can then use the variables with non-zero coefficients from our final Elastic Net model (`en.model.min`) to build a standard logistic regression model (`logistic.en.model`) to get unbiased coefficients, as was done for LASSO. Two of the selected items include MQ11, and S5_11. MQ11 reads, "Once you are aware of the sound(s), because of the sound(s), how often do you actively avoid certain situations, places, things, and/or people in anticipation of the sound(s)." Individuals with misophonia are known to employ a variety of coping strategies (including avoidance) to deal with their triggering sounds, so this variable makes sense theoretically. S5_11 reads, "Please rate your typical reaction to the following stimuli, if produced by another person: Repetitive barking." This item is interesting, because some research has found that not all sounds must be human made to be triggers for individuals with misophonia, for example, this is a sound most often made by dogs, not people. However, it is theoretically sound.

```
elasticNet <- cv.glmnet(x = predTrain,
    y = outcomeTrain, type.measure = "class",
    alpha=0.5, family="binomial", nfolds = 10)
en.model.min <- glmnet(x=predTrain y=outcomeTrain,
alpha=0.5, family="binomial",
    lambda = elasticNet$lamda.min)
selected <- trainDat %>% select(MQDX, MQ4, MQ11,
    MQ12, MQ13, MQ15, MQ16, MQ17, S5_2, S5_7, S5_11,
    S5_24, S5_25, S5_27, S5_31, S5_32, S5_35, S5_36,
    S5_38, S5_40, S5_42, S5_53, S5_56, S5_57, S5_68,
    S5_74, S5_75, S5_78, S5_82)
logistic.en.model <- glm(MQDX ~.,
    family=binomial(link = "logit"),
    data = selected)
```

Coefficient estimates from the Elastic Net model and unbiased coefficients from a standard logistic model can be seen in Table 4. An accuracy of 0.88 was obtained using the coefficient estimates from the Elastic Net model, while an accuracy of 0.80 was obtained using the coefficient estimates from the logistic model. The F-score obtained using the coefficient estimates from the Elastic Net model is 0.94, while the logistic model produces an F-score of 0.88. The code below illustrates how to obtain the 0.80 accuracy value and 0.88 F-score from

the unbiased logistic regression model. The only change the reader would need to make to obtain the estimates from the final Elastic Net model instead would be to substitute `en.model.min` for `logistic.en.model`.

```
pp.en.logistic <- predict(logistic.en.model,
    data.frame(predTest), type = "response")
pc.en.logistic <- ifelse(pp.logistic > .5, 1, 0)
a.en.logistic <- mean(outcomeTest == pc.logistic)
f1.en.logistic <- F1_Score(pc.en.logistic,
    outcomeTest)
```

Table 4. A table containing the variables selected by the Elastic Net model and their corresponding estimated coefficients obtained directly from the Elastic Net model as well as the coefficients estimated by implementing a logistic model (non-biased coefficients).

Variable	Elastic Net Estimate	Logistic Estimate
(Intercept)	-5.796	-1732.902
MQ4	0.133	15.606
MQ11	0.046	20.788
MQ12	0.119	81.502
MQ13	0.103	-6.765
MQ15	0.057	101.386
MQ16	0.075	5.368
MQ17	0.086	145.615
S5_2	0.023	-6.618
S5_7	0.091	12.560
S5_11	-0.051	-190.866
S5_24	-0.095	-39.021
S5_25	0.262	31.171
S5_27	0.031	94.479
S5_31	0.165	58.559
S5_32	0.092	97.889
S5_35	0.147	56.132
S5_36	0.088	38.234
S5_38	0.036	30.691
S5_40	0.048	111.532
S5_42	0.032	42.788
S5_53	-0.020	12.132
S5_56	0.190	94.590
S5_57	0.259	-87.586
S5_68	0.081	126.088
S5_74	0.018	5.356
S5_75	0.197	-17.256
S5_78	-0.089	-101.315
S5_82	0.033	-2.060

Elastic Net also has some limitations. Namely, it may struggle with datasets containing many more variables than observations, it is sensitive to outliers, and, given that it is designed for linear relationships, it may not capture complex non-linear relationships between predictors and the response variable effectively (Wang, Cheng, Liu, & Zhu, 2014).

Genetic Algorithm (GA) Unlike LASSO and Elastic Net, which utilize internal regression models as embedded methods, the genetic algorithm (GA) operates as a wrapper method. As a reminder, this means that the user must specify which model it should use (i.e., a user could wrap the GA around a logistic regression model or something more complex like a random forest or SVM, depending on the nature of their data). As mentioned, wrapper methods each follow their own algorithmic strategy to explore potential solutions (i.e., potential sets of variables to select). One wrapper method that may be familiar to readers is stepwise regression, which builds a model iteratively by either adding or removing variables based on a given criteria (e.g., Akaike Information Criteria; AIC). It uses a greedy approach, selecting the variable at each step that yields the greatest immediate improvement in the chosen criterion (e.g., the largest decrease in AIC). The GA also operates a greedy algorithm; however, its search strategy differs.

Instead of adding or removing a single variable (as is done in stepwise regression), the GA, inspired by the principles of natural selection and evolution, mimics the process of biological evolution to refine potential solutions iteratively. Through crossover, mutation, and selection mechanisms, the GA explores and evolves a population of potential solutions over successive generations, gradually improving the overall quality of solutions. Figure 3 illustrates the general structure of the genetic algorithm, depicting its iterative process of generating, evaluating, and evolving solutions. Each iteration refines the population, guiding the search towards promising regions of the solution space.

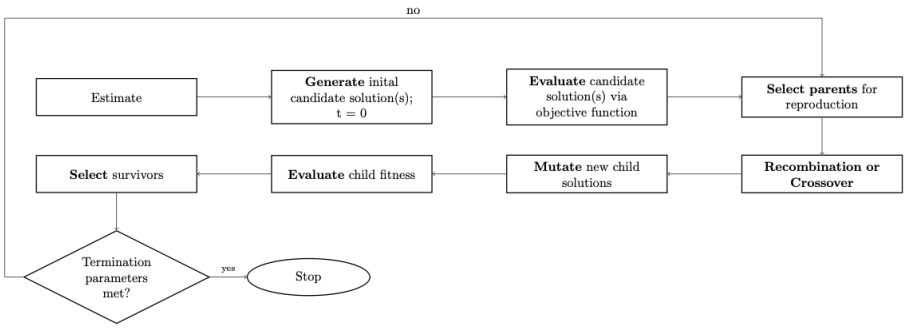


Figure 3. The basic algorithmic steps of the Genetic Algorithm.

For a comprehensive understanding of the genetic algorithm and its application to variable selection, interested readers are encouraged to refer to the work by Bain et al. (2023). Their research provides detailed insights into the underlying principles, implementation strategies, and practical considerations associated with the GA's use in solving two-group classification problems.

There are no accepted sample size recommendations for the GA for variable selection. The required sample size depends heavily on the complexity of the underlying model, the number of predictors, and the strength of the signals. As a rough guideline, samples sizes in the range of 100-500 are often used, but larger samples may be necessary for high-dimensional problems (Cateni et al., 2010; Leardi, 2000).

For this paper, logistic regression is chosen as the model around which the GA will wrap. The optimization function used in this paper is the Hubert and Arabie (1985) Adjusted Rand Index (ARI). ARI is a measure of agreeability between predicted classifications and true (or known) classifications and can be calculated in the following way:

$$ARI = \frac{RI - RI_{Expected}}{max(RI) - RI_{Expected}} \tag{7}$$

$$RI = \frac{a+d}{a+b+c+d} \tag{8}$$

$$RI_{Expected} = \frac{2(a+b)(a+c)}{(a+b+c+d)^2} \tag{9}$$

Here, a is the number of pairs of individuals (or observations) that are in the same class in both the true labels and the predicted labels, b is the number of pairs of individuals that are in the same class in the true labels but are in different classes in the predicted labels, c is the number of pairs of individuals that are in different classes in the true labels but are in the same class in the predicted labels, and d is the number of pairs of individuals that are in different classes in both the true labels and predicted labels.

The implementation of the GA used in this tutorial comes from the `ga()` function in the GA package (Scrucca, 2013, 2017). To implement the GA, the following code can be run:

```
ga.solution <- ga(fitness = function(vars)
   gaOpt(vars=vars, IV.train=data.frame(predTrain),
        DV.train=outcomeTrain),
        type = "binary", nBits = ncol(predTrain),
        names = colnames(predTrain), seed = 123456,
        run=5
)
```

Here, we set type to binary to indicate that we want binary representations of decision variables. This hyperparameter may need to change depending on the nature of the variables of interest. Second, we set `nBits` to be equal to

the number of predictor variables to indicate that all variables in the dataset could be selected. A seed is set for reproducibility. The run hyperparameter has been set to five, indicating that the algorithm should terminate if there is no improvement in the optimization function after five iterations. Note that one of the parameters in this function is the `gaOpt()` function. The `gaOpt()` function is a self-defined, user-specified function that could take on a different name. However, regardless of the name, the function must be passed as a hyperparameter in the `ga()` function. The R code needed to implement this optimization function with a logistic regression model can be seen below. For more information on the hyperparameters of the GA function and their default values, see Table 5.

Table 5. A table containing the hyperparameters of the `ga()` function and their corresponding definitions and default values.

Parameter	Description
fitness	The hyperparameter containing the optimization function is passed. No default is set.
type	The type of ga that needs to be run is dependent upon the nature of the outcome variable. "binary" is selected.
crossover	The type of crossover performed. The default for a binary implementation is found via the `ga_Crossover()` function.
popSize	An R function to generate the initial population. To access available functions, run `ga_Population()`.
pcrossover	The probability of crossover, default of 0.8 is used.
pmutation	The probability of mutation, default of 0.1 is used.
elitism	The number of best fitted chromosomes to survive at the end of each generation, default of `max(1, round(popSize*0.05))` is used.
nBits	A value specifying the number of bits in a potential solution, set equal to the number of predictors.
names	The variable names.
maxIter	The maximum number of iterations to run before the GA search is halted, default of 100 is used.
keepBest	A logical argument specifying if best solutions at each iteration should be saved, default FALSE.
seed	A number allowed to control randomness for reproducibility.
run	The number of consecutive generations that can occur without any improvement before the GA is halted, default is modified from maxiter to 5.

```
gaOpt <- function(vars, IV.train, DV.train){
varNames <- colnames(IV.train)
selectedVarNames <- varNames[vars == "1"]
gaSolutionData <- IV.train[,selectedVarNames]
    gaDat <- cbind(gaSolutionData, DV.train)
    gaMod <- glm(DV.train ~ ., family = "binomial",
        data = gaDat)
```

```
gaProbabilities <- predict(gaMod, IV.train,
        type = "response")
gaPredictedClasses <-
        ifelse(gaProbabilities >= .8, 1, 0)
ari <- adjustedRandIndex(gaPred, DV.train)
return(ari)
}
```

The `glm()` function is the same function we used to calculate logistic regression models previously. The `adjustedRandIndex()` function comes from the *mclust* package (Scrucca, Fop, Murphy, & Raftery, 2016). The `gaOpt()` function takes us through the steps of finding the ARI for the selected subset of variables. First, the names of all candidate variables are acquired, then the names of the variables selected by the GA are found, and we select only those columns from our train data. Since we had previously removed the dependent variable (the misophonia diagnosis) from the dataset, we must recombine our selected variables and our outcome variable into one matrix (line 5 above, here called `gaDat`). Next, the logistic regression model is built using these selected variables. Then the predicted probabilities are obtained, transformed into predicted classes (such that an individual is given a positive misophonia diagnosis if their probability of diagnosis is at least .8, which was chosen because only about 20% of our sample belongs to class 1). Finally, the ARI of the model is calculated and returned to the `ga()` function. To view the selected subset of variables from the `ga()` function, one calls, `ga.solution@solution[1,]`. Note, the returned solution (given by `ga.solution@solution`) contains many potential subsets of variables, but by referencing only the first row (using the indexing `[1,]`), the optimal subset of variables as determined by the GA can be accessed. Two of the selected items include item MQ18 and S5_3. Variable MQ18 reads, "Once you are aware of the sound(s), because of the sound(s), how often do you become physically aggressive" which is theoretically justifiable as individuals with misophonia are known to have disproportional, often violent, reactions to their triggers. Variable S5_3 reads,"Please rate your typical reaction to the following stimuli, if produced by another person: Swallowing," which is justifiable as it pertains to throat noises.

```
allVarNames <- colnames(predTrain)
selectedVarNames <-
    allVarNames[ga.solution@solution[1,]==1]
selectedVars <-
    data.frame(predTest[,selectedVarNames],
    outcomeTest)
ga.model <- glm(outcomeTest~., family="binomial",
data=selectedVars)
```

Since the `ga()` function does not have a specified method for model building, but rather simply returns a list of variable selections, one must first build a model to obtain an accuracy value for the selected variables. Given that the internal model we specified was a logistic regression model, it makes sense to use a simple

logistic model, which can be built using the following code. The coefficients from this model can be seen in Table 6. After building the model, an accuracy and F-score can be obtained using the following code:

```
p <- predict(ga.model, newx = predTest)
c <- ifelse(p >= .8, 1,0)
accuracy <- mean(c == outcomeTest)
f1 <- F1_Score(c,outcomeTest)
```

Table 6. A table containing the variables selected by the GA and their corresponding estimated coefficients in the logistic regression model.

Variable	Coefficient	Variable	Coefficient
(Intercept)	72.896	S5_42	-8.713
MQ4	-7.952	S5_45	8.668
MQ6	-3.454	S5_46	-10.066
MQ8	-5.707	S5_49	-1.448
MQ17	-6.154	S5_50	5.708
MQ18	21.166	S5_51	-0.198
S5_2	9.767	S5_52	-8.592
S5_3	-3.703	S5_53	-2.791
S5_4	-0.814	S5_55	-13.721
S5_6	3.907	S5_57	3.470
S5_7	-18.858	S5_58	-2.086
S5_8	1.915	S5_60	-16.660
S5_9	14.258	S5_62	4.408
S5_10	10.305	S5_63	5.143
S5_11	-11.589	S5_64	-0.372
S5_12	43.946	S5_65	4.143
S5_13	-36.764	S5_66	-8.781
S5_18	10.628	S5_68	-13.752
S5_19	2.410	S5_69	7.001
S5_20	-6.446	S5_72	12.343
S5_21	-0.442	S5_73	19.055
S5_23	3.657	S5_76	-9.715
S5_25	2.824	S5_77	-4.178
S5_26	1.490	S5_78	5.347
S5_27	4.120	S5_79	-7.315
S5_31	-4.880	S5_81	7.567
S5_32	-14.408	S5_83	-4.304
S5_33	-11.206	S5_84	-1.235
S5_38	5.880	S5_86	5.970
S5_41	11.462	S5_87	-14.504

Accuracy and F-score values of 1 are obtained, indicating a perfect fit, as with the past models built in this tutorial. Current literature indicates that the GA is prone to overfitting (Frohlich, Chapelle, & Scholkopf, 2003; Leardi, 2000;

Loughrey & Cunningham, 2005), suggesting the model would not fit quite as well if a new sample was collected, despite the accuracy of the model fit for the test sample used in this tutorial.

2.2 Support Vector Machines

Support Vector Machines (SVM) are a class of supervised learning models widely employed in classification and regression tasks (Fernandez et al., 2011; Karatzoglou, Meyer, & Hornik, 2006). SVMs operate by finding the optimal hyperplane that maximizes the margin between different classes of data points. By maximizing the margin between classes, SVM achieves good generalizability and is robust to outliers (Singla & Shukla, 2020; Xu, Caramanis, & Mannor, 2009). SVM can handle both linearly separable and non-linearly separable data by using a kernel function that artificially projects the original data into a higher-dimensional space (Karatzoglou et al., 2006).

Elastic SCAD SVM SVM, by itself, is a classification algorithm. However, researchers have created implementations of SVM that simultaneously perform classification and variable selection (Becker et al., 2011; Bierman & Steel, 2009; Tharwat & Hassanien, 2019). This tutorial uses an approach like LASSO and Elastic Net in that it selects variables via the addition of a penalty that comes from the *penalizedSVM* package (Becker et al., 2011). The penalty utilized in this tutorial is the Elastic smoothly clipped absolute deviation (SCAD) penalty, which when included in an SVM, reads:

$$SVM_{ESCAD} = min_{b,w}[sign(\mathbf{w}^T\mathbf{x} + \mathbf{b}) + \sum_{j=1}^{p} \mathscr{P}_{SCAD}\lambda_1(\mathbf{W}_j) + \lambda_2\|w\|_2^2] \quad (10)$$

where λ_1 controls the degree of shrinkage applied by the SCAD ($\mathscr{P}_{SCAD}\lambda_1(\mathbf{W}_j)$) penalty and λ_2 controls the degree of shrinkage applied by the Elastic Net ($\lambda_2\|w\|_2^2$) penalties. Higher values of either λ increase the degree of shrinkage applied by their given penalty. For more information on the SCAD penalty, see work by Becker et al. (2011). Just as with Elastic Net, the λ_1 and λ_2 values can be equal or set individually to differentially apply the penalties. The initial part of the equation ($sign(\mathbf{w}^T\mathbf{x} + \mathbf{b})$ is the base equation for an SVM where $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the input feature vector, $\mathbf{b}$ is the bias term vector, $sign(.)$ is the sign function, which returns +1 if the argument is positive, -1 if negative, and 0 if zero. All hyperparameters are set to default values in this tutorial. In addition, data needs to be restructured for this function. For a clearer understanding of the additional hyperparameters in the `svmfs()` function, see Table 7.

Generally, research shows that SVMs improve as sample sizes increase (Bain et al., 2023). However, some research has shown that sample sizes as small as 80 produce adequate classification models (average RMSEA below 0.01; Figueroa, Zeng-Treitler, Kandula, & Ngo, 2012), though the required size may increase as

Table 7. A table containing the hyperparameters of the `svmfs()` function as well as their corresponding definitions.

Parameter	Description
x	Matrix of the input or predictor variables where the columns are the variables, and the rows are the observations.
y	A numerical vector of class labels, -1, 1.
fs.method	The feature (or variable) selection method. Available methods include 'scad', '1norm' used for LASSO, 'DrHSVM' for Elastic Net, and 'scad+L2'; for Elastic SCAD.
bounds	For an interval grid search a list of values for lambda1 and lambda 2 must be provided to the model.
grid.search	The inner validation method used to obtain the values for lambda1 and lambda2.
inner.val.method	Whether or not the plots of DIRECT algorithm should be shown.
show	Specification of how hyperparameters should be recoded or if no recoding should occur.
parms.coding	By specifying a seed, the results become reproducible. It is included here for the sake of those readers following along.
seed	Matrix of the input or predictor variables where the columns are the variables, and the rows are the observations.

models become more complex (Guo, Graber, McBurney, & Balasubramanian, 2010). We are aware of no sample size recommendations exist for a penalized SVM such as this. The `svmfs()` function can be applied in the following manner.

```
Bounds <- t(data.frame(log2lambda1=c(-10, 10),
    log2lambda2=c(-10,10)))
colnames(bounds)<-c("lower", "upper")
svm.model <- svmfs(x=predTrain, y = svmTrainOutcome,
    fs.method = "scad+L2", bounds=bounds,
    grid.search = "interval", inner.val.method = "cv",
    show = "none", parms.coding = "none",
    seed=123456)
```

The output of the model created using the `svmfs()` function has its own nomenclature that requires explanation. First, rather than referring to the coefficients as coefficients, the model uses the `w` parameter (coming from the term beta weight). The `b` parameter illustrates the intercept of the SVM hyperplane and can be thought of like the `b0` of a regression model. The `xind` parameter tells the user the index (or column location) of the variables selected in the dataset. The full output can be seen in Table 8. Two items selected by this model were MQ16 and S5_66. Variable MQ16 reads, "Once you are aware of the sound(s), because of the sound(s), how often do you have violent thoughts" and S5_66 reads, "Some sounds are so unbearable that I have shouted at people for making them, to make them stop". Both of these items are related to typical responses to triggers by those with misophonia and therefore make theoretical sense.

To examine the accuracy of this model, the same predict function can be used as was implemented previously, but the outputted predictions will require some restructuring, as they come in the form of a factor with underlying numeric values 1 and 2 and they need to have numeric values of 0 and 1. The Elastic SCAD SVM model obtained an accuracy of 0.83 and an F-score of 0.91. The code required to calculate that accuracy and F values are below.

Table 8. A table containing all calculated coefficients of all variables in the Elastic SCAD SVM.

Variable	Coefficient	Variable	Coefficient
(Intercept)	-1.209	S5_38	0.003
MQ3	0.002	S5_39	0.003
MQ5	0.003	S5_40	0.001
MQ8	0.003	S5_41	-0.002
MQ11	0.002	S5_42	0.002
MQ12	0.003	S5_43	0.002
MQ16	0.003	S5_49	0.002
MQ17	0.003	S5_53	-0.003
MQ18	0.002	S5_55	0.003
S5_1	0.001	S5_56	0.007
S5_2	0.005	S5_57	0.005
S5_7	0.006	S5_59	0.003
S5_10	-0.003	S5_65	-0.005
S5_11	-0.002	S5_66	0.001
S5_13	-0.001	S5_68	0.004
S5_24	-0.005	S5_69	0.001
S5_25	0.006	S5_72	0.001
S5_26	0.002	S5_74	0.005
S5_28	0.002	S5_75	0.007
S5_31	0.005	S5_78	-0.008
S5_32	0.005	S5_82	0.005
S5_35	0.005	S5_83	0.006
S5_37	0.002	S5_85	0.003

```
esvm.predictions <- predict(svm.model,
    newdata = svmTestPreds)
esvm.predictions.formatted <-
    as.numeric(esvm.predictions$pred.class)-1
esvm.accuracy <-
    mean(esvm.predictions.formatted == outcomeTest)
esvm.f1 <- F1_Score(esvm.predictions.formatted,
    outcomeTest)
```

Limitations of SVM include the researcher's selection of the kernel function, computation time, and dimension constraints. By default, the `svmfs()` function

utilizes a linear kernel function. Since the kernel is chosen a priori by the researcher, an optimal function must be used for optimal results. SVM models are computationally more expensive than a simpler classification technique (e.g., logistic regression) and will take longer to compute. SVM models face the same degree of freedom problem as LASSO and Elastic Net, which are limited by the number of observations. As such, an ideal dataset for SVM would contain more observations than variables.

2.3 Tree Based Models

Random Forest Another powerful classifier is a decision (or classification) tree (Breiman et al., 2017; Strobl et al., 2009). An example can be seen in Figure 4. From this decision tree, it can be concluded that anyone whose score on variable S5_57 is less than 3 and score on variable S5_60 is less than 3 does not qualify for a misophonia diagnosis. Decision trees are not only powerful classifiers, but they also produce an output that is easy to interpret. However, decision trees are prone to overfitting – so much so that overfitting is almost guaranteed (Bengio, Delalleau, & Simard, 2010). One of the most efficient ways to avoid overfitting is by using multiple trees (i.e., creating a random forest). Random forest creates many decision trees using a randomly selected subset of the data to create each individual tree. The results of all trees are then aggregated to predict the desired outcome. Some major benefits of a random forest classifier are that it can be used with an outcome variable that has any number of levels (Brieuc, Waters, Drinan, & Naish, 2018), meaning that unlike logistic regression, which only works with binary variables, random forest could handle a variable with 3, 4, or even 10 different levels. However, these trees are only used for classification, meaning that they do not perform variable selection. Thus, researchers have had to adapt the classifier to perform variable selection. The utilization of random forest in the *Boruta* package performs well in many different conditions (Kursa & Rudnicki, 2010), and, therefore, is the implementation demonstrated in this tutorial.

The Boruta package contains a series of functions pertaining to variable selection techniques using different measures of importance to select the variables. A measure of importance simply indicates a given variable's value to the model's overall strength. The more useful variables, meaning that they are stronger predictors of the outcome variable, are deemed more important and thus are more likely to be selected than those of lesser importance (i.e., less predictive power). Note that in this paper, mean decreased accuracy is the metric used to calculate variable importance. The Boruta package also has its own sample size suggestions. The original paper implementing the the package states that for typical problems, samples of 5-200 are often sufficient, assuming the number of true predictors is not extremely small compared to the total (Kursa & Rudnicki, 2010). It notes that as problems get more complex, the sample size should increase.

A simple regression formula statement is used to run the model: `outcome predictors`. Because all predictors will be used, a shortcut can be implemented using a period (`.`) in place of predictors as seen in the code below. If not, all variables were to be included in the model, the user would need to type all the

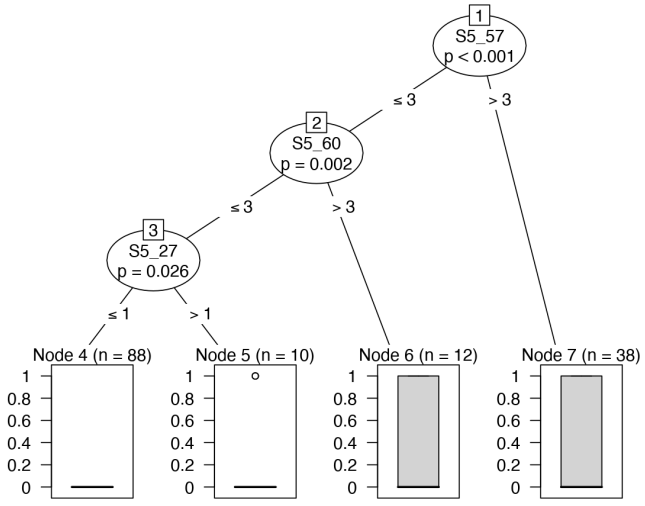


Figure 4. An example of a decision tree built on the misophonia data using the `ctree()` function.

relevant predictors names in the formula statement concatenated with addition symbols (+). Knowing this, the model can then be built using the following code:

```
set.seed(123456)
boruta.model <- Boruta(as.factor(MQDX) ~. ,
    data=trainDat)
```

The `Boruta()` function classifies variables as either important, unimportant, or of tentative importance. Regarding the misophonia dataset, 15 were deemed important, 74 were deemed unimportant, and the remaining 17 were placed in the tentative category. For a list of all variables that were classified in each category and a visualization of the `boruta.model` output, see Table 9. Figure 5 illustrates the variability of the importance score calculated for each variable during the Boruta process and their ultimate classification. A model can be built using either a) all variables that were not deemed unimportant (non-rejected variables) or b) only the confirmed important variables. For the purpose of this tutorial, only variables that have been confirmed important are included in the model. Two items that were confirmed important are MQ16 and S5_59. Variable MQ16 was justified in the SVM section as it pertains to having violent thoughts. S5_59 reads "If I cannot avoid certain sounds I feel helpless." Helplessness is often associated with anxiety (i.e., learned helplessness) which is often co-diagnosed with misophonia, and as such, this variable is theoretically justified.

This model is then built using the `randomForest()` function since Boruta implements a random forest model internally. The model is built in the following way.

Table 9. A table containing the classifications of importance for each variable as determined by the `Boruta()` function. Note that the implementation of Boruta used in this paper utilized mean decreased accuracy as the metric to calculate variable importance.

Variables	Items
Confirmed Important	MQ12, MQ13, MQ16, S5_3, S5_34, S5_35, S5_39, S5_40, S5_53, S5_56, S5_57, S5_59, S5_60, S5_67, S5_75
Rejected	MQ1, MQ2, MQ3, MQ4, MQ5, MQ6, MQ7, MQ8, MQ10, MQ11, MQ14, MQ15, MQ18, MQ20, S5_1, S5_4, S5_6, S5_7, S5_8, S5_9, S5_10, S5_11, S5_12, S5_13, S5_14, S5_15, S5_16, S5_17, S5_19, S5_20, S5_23, S5_24, S5_26, S5_28, S5_29, S5_30, S5_32, S5_33, S5_36, S5_37, S5_41, S5_42, S5_43, S5_44, S5_45, S5_46, S5_47, S5_48, S5_49, S5_50, S5_51, S5_52, S5_54, S5_55, S5_58, S5_64, S5_65, S5_66, S5_68, S5_70, S5_71, S5_72, S5_73, S5_74, S5_76, S5_77, S5_78, S5_79, S5_80, S5_82, S5_83, S5_84, S5_86, S5_87
Tentative	MQ17, MQ19, S5_2, S5_5, S5_18, S5_21, S5_22, S5_25, S5_27, S5_31, S5_38, S5_61, S5_62, S5_63, S5_69, S5_81, S5_85

```
set.seed(123456)
finalBoruta <- getConfirmedFormula(boruta.model)
selectedModel <- randomForest(finalBoruta,
    data=trainDat)
```

The predictive accuracy of the random forest model can be calculated using the `predict()` function, just as it has been for other models. An accuracy of .88 was obtained for this model and an F-score of 0.93. The algorithm may not perform well with highly unbalanced classifications or in situations where a given level contains a very small number of classifications.

2.4 Comparing All Models

For a comparison of the accuracy values and F-scores obtained by all techniques implemented in this tutorial, see Table 10. From this, we can state that the GA produced the most accurate model. However, there was no difference in the accuracy of the LASSO non-biased (e.g., the standard regression model built using variable selected via the LASSO), Boruta, and Elastic Net models. Depending on the purpose of your model, you may want to use a performance metric other than accuracy. Within the context of our motivating example, it may be worth examining the following:

- **Sensitivity**: Given the individual truly has misophonia, how likely is the classifier to realize that?
- **Specificity**: Given the individual truly does not have misophonia, how likely is the classifier to realize that?

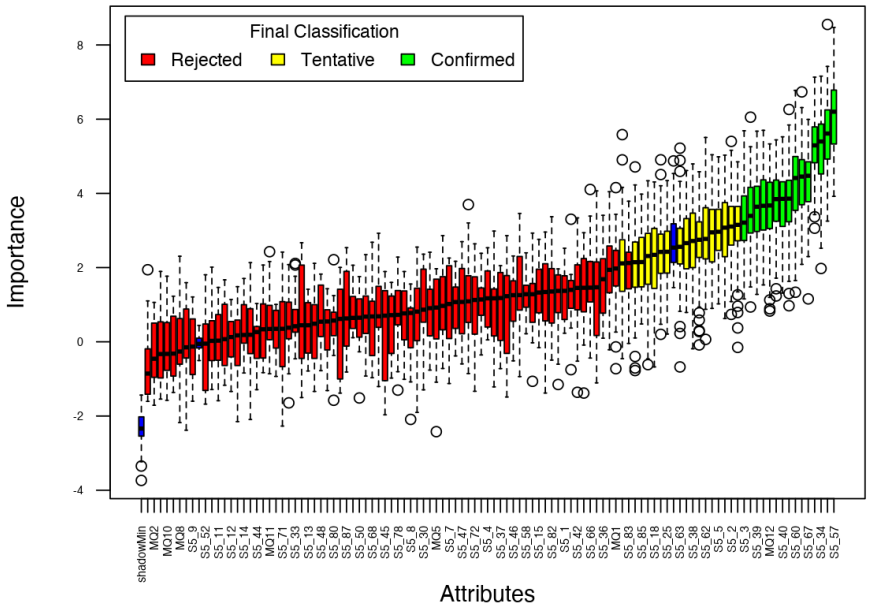


Figure 5. A plot containing the Z-score transformed estimates of variable importance scores for each variable in the `Boruta()` model. Blue boxplots correspond to minimal, average, and maximum Z-scores of a shadow attribute. Red and green boxplots represent Z-scores of rejected and confirmed attributes respectively. Note that the implementation of Boruta used in this paper utilized mean decreased accuracy as the metric to calculate variable importance.

- **Positive predictive value**: Given the classifier claims the individual to have misophonia, how likely is it that the individual really has misophonia?
- **Negative predictive value**: Given the classifier claims the individual does not have misophonia, how likely is it that the individual really does not have the disease?

While accuracy serves as a useful general indicator of model performance, it can be misleading, particularly when dealing with unbalanced datasets where one class is significantly more prevalent than the other. In such cases, a model can achieve high accuracy by simply predicting the majority class, even if it performs poorly on the minority class. Therefore, it's essential to consider alternative performance metrics that provide a more nuanced understanding of a model's strengths and weaknesses. For instance, sensitivity (the true positive rate) measures the proportion of actual positives that are correctly identified, while specificity (the true negative rate) quantifies the proportion of actual neg-

Table 10. A table containing the predictive accuracy values obtained by all models built in this tutorial paper. Methods are listed such that the accuracy values are ordered from least accurate to most accurate. Significance is determined relative to the previous model (i.e., Elastic SCAD SVM was determined to have a statistically significant better accuracy than Elastic Net non-biased) according to a McNemar's Chi-squared test with continuity correction. Note significant differences were not evaluated for F-scores.

Method	**Cross-validated Accuracy**	**Cross-validated F-Score**
Elastic Net non-biased	0.797	0.881
Elastic SCAD SVM	0.828**	0.905
LASSO	0.859**	0.918
Elastic Net	0.875	0.938
Boruta	0.875	0.930
LASSO non-biased	0.891	0.916
GA	1***	1

Note: * $p < .05$, ** $p < .01$, *** $p < .0001$

atives that are correctly classified. These metrics are crucial when the cost of misclassification differs for each class, such as in medical diagnosis where failing to identify a true case (low sensitivity) can have more severe consequences than a false positive (low specificity). Precision reflects the proportion of predicted positives that are actually positive, while recall is synonymous with sensitivity. Another valuable metric is the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, which comprehensively measures a model's ability to discriminate between classes across various thresholds. Researchers should carefully consider their research question's specific goals and context to select the most appropriate performance metrics, ensuring a balanced and insightful evaluation of their models.

Given that our example pertains to diagnosis, it is possible that one may favor sensitivity over specificity in that we want to minimize the number of missed cases. However, it is also possible that we would want to minimize the number of false diagnoses to save individuals the cost of unnecessary intervention. A confusion matrix (discussed briefly in Appendix B) might be useful. Alternatively, one could use the AUC of the ROC curve. One should carefully consider these factors when deciding on the performance metric by which to evaluate a model.

Examining the selected variables reveals interesting method-dependent patterns. Elastic SCAD SVM selected many more variables than LASSO, but had a worse accuracy. Given this outcome, it may not be ideal to use all variables selected by Elastic SCAD SVM in this dataset. There was only one variable (S5_57) that was selected by all five methods. So, there is a clear method effect on the variables that are deemed to be important. Within the context of our example, we could interpret this to mean that the question, "Sometimes in response to sounds, I feel rage that is difficult to control," is an incredibly important predictor for misophonia and may capture a defining characteristic of the disorder. Beyond improving predictive accuracy, understanding why certain

variables are deemed important can provide valuable insights into the underlying mechanisms or factors driving the outcome of interest. This insight could guide future research exploring the role of emotional regulation in misophonia and potentially inform the development of targeted interventions. Furthermore, the identification of unexpected or previously overlooked variables as important predictors can spark new research questions and hypotheses. This iterative process of variable selection, model building, and hypothesis generation can lead to a more nuanced and comprehensive understanding of complex phenomena. By carefully examining the selected variables, researchers can generate hypotheses, refine theoretical models, and ultimately gain a deeper understanding of complex human behavior and health outcomes.

3 Discussion

This tutorial provided an overview and a practical guide for the implementation of LASSO (Friedman et al., 2010), Elastic Net (Friedman et al., 2010), a genetic algorithm (Scrucca, 2013, 2017), Elastic SCAD SVM (Becker et al., 2009), and random forest via Boruta (Kursa & Rudnicki, 2010) in R v. 4.2.1. Proper analysis of the output as well as comparisons on the predictive accuracy of each method are also discussed. More information on R, other useful machine learning software, and some of these functions were provided in the Appendices. Lastly, an OSF project containing all code implemented in this tutorial, additional code the reader may find useful, and the data used is available. For a full link to the project, see the availability of data and materials section of this paper.

Variable selection allows researchers to find parsimonious models that are also good predictive or classifying models. Given R's increasing popularity among researchers due to the software's free and open access nature, it is valuable to the field to provide more guidance on the variable selection methods available in R. In addition, the extent to which some of these methods overfit data should not be ignored when implementing them on real-world data. Suppose a researcher is concerned with creating a generalizable model. In that case, it is recommended that the results be validated not only through some form of cross-validation but also through the collection of a new sample. Through this tutorial, we aim to push the field towards more transparent guidelines and standardization for the use of variable selection techniques and machine learning in psychological research.

While variable selection offers numerous advantages, it's crucial to acknowledge its potential ethical implications, particularly in sensitive applications like clinical diagnosis or risk assessment (Obermeyer, Powers, Vogeli, & Mullainathan, 2019). If biased or incomplete data is used for training, variable selection algorithms can perpetuate and even amplify existing societal biases, leading to unfair or discriminatory outcomes (Mehrabi, Morstatter, Saxena, Lerman, & Galstyan, 2021). For example, if a dataset used to predict criminal recidivism is skewed towards certain demographics, the selected variables might unfairly target individuals from those groups, even if the variables are not causally related to re-

cidivism. Similarly, in clinical diagnosis, relying on variables that are correlated with social determinants of health rather than underlying biological mechanisms could result in misdiagnosis or inadequate treatment for marginalized populations (Vyas, Eisenstein, & Jones, 2020). Therefore, researchers must carefully consider the potential for bias in their data and strive to develop and implement variable selection techniques that prioritize fairness and equity.

Beyond enhancing model performance, variable selection holds significant potential for translational impact in the social and health sciences. By identifying the most influential predictors, researchers can better understand the underlying mechanisms driving complex phenomena, leading to more effective interventions, treatments, and public health strategies. For instance, in personalized medicine, variable selection can help tailor treatments to individual patients based on their unique genetic, environmental, and lifestyle factors . Identifying key risk factors for chronic diseases through variable selection in public health can inform targeted prevention programs and resource allocation strategies. Moreover, in developing psychological interventions, variable selection can aid in identifying the most effective treatment components and tailoring therapies to specific patient needs and characteristics (Vyas et al., 2020). By focusing research and interventions on the most impactful variables, variable selection can contribute more effective and efficient solutions to pressing social and health challenges.

There are many ways a researcher can define accuracy. When interested in classification, an optimal model is one with minimal classification error, as we have highlighted throughout this tutorial (Huang, 2015). However, previous research notes that if classification is not the goal, minimal error can be conceptualized as selecting variables with the highest relevance to the given outcome (Peng, Long, & Ding, 2005). With this in mind, it is important that variables are not falsely discovered (i.e., a variable that is not relevant is selected; Type I error in selection). An interested reader is pointed to the *knockoff* package (Candés, Fan, Janson, & Lv, 2018) and work by Zimmermann, Baillie, Kormaksson, Ohlssen, and Sechidis (2024). Another important aspect of variable selection, especially for the applied researcher, is the stability of a model (i.e., how robust a particular model is to small changes in the data). We discussed one way to address this concern through the concept of cross-validation, however, there are additional ways one might go about addressing this concern (Bommert & Lang, 2021; Nogueira, Sechidis, & Brown, 2018). The field would benefit from additional tutorial papers discussing the balance of these issues with accuracy to help guide the applied researcher.

Many additional R packages will perform variable selection using random forest as well as SVMs, but only one of each was demonstrated in this tutorial. The demonstrated methods in the current tutorial were selected because they are commonly used in the psychological sciences, are powerful techniques for classification (e.g., diagnosing individuals with misophonia) and variable selection, and are all freely available to researchers in R. In a similar vein, we have included only five machine learning methods here but many more exist, and additional tutorials should be provided to applied researchers about how best to implement

them following research demonstrating each algorithm's performance to indicate which algorithm is best for addressing certain research questions. For the interested reader, a comparison of the performance of each method demonstrated in this tutorial can be found in Bain et al. (2023).

4 Conclusion

This tutorial presented an overview and a practical guide for implementing five variable selection techniques: LASSO (Friedman et al., 2010), Elastic Net (Friedman et al., 2010), a genetic algorithm (Scrucca, 2013, 2017), Elastic SCAD SVM (Becker et al., 2009), and random forest via Boruta (Kursa & Rudnicki, 2010) in R. Proper analysis of the output as well as comparisons on the predictive accuracy of each method are also discussed. More information on R, other useful machine learning software, and some of these functions were provided in the Appendices. Lastly, an OSF project containing all code implemented in this tutorial, additional code the reader may find useful, and the data used is available. For a full link to the project, see the availability of data and materials section of this paper.

This paper highlighted the increasing availability of large and complex datasets in the social and health sciences, requiring a move beyond traditional variable selection techniques like stepwise regression. This tutorial demonstrates that modern machine learning methods offer powerful and accessible alternatives for identifying the most informative variables, improving model accuracy, and gaining a deeper understanding of complex phenomena. By embracing these advancements and continuing to explore the ethical and interpretive dimensions of variable selection, researchers can enhance the rigor, reproducibility, and, ultimately, the translational impact of their work. We encourage readers to consult the documentation for each method for further examples and details. The user is to refer to each method's full documentation for additional examples and details. We hope that this tutorial makes these methods more easily accessible to the everyday psychological researcher, opens doors to applications of variable selection in new areas, and leads to a decreased presence of less powerful methods (e.g., stepwise selection) in the literature.

Availability of Data and Materials

The accompanying code and data utilized in this tutorial can be found here: https://osf.io/pr6j8/?view_only=c778e322f1d54429990067580e615afb. Additional supplementary information such as a glossary of key terms, R package recommendations, etc. are also available through OSF.

Authors' Contributions

CMB conducted all analyses and drafted the manuscript; DS contributed to manuscript draft and recreation from its original form, and supervised manuscript

preparation. YMB contributed to manuscript draft and recreation from its old form. Regarding the data utilized here, LEE conceived of the study design of the project and supervised all aspects of funding, participant recruitment, and data collection while JEN aided in the original study design, led all data collection and data preprocessing; JEL supervised data analysis and manuscript preparation. All authors contributed significantly to manuscript preparation.

References

Adams, L. J., Bello, G., & Dumancas, G. G. (2015). Development and Application of a Genetic Algorithm for Variable Optimization and Predictive Modeling of Five-Year Mortality Using Questionnaire Data. *Bioinformatics and Biology Insights*, *9s3*, BBI.S29469. doi: https://doi.org/10.4137/BBI.S29469

Alamri, L. H., Almuslim, R. S., Alotibi, M. S., Alkadi, D. K., Ullah Khan, I., & Aslam, N. (2021). Predicting Student Academic Performance using Support Vector Machine and Random Forest. In *Proceedings of the 2020 3rd International Conference on Education Technology Management* (pp. 100–107). New York, NY, USA: Association for Computing Machinery. doi: https://doi.org/10.1145/3446590.3446607

Algamal, Z. Y., & Lee, M. H. (2015). Applying Penalized Binary Logistic Regression with Correlation Based Elastic Net for Variables Selection. *Journal of Modern Applied Statistical Methods*, *14*(1), 168–179. doi: https://doi.org/10.22237/jmasm/1430453640

Amene, E., Hanson, L. A., Zahn, E. A., Wild, S. R., & Döpfer, D. (2016). Variable selection and regression analysis for the prediction of mortality rates associated with foodborne diseases. *Epidemiology and Infection*, *144*(9), 1959–1973.

Aragón-Royón, F., Jiménez-Vílchez, A., Arauzo-Azofra, A., & Benítez, J. M. (2020). *FSinR: an exhaustive package for feature selection.* arXiv.

Arjomandi-Nezhad, A., Guo, Y., Pal, B. C., & Varagnolo, D. (2023). *A Model Predictive Approach for Enhancing Transient Stability of Grid-Forming Converters.* arXiv.

Bain, C., Shi, D., Boness, C. L., & Loeffelman, J. (2023). *A Simulation Study Comparing the Use of Supervised Machine Learning Variable Selection Methods in the Psychological Sciences.* PsyArXiv. doi: https://doi.org/10.31234/osf.io/y53t6

Barceló, P., Monet, M., Pérez, J., & Subercaseaux, B. (2020). Model interpretability through the lens of computational complexity. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (pp. 15487–15498). Red Hook, NY, USA: Curran Associates Inc.

Basarkod, G., Sahdra, B., & Ciarrochi, J. (2018). *Body Image-Acceptance and Action Questionnaire-5: An Abbreviation Using Genetic Algorithms* (Tech. Rep.).

Battineni, G., Chintalapudi, N., & Amenta, F. (2019). Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). *Informatics in Medicine Unlocked*, *16*, 100200. doi: https://doi.org/10.1016/j.imu.2019.100200

Becker, N., Toedt, G., Lichter, P., & Benner, A. (2011). Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. *BMC Bioinformatics*, *12*(1), 138. doi: https://doi.org/10.1186/1471-2105-12-138

Becker, N., Werft, W., Toedt, G., Lichter, P., & Benner, A. (2009). penalizedSVM: a R-package for feature selection SVM classification. *Bioinformatics*, *25*(13), 1711–1712. doi: https://doi.org/10.1093/bioinformatics/btp286

Bengio, Y., Delalleau, O., & Simard, C. (2010). Decision Trees Do Not Generalize to New Variations. *Computational Intelligence*, *26*(4), 449–467. doi: https://doi.org/10.1111/j.1467-8640.2010.00366.x

Bierman, S., & Steel, S. (2009). Variable Selection for Support Vector Machines. *Communications in Statistics - Simulation and Computation*, *38*(8), 1640–1658. doi: https://doi.org/10.1080/03610910903072391

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*(1), 245–271. doi: https://doi.org/10.1016/S0004-3702(97)00063-5

Bommert, A., & Lang, M. (2021). stabm: Stability Measures for Feature Selection. *Journal of Open Source Software*, *6*(59), 3010. doi: https://doi.org/10.21105/joss.03010

Bourdès, V., Bonnevay, S., Lisboa, P., Defrance, R., Pérol, D., Chabaud, S., ... Négrier, S. (2010). Comparison of Artificial Neural Network with Logistic Regression as Classification Models for Variable Selection for Prediction of Breast Cancer Patient Outcomes. *Advances in Artificial Neural Systems*, *2010*, 1–11. doi: https://doi.org/10.1155/2010/309841

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). *Classification and Regression Trees*. New York: Chapman and Hall/CRC. doi: https://doi.org/10.1201/9781315139470

Brieuc, M. S. O., Waters, C. D., Drinan, D. P., & Naish, K. A. (2018). A practical introduction to Random Forest for genetic association studies in ecology and evolution. *Molecular Ecology Resources*, *18*(4), 755–766. doi: https://doi.org/10.1111/1755-0998.12773

Cafri, G., Li, L., Paxton, E. W., & Fan, J. (2018). Predicting risk for adverse health events using random forest. *Journal of Applied Statistics*, *45*(12), 2279–2294. doi: https://doi.org/10.1080/02664763.2017.1414166

Calcagno, V., & Mazancourt, C. D. (2010). **glmulti** : An *R* Package for Easy Automated Model Selection with (Generalized) Linear Models. *Journal of Statistical Software*, *34*(12). doi: https://doi.org/10.18637/jss.v034.i12

Candés, E., Fan, Y., Janson, L., & Lv, J. (2018). Panning for Gold: 'Model-X' Knockoffs for High Dimensional Controlled Variable Selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *80*(3).

Cateni, S., Colla, V., & Vannucci, M. (2010). Variable Selection through Genetic Algorithms for Classification Purposes. In *Artificial Intelligence and Applications.* Innsbruck, Austria: ACTAPRESS. doi: https://doi.org/10.2316/P.2010.674-080

Chen, Y., & Yang, Y. (2021). The One Standard Error Rule for Model Selection: Does It Work? *Stats*, *4*(4), 868–892. doi: https://doi.org/10.3390/stats4040051

Chowdhury, M. Z. I., & Turin, T. C. (2020). Variable selection strategies and its importance in clinical prediction modelling. *Family Medicine and Community Health*, *8*(1), e000262. doi: https://doi.org/10.1136/fmch-2019-000262

Chu, M., Fang, Z., Mao, L., Ma, H., Lee, C.-Y., & Chiang, Y.-C. (2024). Creating A child-friendly social environment for fewer conduct problems and more prosocial behaviors among children: A LASSO regression approach. *Acta Psychologica*, *244*, 104200. doi: https://doi.org/10.1016/j.actpsy.2024.104200

Derksen, S., & Keselman, H. J. (1992). Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, *45*(2), 265–282. doi: https://doi.org/10.1111/j.2044-8317.1992.tb00992.x

El Haouij, N., Poggi, J.-M., Ghozi, R., Sevestre-Ghalila, S., Jaïdane, M., Poggi Jean-Michel, J.-M., & El Haouij, N. (2018). Random forest-based approach for physiological functional variable selection for driver's stress level classification. *Statistical Methods & Applications*. doi: https://doi.org/10.1007/s10260-018-0423-5

Engebretsen, S., & Bohlin, J. (2019). Statistical predictions with glmnet. *Clinical Epigenetics*, *11*(1), 123. doi: https://doi.org/10.1186/s13148-019-0730-1

Fernandez, M., Caballero, J., Fernandez, L., & Sarai, A. (2011). Genetic algorithm optimization in drug design QSAR: Bayesian-regularized genetic neural networks (BRGNN) and genetic algorithm-optimized support vectors machines (GA-SVM). *Molecular Diversity*, *15*(1), 269–289. doi: https://doi.org/10.1007/s11030-010-9234-9

Figueroa, R. L., Zeng-Treitler, Q., Kandula, S., & Ngo, L. H. (2012). Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, *12*(1), 8. doi: https://doi.org/10.1186/1472-6947-12-8

Foucart, S., Tadmor, E., & Zhong, M. (2023). On the Sparsity of LASSO Minimizers in Sparse Data Recovery. *Constructive Approximation*, *57*(2), 901–919. doi: https://doi.org/10.1007/s00365-022-09594-1

Fox, E. W., Hill, R. A., Leibowitz, S. G., Olsen, A. R., Thornbrugh, D. J., & Weber, M. H. (2017). Assessing the accuracy and stability of variable selection methods for random forest modeling in ecology. *Environmental Monitoring and Assessment*, *189*(7), 316. doi: https://doi.org/10.1007/s10661-017-6025-0

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for

Generalized Linear Models via Coordinate Descent. *Journal of statistical software*, *33*(1), 1–22.

Frohlich, H., Chapelle, O., & Scholkopf, B. (2003). Feature selection for support vector machines by means of genetic algorithm. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence* (pp. 142–148). Sacramento, CA, USA: IEEE Comput. Soc. doi: https://doi.org/10.1109/TAI.2003.1250182

Gan, C. C., & Learmonth, G. (2016). *Developing an ICU scoring system with interaction terms using a genetic algorithm.* arXiv. doi: https://doi.org/10.48550/arXiv.1604.06730

Genuer, R., Poggi, J.-M., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, *31*(14), 2225–2236. doi: https://doi.org/10.1016/j.patrec.2010.03.014

Ghojogh, B., & Crowley, M. (2023). *The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial.* arXiv. doi: https://doi.org/10.48550/arXiv.1905.12787

Gunn, H. J., Hayati Rezvan, P., Fernández, M. I., & Comulada, W. S. (2023). How to apply variable selection machine learning algorithms with multiply imputed data: A missing discussion. *Psychological Methods*, *28*(2), 452–471. doi: https://doi.org/10.1037/met0000478

Guo, Y., Graber, A., McBurney, R. N., & Balasubramanian, R. (2010). Sample size and statistical power considerations in high-dimensionality data settings: a comparative study of classification algorithms. *BMC Bioinformatics*, *11*(1), 447. doi: https://doi.org/10.1186/1471-2105-11-447

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(7-8), 1157–1182. doi: https://doi.org/10.1162/153244303322753616

Han, H., & Dawson, K. J. (2021). Applying elastic-net regression to identify the best models predicting changes in civic purpose during the emerging adulthood. *Journal of Adolescence*, *93*, 20–27. doi: https://doi.org/10.1016/j.adolescence.2021.09.011

Heinze, G., Wallisch, C., & Dunkler, D. (2018). Variable selection – A review and recommendations for the practicing statistician. *Biometrical Journal*, *60*(3), 431–449. doi: https://doi.org/10.1002/bimj.201700067

Helwig, N. E. (2017). Adding bias to reduce variance in psychological results: A tutorial on penalized regression. *The Quantitative Methods for Psychology*, *13*(1), 1–19. doi: https://doi.org/10.20982/tqmp.13.1.p001

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, *12*(1), 55–67. doi: https://doi.org/10.2307/1267351

Huang, S. H. (2015). Supervised feature selection: A tutorial. *Artificial Intelligence Research*, *4*(2), p22. doi: https://doi.org/10.5430/air.v4n2p22

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218. doi: https://doi.org/10.1007/BF01908075

Iwendi, C., Bashir, A. K., Peshkar, A., Sujatha, R., Chatterjee, J. M., Pa-

supuleti, S., ... Jo, O. (2020). COVID-19 Patient Health Prediction Using Boosted Random Forest Algorithm. *Frontiers in Public Health*, *8*. doi: https://doi.org/10.3389/fpubh.2020.00357

Jia, W., Sun, M., Lian, J., & Hou, S. (2022). Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, *8*(3), 2663–2693. doi: https://doi.org/10.1007/s40747-021-00637-x

Karatzoglou, A., Meyer, D., & Hornik, K. (2006). Support Vector Machines in R. *Journal of Statistical Software*, *15*(9). doi: https://doi.org/10.18637/jss.v015.i09

Kerkhoff, D., & Nussbeck, F. W. (2019). The Influence of Sample Size on Parameter Estimates in Three-Level Random-Effects Models. *Frontiers in Psychology*, *10*. doi: https://doi.org/10.3389/fpsyg.2019.01067

Kirpich, A., Ainsworth, E. A., Wedow, J. M., Newman, J. R. B., Michailidis, G., & McIntyre, L. M. (2018). Variable selection in omics data: A practical evaluation of small sample sizes. *PLOS ONE*, *13*(6), e0197910. doi: https://doi.org/10.1371/journal.pone.0197910

Kohavi, R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 202–207). Portland, Oregon: AAAI Press.

Kok, B. C., Choi, J. S., Oh, H., & Choi, J. Y. (2021). Sparse Extended Redundancy Analysis: Variable Selection via the Exclusive LASSO. *Multivariate Behavioral Research*, *56*(3), 426–446. doi: https://doi.org/10.1080/00273171.2019.1694477

Kuchirko, Y., Bennet, A., Halim, M. L., Costanzo, P., & Ruble, D. (2021). The Influence of Siblings on Ethnically Diverse Children's Gender Typing Across Early Development. *Developmental Psychology*. doi: https://doi.org/10.1037/dev0001173.supp

Kursa, M. B., & Rudnicki, W. R. (2010). Feature Selection with the **Boruta** Package. *Journal of Statistical Software*, *36*(11). doi: https://doi.org/10.18637/jss.v036.i11

Leardi, R. (2000). Application of genetic algorithm-PLS for feature selection in spectral data sets. *Journal of Chemometrics*, *14*(5-6), 643–655. doi: https://doi.org/10.1002/1099-128X(200009/12)14:5/6¡643::AID-CEM621¿3.0.CO;2-E

Lenters, V., Vermeulen, R., & Portengen, L. (2018). Performance of variable selection methods for assessing the health effects of correlated exposures in case–control studies. *Occupational and Environmental Medicine*, *75*(7), 522–529. doi: https://doi.org/10.1136/oemed-2016-104231

Liu, X., Cao, P., Gonçalves, A. R., Zhao, D., & Banerjee, A. (2018). Modeling Alzheimer's Disease Progression with Fused Laplacian Sparse Group Lasso. *ACM Transactions on Knowledge Discovery from Data*, *12*(6), 65:1–65:35. doi: https://doi.org/10.1145/3230668

Loef, B., Wong, A., Janssen, N. A. H., Strak, M., Hoekstra, J., Picavet, H. S. J., ... Herber, G.-C. M. (2022). Using random forest to identify longitudinal

predictors of health in a 30-year cohort study. *Scientific Reports*, *12*(1), 10372. doi: https://doi.org/10.1038/s41598-022-14632-w

Loughrey, J., & Cunningham, P. (2005). Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets. In M. Bramer, F. Coenen, & T. Allen (Eds.), *Research and Development in Intelligent Systems XXI* (pp. 33–43). London: Springer London. doi: https://doi.org/10.1007/1-84628-102-4_3

Luo, J., Ren, S., Li, Y., & Liu, T. (2021). The Effect of College Students' Adaptability on Nomophobia: Based on Lasso Regression. *Frontiers in Psychiatry*, *12*. doi: https://doi.org/10.3389/fpsyt.2021.641417

Marafino, B. J., John Boscardin, W., & Adams Dudley, R. (2015). Efficient and sparse feature selection for biomedical text classification via the elastic net: Application to ICU risk stratification from nursing notes. *Journal of Biomedical Informatics*, *54*, 114–120. doi: https://doi.org/10.1016/j.jbi.2015.02.003

McDonald, G. C. (2009). Ridge regression. *WIREs Computational Statistics*, *1*(1), 93–100. doi: https://doi.org/10.1002/wics.14

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.*, *54*(6), 115:1–115:35. doi: https://doi.org/10.1145/3457607

Mendez-Civieta, A., Aguilera-Morillo, M. C., & Lillo, R. E. (2021). Adaptive sparse group LASSO in quantile regression. *Advances in Data Analysis and Classification*, *15*(3), 547–573. doi: https://doi.org/10.1007/s11634-020-00413-8

ML | Underfitting and Overfitting. (2017, November). Retrieved 2024-07-10, from `https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/` (Section: Machine Learning)

Moore, J. H., Andrews, P. C., Olson, R. S., Carlson, S. E., Larock, C. R., Bulhoes, M. J., ... Armentrout, S. L. (2017). Grid-based stochastic search for hierarchical gene-gene interactions in population-based genetic studies of common human diseases. *BioData Mining*, *10*(1), 19. doi: https://doi.org/10.1186/s13040-017-0139-3

Nogueira, S., Sechidis, K., & Brown, G. (2018). On the Stability of Feature Selection Algorithms. *Journal of Machine Learning Research*, *18*(174), 1–54.

Norris, J. E., Kimball, S. H., Nemri, D. C., & Ethridge, L. E. (2022). Toward a Multidimensional Understanding of Misophonia Using Cluster-Based Phenotyping. *Frontiers in Neuroscience*, *16*. doi: https://doi.org/https://doi.org/10.3389/fnins.2022.832516

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, *366*(6464), 447–453. doi: https://doi.org/10.1126/science.aax2342

Peduzzi, P., Concato, J., Feinstein, A. R., & Holford, T. R. (1995). Importance of events per independent variable in proportional hazards regression analysis. II. Accuracy and precision of regression estimates. *Journal of Clin-*

ical Epidemiology, *48*(12), 1503–1510. doi: https://doi.org/10.1016/0895-4356(95)00048-8

Peduzzi, P., Concato, J., Kemper, E., Holford, T. R., & Feinstein, A. R. (1996). A simulation study of the number of events per variable in logistic regression analysis. *Journal of Clinical Epidemiology*, *49*(12), 1373–1379. doi: https://doi.org/10.1016/s0895-4356(96)00236-3

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. doi: https://doi.org/10.1109/TPAMI.2005.159

Pratik, S., Nayak, D., Prasath, R., & Swarnkar, T. (2022). Prediction of Smoking Addiction Among Youths Using Elastic Net and KNN: A Machine Learning Approach. In (pp. 199–209). doi: https://doi.org/10.1007/978-3-031-21517-9_20

Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software*, *53*(4), 1–37. doi: https://doi.org/10.18637/jss.v053.i04

Scrucca, L. (2017). On some extensions to GA package: Hybrid optimisation, parallelisation and islands evolution. *R Journal*, *9*(1), 187–206. doi: https://doi.org/10.32614/rj-2017-008

Scrucca, L., Fop, M., Murphy, T., Brendan, & Raftery, A., E. (2016). mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal*, *8*(1), 289. doi: https://doi.org/10.32614/RJ-2016-021

Serang, S., Jacobucci, R., Brimhall, K. C., & Grimm, K. J. (2017). Exploratory Mediation Analysis via Regularization. *Structural equation modeling : a multidisciplinary journal*, *24*(5), 733–744. doi: https://doi.org/10.1080/10705511.2017.1311775

Shi, D., Shi, D., & Fairchild, A. J. (2023). Variable Selection for Mediators under a Bayesian Mediation Model. *Structural Equation Modeling: A Multidisciplinary Journal*, *0*(0), 1–14. doi: https://doi.org/10.1080/10705511.2022.2164285

Singla, M., & Shukla, K. K. (2020). Robust statistics-based support vector machine and its variants: a survey. *Neural Computing and Applications*, *32*(15), 11173–11194. doi: https://doi.org/10.1007/s00521-019-04627-6

Smith, G. (2018). Step away from stepwise. *Journal of Big Data*, *5*(1), 32. doi: https://doi.org/10.1186/s40537-018-0143-6

Song, Q. C., Tang, C., & Wee, S. (2021). Making Sense of Model Generalizability: A Tutorial on Cross-Validation in R and Shiny. *Advances in Methods and Practices in Psychological Science*, *4*(1), 2515245920947067. doi: https://doi.org/10.1177/2515245920947067

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, *14*(4), 323–348. doi: https://doi.org/10.1037/a0016973

Tan, L., Main, J. B., & Darolia, R. (2021). Using random forest analysis to identify student demographic and high school-level factors that predict college engineering major choice. *Journal of Engineering Education*, *110*(3), 572–593. doi: https://doi.org/10.1002/jee.20393

Tay, J. K., Narasimhan, B., & Hastie, T. (2023). Elastic Net Regularization Paths for All Generalized Linear Models. *Journal of Statistical Software*, *106*(1). doi: https://doi.org/10.18637/jss.v106.i01

Tharwat, A., & Hassanien, A. E. (2019). Quantum-Behaved Particle Swarm Optimization for Parameter Optimization of Support Vector Machine. *Journal of Classification*, *36*, 576–598. doi: https://doi.org/10.1007/s00357-018-9299-1

Thompson, B. (1995). Stepwise Regression and Stepwise Discriminant Analysis Need Not Apply here: A Guidelines Editorial. *Educational and Psychological Measurement*, *55*(4), 525–534. doi: https://doi.org/10.1177/0013164495055004001

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288.

Trevino, V., & Falciani, F. (2006). GALGO: an R package for multivariate variable selection using genetic algorithms. *Bioinformatics*, *22*(9), 1154–1156. doi: https://doi.org/10.1093/bioinformatics/btl074

van Vuuren, C. L., van Mens, K., de Beurs, D., Lokkerbol, J., van der Wal, M. F., Cuijpers, P., & Chinapaw, M. J. M. (2021). Comparing machine learning to a rule-based approach for predicting suicidal behavior among adolescents: Results from a longitudinal population-based survey. *Journal of Affective Disorders*, *295*, 1415–1420. doi: https://doi.org/10.1016/j.jad.2021.09.018

Vyas, D. A., Eisenstein, L. G., & Jones, D. S. (2020). Hidden in Plain Sight — Reconsidering the Use of Race Correction in Clinical Algorithms. *New England Journal of Medicine*, *383*(9), 874–882. doi: https://doi.org/10.1056/NEJMms2004740

Wang, L., Cheng, H., Liu, Z., & Zhu, C. (2014). A robust elastic net approach for feature learning. *Journal of Visual Communication and Image Representation*, *25*(2), 313–321. doi: https://doi.org/10.1016/j.jvcir.2013.11.002

Wehrens, R., & Franceschi, P. (2012). Meta-statistics for variable selection: The R package BioMark. *Journal of Statistical Software*, *51*(10). doi: https://doi.org/10.18637/jss.v051.i10

Wen, Q., Mustafi, S. M., Li, J., Risacher, S. L., Tallman, E., Brown, S. A., ... Wu, Y.-C. (2019). White matter alterations in early-stage Alzheimer's disease: A tract-specific study. *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, *11*, 576–587. doi: https://doi.org/10.1016/j.dadm.2019.06.003

Wettstein, A., Jenni, G., Schneider, I., Kühne, F., grosse Holtforth, M., & La Marca, R. (2023). Predictors of Psychological Strain and Allostatic Load in Teachers: Examining the Long-Term Effects of Biopsychosocial Risk and Protective Factors Using a LASSO Regression Approach. *In-*

ternational Journal of Environmental Research and Public Health, *20*(10), 5760. doi: https://doi.org/10.3390/ijerph20105760

Whittingham, M. J., Stephens, P. A., Bradbury, R. B., & Freckleton, R. P. (2006). Why do we still use stepwise modelling in ecology and behaviour? *Journal of Animal Ecology*, *75*(5), 1182–1189. doi: https://doi.org/10.1111/j.1365-2656.2006.01141.x

Wickham, H., François, R., Henry, L., Müller, K., Vaughan, D., Software, P., & PBC. (2023). *dplyr: A Grammar of Data Manipulation.*

Wiegand, R. E. (2010). Performance of using multiple stepwise algorithms for variable selection. *Statistics in Medicine*, *29*(15), 1647–1659. doi: https://doi.org/10.1002/sim.3943

Wu, M. S., Lewin, A. B., Murphy, T. K., & Storch, E. A. (2014). Misophonia: Incidence, Phenomenology, and Clinical Correlates in an Undergraduate Student Sample: Misophonia. *Journal of Clinical Psychology*, *70*(10), 994–1007. doi: https://doi.org/10.1002/jclp.22098

Xu, H., Caramanis, C., & Mannor, S. (2009). Robustness and Regularization of Support Vector Machines. *Journal of Machine Learning Research 1*, *10*, 1485–1510.

Yan, Y. (2024). *MLmetrics: Machine Learning Evaluation Metrics.*

Yarkoni, T., & Westfall, J. (2017). Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning. *Perspectives on Psychological Science*, *12*(6), 1100–1122. doi: https://doi.org/10.1177/1745691617693393

Yukselturk, E., Ozekes, S., & Türel, Y. K. (2014). Predicting Dropout Student: An Application of Data Mining Methods in an Online Education Program. *European Journal of Open, Distance and E-Learning*, *17*(1), 118–133. doi: https://doi.org/10.2478/eurodl-2014-0008

Zimmermann, M. R., Baillie, M., Kormaksson, M., Ohlssen, D., & Sechidis, K. (2024). All that Glitters Is not Gold: Type-I Error Controlled Variable Selection from Clinical Trial Data. *Clinical Pharmacology & Therapeutics*, *115*(4), 774–785. doi: https://doi.org/10.1002/cpt.3211

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301–320. doi: https://doi.org/10.1111/j.1467-9868.2005.00503.x

Appendix A

Table 1. Demographic information for the sample used in the illustrative example.

Variable	n (%)
Age (Years)	M = 18.96, SD = 1.7
Gender	
Male	104 (30.3%)
Female	239 (69.7%)
Ethnicity	
White	263 (76.7%)
Black/African American	32 (9.3%)
Latino/Hispanic	46 (13.4%)
Asian/Asian American	28 (8.2%)
American Indian/Alaska Native	26 (7.6%)
Native Hawaiian/Other Pacific Islander	2 (0.6%)
Other	2 (0.6%)
Education	
Less than high school	2 (0.6%)
High school graduate	129 (37.6%)
Some years of college/university (no degree)	194 (56.6%)
Vocational training	2 (0.6%)
Associates degree	8 (2.3%)
Bachelor's degree	5 (1.5%)
Master's degree	1 (0.3%)

Appendix B

The random forest output contains different information than any other technique discussed in this paper because it performs a type of cross-validation internally through looking at something called Out of Bag error (OOB; sometimes referred to as the out-of-bag estimate). The OOB is an approach to measuring the prediction error of a random forest model or of other decision tree models. OOB error is the mean prediction error of a given sample, using only the trees which did not have that sample in their bootstrapped sample. This sounds potentially confusing, but it simply means that the OOB error is the average prediction error of a given sample of data when that sample of data is treated as a test sample rather than a train sample (i.e., a tree is evaluated on that data since it has yet to see it). OOB error is also used for other machine learning models implementing something called bootstrap aggregation (bagging). Bagging is the official term for only considering a random sample of the data when random forest creates each tree. It is unique in that it is a random sample that allows

for repetition, meaning that the records for a single participant could be represented more than once in the sample. For more on the theory behind bagging, see work by Ghojogh and Crowley (2023). In addition to the OOB error rate, the output provides a confusion matrix, something that is often used to discuss the performance of a classification method. A confusion matrix follows the form below:

Table 2. Confusion Matrix with Signal Detection Theory Terminology

	True 0	**True 1**
Predicted 0	Correct Rejection	Miss
Predicted 1	False Alarm	Hit

It is ideal to have a high number of both hits and correct rejections and a low number of both false alarms and misses. It is possible that one may wish to allow for more false alarms so as to decrease miss rates in some cases (e.g., a doctor would likely rather have a false positive screening for cancer than miss a cancer diagnosis). In other cases, one may want to minimize false alarms (e.g., in the court system, it is ideal to minimize the number of innocent people who are sent to jail). Thus, it is incredibly beneficial to understand each of these statistics when evaluating the performance of a classification model, as they both factor into calculating accuracy. The `randomForest()` output provides a classification error representing the proportion of a given class which has been misclassified (e.g., a true 0 that was classified as 1 or the reverse). For the model demonstrated, there is no classification error for either class since perfect accuracy occurred.