Statistics 101C Final Project

Predictive Analysis of Alcoholic Status

Mandy Lu, Ian Zhang, Shovanne Juang

Lecture 1

Abstract

The goal of this project is to predict alcoholic status using classification models on the given data. Our report provides insight into how we used statistical learning models to classify the response variable of alcoholic status. Included are descriptions of the background context, our data analysis of the variables given, different methods and models fitted, and which final model we chose based on its strengths and limitations. Our final model is based on boosting with MICE imputation and uses all variables in a full model. It has a Kaggle score of 0.73086, which brings us to 25th place.

I. Introduction

Alcoholism has been a very prevalent issue in the United States and continues to affect many Americans today, as an estimated 88,000 people die from alcohol-related causes every year. This number has risen to 95,000 in 2023, which is equivalent to 261 deaths every day. Long-term abuse of alcohol is associated with an increased risk of injuries, chronic liver and heart diseases, cancer, and pregnancy issues as well. The generational impact of alcoholism is also crippling – over half of all American adults have a family history of problematic drinking tendencies or an alcohol addiction. Immediate effects of excessive alcohol use like binge drinking include injuries, alcohol poisoning, car crashes from drunk driving, violence like homicide, suicide, sexual assault, and miscarriage (among other pregnancy or birth defects). Every day, 37 people in the United States die in car crashes with an alcohol-impaired driver, and this number has accumulated to 11,000 deaths annually per year from 2011 to 2021 – alcohol-impaired drivers account for more than 31% of all traffic-related deaths. It is more

important now than ever to detect signs of early alcoholism, as 1 in 4 crashes with teens involve an underage drunk driver. If we can prevent such severe cases of alcohol addiction, we will be able to save thousands more lives every year, and therefore imperative to create an efficient model to accurately and efficiently classify alcoholism status in Americans.

In the Kaggle Project, the alcohol drinking data set has 26 predictors detailing an individual's medical information, including age, blood pressure, and smoking status. The training data set has 70,000 observations, while the testing data set has 30,000 observations. Both numerical and categorical predictors are present in the data set. The numerical predictors show specific values regarding an individual's medical background. Some numerical predictors include an individual's cholesterol levels, total hemoglobin, and various blood pressure levels. The categorical predictors explain the category each individual belongs to, ie. Smoking Status category– Still Smoking, Used to Smoke, and Never Smoked. Our goal for the Kaggle project is to introduce a classification model that predicts an individual's alcoholic status in the testing data with significant predictors.

II. Data analysis

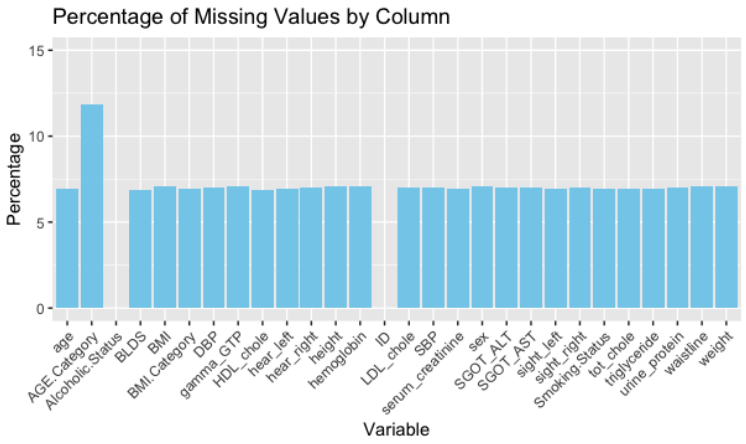
A. Overview of Data Analysis Methodology

After analyzing our datasets, we noticed our training and testing datasets had missing values. To improve our model's performance, our group decided to conduct feature selection to find the most significant predictors, as well as imputation techniques, like crude imputation with mean, Hmisc, RandomForest, and MICE.

B. Missing Values

Within the training and testing datasets, there are a total of 130,776 and 56,035 missing

values respectively.



The missing values were spread throughout the categorical and numerical variables, having the same proportion of NA values (~7%) besides AGE.Category, which had significantly more NA's than the rest of the variables (~12%). Cleaning the NA's are pertinent when using machine learning algorithms as well as achieving a lower misclassification rate. Since the number of NA's in each variable is roughly the same, it doesn't make sense to remove any variable, and thus we need to use imputation in order to successfully deal with the NA's within the predictors.

C. Cleaning data

In order to clean the data, we conducted several methods to test which method gave us the best results. The first method we tried was imputation with mean. Initially, in class, we learned about the na.omit() function, where we directly removed the NA values. However, a better way to improve the accuracy is imputation with the mean, median, or mode of the dataset. This rough approximation method can work depending on if the variation is low or if the variable has low leverage over the response. However, it is a crude method and we soon learned other imputation methods that were far more effective. In addition to this, we also tried Hmisc, randomForest (na.roughfix), and MICE.

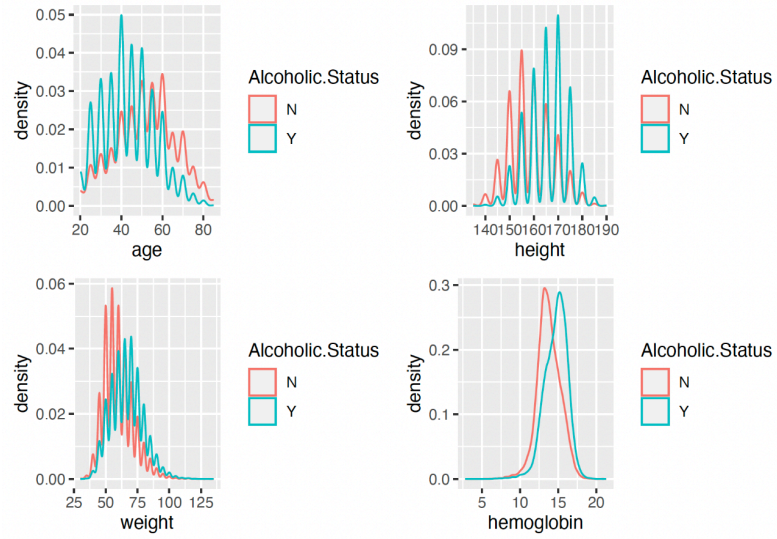
Our next attempt was the ‘na.roughfix’ function in the ‘randomForest’ package. After gaining a better understanding from outside resources, we decided to use it to fill in missing values. This method replaces missing values with column medians for numerical predictors and column modes for categorical predictors, providing a straightforward yet rough imputation technique. However, this method improved our accuracy successfully.

In the “Merging Data and Imputation of Missing Values” lesson from class, we learned that Hmisc and MICE could be very powerful tools in imputing NA values in a dataset to improve model accuracy. We tried Hmisc, which is useful for data analysis of high-level graphics. We focused on its two most powerful functions for imputing missing values, impute() and aregImpute() (note that we did not use transcan(), because aregImpute() is better to use, as stated in the slides). We attempted to use aregImpute, which focuses on mean imputation using additive regression, bootstrapping, and predictive mean matching. Using aregImpute() was more difficult and a more time-consuming process than many of the other methods, and although we used it to impute missing variables in the dataset, we did not end up using the imputed data in any of our subsequent models. MICE was a much better bet in this case to use for our models, as we will explain.

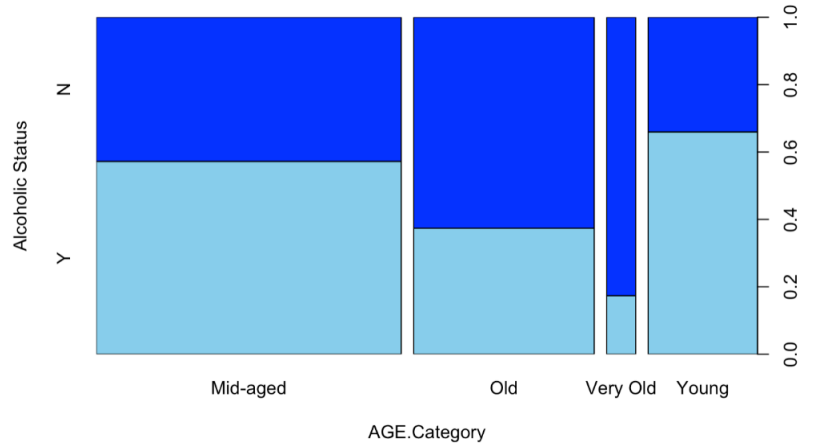
Multivariate Imputation by Chained Equations (MICE) provides the most flexibility and we also tried it to impute missing values. The method iterates through each variable and imputes missing values based on the observed values. The process is repeated to create multiple datasets which are then combined to obtain estimates that account for the uncertainty. This approach became the best in terms of data cleaning, stemming from its flexibility, ability to handle various variables, and capacity to handle NA values in complex datasets.

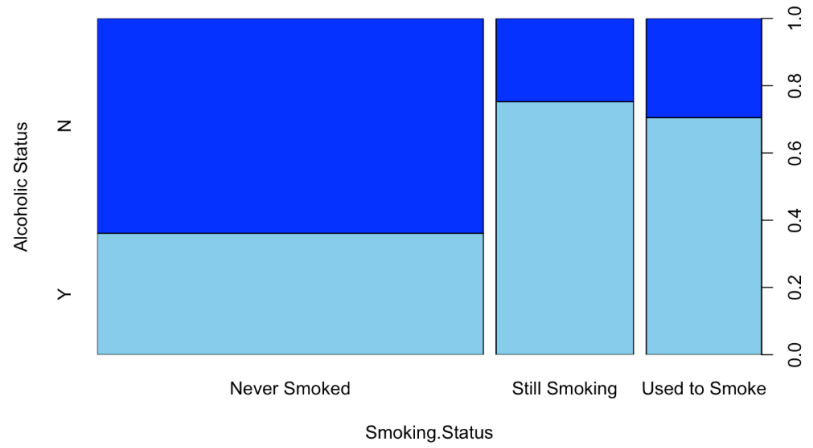
D. Feature Selection (categorical/numeric selection)

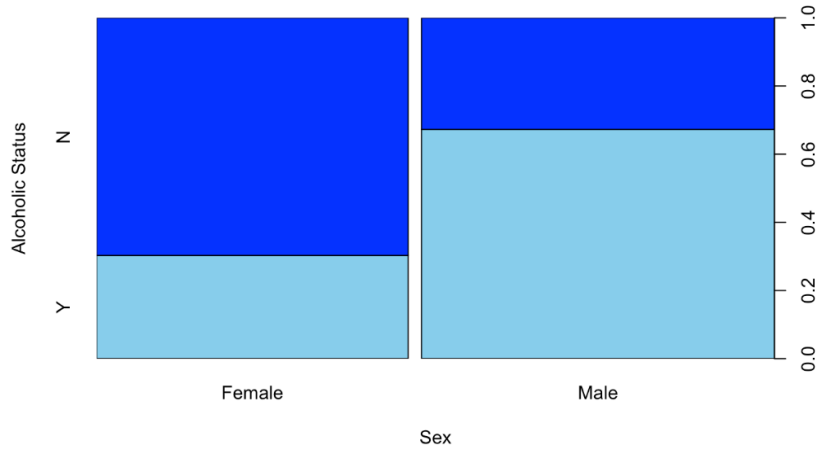
Oftentimes it is more optimal to use a subset of predictors versus the full set to create a model – this is due to possible collinearity between predictors, or the relative importance of the variable in the context of the problem. To investigate, we separated the numeric and categorical variables to determine which to cut out, if needed. For the numeric predictors, we constructed density plots with the training data, and used grid.arrange() to compare the density plots of each predictor against the response variable, Alcoholic.Status. Using the "who's your daddy" plot-selection process, we found that age, height, weight, and hemoglobin are the best numeric predictors. See the density plots below:



For categorical variables, we used stacked bar charts to see how each predictor influences Alcoholic Status. From the charts below, you can see that AGE.Category, Smoking.Status, and Sex were the best categorical predictors.







So, in addition to a full dataset, for our model creation, we also considered a subset of these seven predictors: age, height, weight, hemoglobin, age category, smoking status, and sex.

III. Methods and models

A. Logistic Regression

The first method we tried was logistic regression. Logistic regression is a supervised learning method as well as a simple model that allows us to predict a categorical dependent variable based on the predictors. Logistic regression is very easy to train, as well as straightforward to interpret, and performs very well if the data is linear. However, since the original dataset has 26 predictors, we decided to take a subset of these predictors from the density plots above and apply logistic regression to those 7 predictors. We used imputation with means with this model. The final logistic regression model with 7 predictors yielded a misclassification rate of 29.742% and a Kaggle score of 0.70713.

B. LDA

We then tried using linear discriminant analysis, which is closely related to analysis of variance (ANOVA) and regression analysis, methods that express one response variable as a linear combination of a set of predictors. Discriminant analysis is different from ANOVA in that it uses continuous independent variables and a categorical dependent variable, similar to logistic regression (but logistic regression deals with only two-class classification problems). The main focus of LDA is that it looks for linear combinations of variables that best explain the data. It is used as a supervised linear transformation technique to reduce dimensionality with datasets of more than two classes. In this case, we decided to use LDA and input our subsetted predictors (age, height, weight, hemoglobin, age category, smoking status, and sex) into the model to predict Alcoholic Status from the data imputed with means. This LDA model had a slightly lower misclassification rate (28.72%) and a higher Kaggle Score (0.7188) than the logistic regression model, but we wanted to see if more advanced models like random forest or boosting

could help us improve our results.

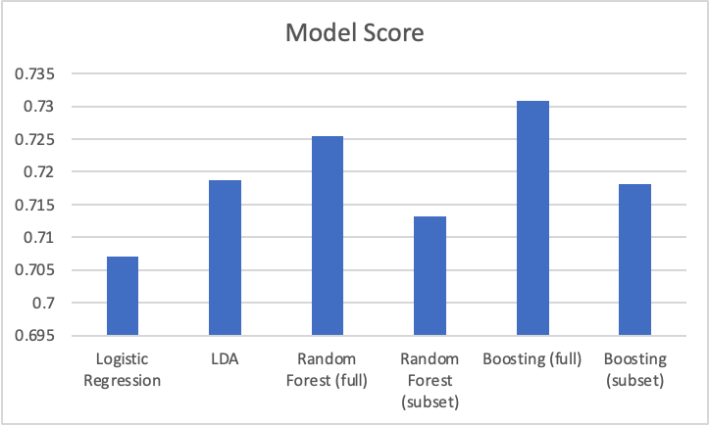
C. Random Forest

We went on to try RandomForest twice by using different data-cleaning techniques to enhance predictive accuracy. The random forest model is a machine learning algorithm that leverages multiple decision trees that consider a random sample of predictors, which ultimately de-correlates the trees and strengthens the prediction. Understanding both the strengths and weaknesses, we decided that random forest was a strong algorithm. Initially, we used the 'na.roughfix' function with all predictor variables to handle missing values. This imputation method combined with random forest achieved a lower misclassification rate and a Kaggle score of 0.72556, surpassing the performance of both logistic regression and linear discriminant analysis models. Subsequently, we attempted another approach using random forest on a subset of predictors. From the density plots, we identified seven significant predictors: age, height, weight, hemoglobin, sex, Age.Category, Smoking.Status. Despite the removal of variables, the misclassification rate for the subset model was higher than the full model and had a Kaggle score of 0.71326, suggesting that a more extensive set of predictors is better.

D. Boosting

Lastly, we attempted the boosting method twice with the full model and subsetted models. Boosting is similar to bagging and random forest in the sense that multiple decision trees are fit. However, boosting fits trees sequentially instead of completely randomly, which allows it to learn slowly as well as build upon the trees already created, increasing the accuracy of the model. Boosting also does not involve bootstrap sampling; instead, each tree is fit on a modified version of the sample. Even though boosting has the limitation that it may be more computationally expensive to run, our data set was not large enough for that drawback to

substantially affect our model, so we were able to apply boosting to our data. We used the MICE imputation method and trained a boosting model with all predictors to achieve a lower misclassification rate as well as a Kaggle score of 0.73086, meaning that this model was the best out of all the models we fit. We then tried trimming down the predictors according to the density plots from before to see if subsetting the predictors would have a positive effect on the model. We fit another model with the 7 predictor-subset from before and found that removing predictors increased our misclassification rate and decreased the Kaggle score to 0.7181, which suggests that boosting with the full predictor set is the best model.



IV. Discussion and limitations

We got 25th place in the leaderboard and thus acknowledge that there are some limitations in our model that we could examine to further improve our accuracy for the future. The different methods that we tried have their limitations. For example, logistic regression doesn't work very well if the predictors are not linear. If the relationship between the response variable and the predictors is not linear, then another model would do better. Similarly, Linear Discriminant Analysis (LDA) is also limited, as its name suggests, to linear relationships since it is a transformative technique based on using linear combinations. For data that more closely

follows a quadratic or other polynomial structure, then QDA or some other method would be better. The drawbacks of random forest consist of being computationally expensive as we have experienced the algorithm running extremely slowly. Additionally, the random forest model tends to be biased toward attributes with more levels and is very difficult to interpret. With boosting, the weaknesses consist of being computationally expensive, potential overfitting issues, and similarly to random forest being difficult to interpret.

After attempting various machine learning algorithms, our best models were random forest using all predictors, boosting using all predictors, and our subsetted boosting model. The subsetted boosting model using seven key predictors had a misclassification rate of 28.17% and had third in our Kaggle score. The random forest model had a misclassification rate of 27.47%, earning second in our Kaggle Score. The boosting model using the full model gave us the lowest misclassification rate of 26.43% with the highest Kaggle score. Aside from the scores, we had to take into account the simplicity of each model– random forest and boosting. Comparing the methods, random forest is better in terms of simplicity whereas boosting corrects errors made by previous models, making it more complex. In terms of use cases, random forest is best for quick implementation and interpretability. For boosting, it is most effective when accuracy is a strong component in the model. The decision between choosing the random forest model or the boosting model was carefully analyzed from their pros and cons.

V. Conclusion and recommendation

In our final model, we chose boosting with all predictors after using the MICE imputation to clean values. The misclassification rate is 28.17% which was the lowest compared to all models we attempted. Additionally, the model achieved a Kaggle score of 0.73086, placing our model at a rank of 25th.

MICE performs variable-by-variable imputation, offering a more nuanced approach compared to methods restricted to the entire dataset. In contrast, approaches like MVN lag as it uses a joint modeling approach based on multivariate normal distribution that requires transformations to approximate normality – an additional step that MICE does not require. Shifting focus from data imputation to our model, boosting refines subsequent trees from previously grown trees. The method "learns slowly", which allows for a complex model with high accuracy rates. However, there are some limitations in our model. In our data cleaning that used the MICE imputation, there may be some biased estimates if the data is not Missing at Random. For our variable selection, our group could have attempted different combinations of predictors instead of solely focusing on density plot results, which may have potentially given us higher accuracy with the subset model rather than the full model. Although boosting was the best method in terms of accuracy, the limitation of boosting is the fact that it is difficult to interpret due to its complexity.

After finding the most significant predictors, we found that the most important predictors are sex, Smoking.Status, and age in the model. Therefore, if you knew someone's sex, age, and smoking status, it is likely to accurately predict if they were alcoholic or not.

VI. Acknowledgments

Thank you Professor Almohalwas for a great quarter! We appreciate your passion for statistics, making the material not only enjoyable and understandable but also applicable to our future careers. We are grateful for all of the knowledge and skills we have gained this quarter!

References

Almohalwas, Akram Mousa. “Prediction of Alcoholic Status”. Dec, 1, 2023.

Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011, March). *Multiple imputation by chained equations: What is it and how does it work?*. International journal of methods in psychiatric research. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning: With applications in R*, n.d.

Imputing missing data with R; Mice Package. DataScience+. (n.d.). https://datascienceplus.com/imputing-missing-data-with-r-mice-package/

Na.roughfix: Rough imputation of missing values. RDocumentation. (n.d.). https://www.rdocumentation.org/packages/randomForest/versions/4.7-1.1/topics/na.roughfix

Tri, M. (2022, August 12). *Dealing with missing data: Imputation*. RPubs. https://rpubs.com/minhtri/968586