Kaggle Competition

Predicting Alcohol Status

By: Shovanne Juang, Mandy Lu, Ian Zhang
(Lecture 1)

Table of *Contents*

01 Introductions

Alcohol Drinking Data Set Overview

America’s Alcoholic Problem

Alcoholism remains a significant issue in the United States, as alcohol-related deaths continuously rise.

With the high death rate that alcohol imposes, efficiently detecting early signs of alcoholism and implement an efficient model to classify individuals' alcoholism status is crucial to address this health concern and safe lives.

Alcohol Data Set

Observations
70,000

Each observation represents an individual's vitals and medical information.

Variables
26

The variables contained detailed information recorded about each individual (eg. Alcoholic Status, BMI, Smoking Status, etc).

02

Methodology

Data Cleaning and Modeling

The Process

Clean Data

Remove NA's and predictors that have high collinearity.

Model Data

Conduct different models to create predictions on Alcoholic Status.

Analyze Models

Comparing the lowest misclassification rates.

Compare Models

Pick the best overall model from the data.

Data Cleaning: Imputing NA's

Imputation with Mean

At first, we tried using a function to replace missing or NA values with the mean of the dataset, as versus just simply removing the NA values. This is considered a crude method, but it was an easy fix to start off with. We later discovered that other imputation methods, like MICE, were far more effective, as we will explain later in the slides.

Data Cleaning: Imputing NA's

Different imputation methods tried:

Hmisc

We used Hmisc to impute missing values by additive regression, bootstrapping, and predictive mean matching using *aregImpute()* function.

Random Forest

Using random forest, we imputed missing values by medium/mode using the *na.roughfix()* function.

MICE*

We used MICE, or Multivariate Imputation by Chained Equation, to impute missing values using multiple complete copies of different data frames with different imputations of missing data using the *mice()* and *complete()* functions.

Data Modeling

Logistic Regression

Strengths:
Simple
Interpretability

Weaknesses:
Overfitting
Linearity Assumption

RandomForest

Strengths:
Accurate
Large Data sets
Unbiased Estimates

Weaknesses:
Runs Slow
Biased (attributes with more levels)

LDA

Strengths:
Stable (Normal Distributions)
Feature Selection (Handles Collinearity)

Weaknesses:
Less Flexibility
Outlier Sensitive

Boosting*

Strengths:
Most Accurate (*Learns Slowly*)

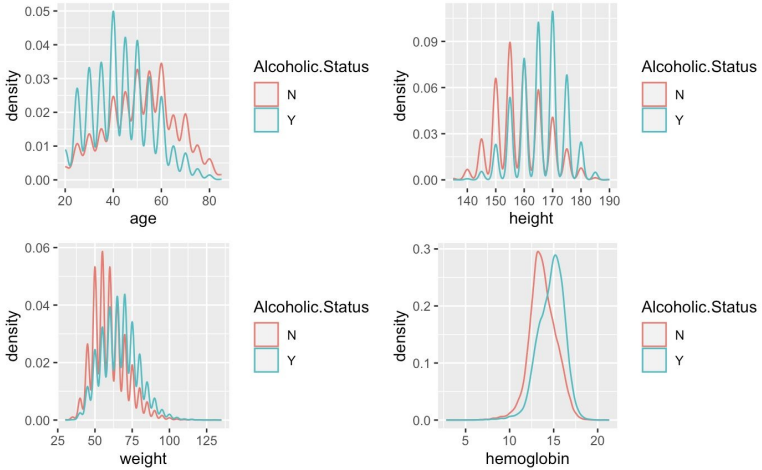
Weaknesses:
Computationally Expensive
Hard to Interpret
May Overfit

Feature Selection

Density Plots

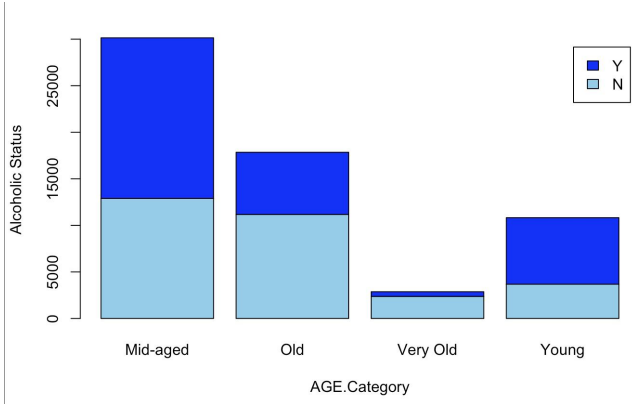
We constructed density plots with the training data, and compared their density plots with grid.arrange() functions for the numeric predictors and their stacked bar plots for the categorical predictors . We determined that the following 7 predictors were the best: age, height, weight, hemoglobin, sex, AGE.Category, Smoking.Status. We then used this trimmed set of predictors in certain models later on to predict Alcoholic Status.

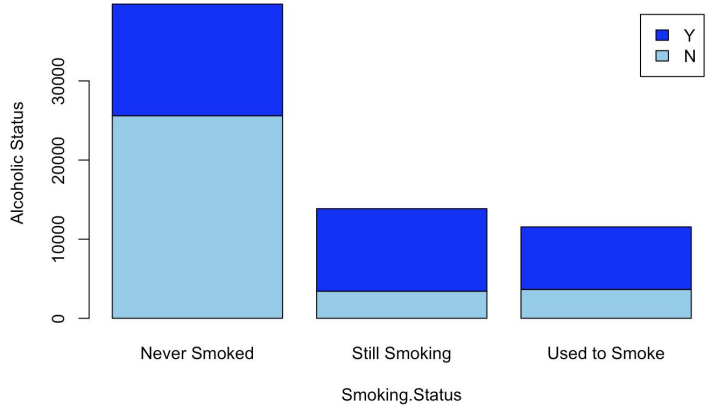
For **numerical** predictors age, height, weight, hemoglobin:

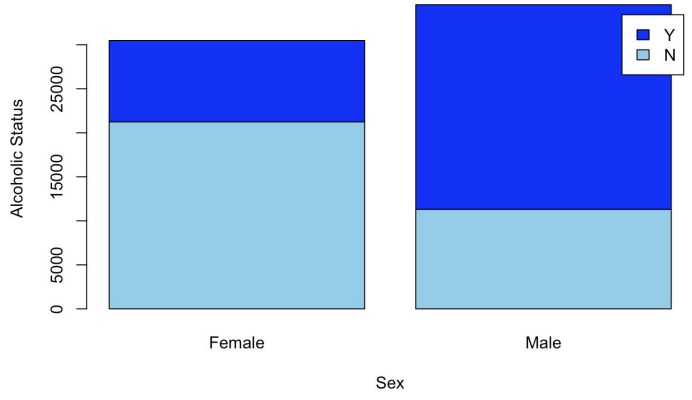


Feature Selection

These are the stacked bar plots to compare the **categorical** predictors sex, AGE.Category, Smoking.Status:







Logistic Regression Model (subsetted)

Method

We subsetted by removing predictors based on density plots.

We constructed a logistic regression model using data imputed with means to replace the NA values with the following predictors:

age, height, weight, hemoglobin, sex, AGE.Category, Smoking.Status

We achieved a misclassification rate of 29.7% and a Kaggle score of 0.70713. Let's see if we can try more complex models we learned in class!

Confusion Matrix:

	N	Y
N	24644	10351
Y	10469	24536

Misclassification Rate:

29.742%

Kaggle Score:

0.70713

LDA Model (subsetted)

Method

We constructed a Linear Discriminant Analysis Model using a subset of predictors (same as the predictors used in the logistic regression model: age, height, weight, hemoglobin, sex, AGE.Category, Smoking.Status) and the data imputed with means to replace the NA values.

The LDA model has a slightly lower misclassification rate and higher Kaggle score than the logistic regression model. This misclassification rate is not bad, but let's see if random forest and/or boosting can give us a better result.

Confusion Matrix:

	N	Y
N	25152	10143
Y	9961	24744

Misclassification Rate:

28.72 %

Kaggle Score:

0.7188

Random Forest Model (full)

Method

We constructed a random forest model by cleaning the data set with na.roughfix with all predictor variables on the training data:

Then, we predicted Alcoholic Status on the testing data using the model to get a lower misclassification rate and a Kaggle score of 0.72556.

This result is better than both the logistic regression and the LDA models, implying that the data is likely to be non-linear and thus, a complex model would fit better.

Confusion Matrix:

	N	Y
N	24955	10158
Y	9070	25817

Misclassification Rate:

27.47%

Kaggle Score:

0.72556

Random Forest Model (subsetted)

To see whether removing variables would decrease the misclassification rate, we trained another random forest model on a subset of the predictors.

We subsetted by removing predictors based on density plots.

Based on the density plots, we chose 7 best predictors (age, height, weight, hemoglobin, sex, AGE.Category, Smoking.Status) in a subset to build a random forest model.

The misclassification rate is higher than that of the full model, which suggests that more predictors are needed to accurately train the models.

Confusion Matrix:

	N	Y
N	23455	11658
Y	8931	25956

Misclassification Rate:

29.41%

Kaggle Score:

0.71326

Boosting Model (full)

Method

Since random forest and boosting theoretically are the best methods, we tried boosting to see if it did as well/better than random forest.

We constructed a boosting model using all predictors and using the MICE imputation method.

In terms of accuracy, it has given us the lowest misclassification rate and highest Kaggle Score.

Confusion Matrix:

	N	Y
N	25911	9298
Y	9202	25589

Misclassification Rate:

26.43%

Kaggle Score:

0.73086

Boosting Model (subsetted)

Method

After training a boosting model with all predictors, we constructed a boosting model using the 7 best predictors based on the density plots and the MICE imputation method to determine whether the removal of predictors would affect the boosting model:

age, height, weight, hemoglobin, sex, AGE.Category, Smoking.Status

In terms of accuracy, it appears that the full boosting model performed better than the subsetted model with a higher misclassification rate and lower Kaggle score.

Confusion Matrix:

	N	Y
N	28003	12607
Y	7110	22280

Misclassification Rate:

28.17%

Kaggle Score:

0.7181

Comparing Models

Random Forest (Full Model)

27.47%

MSE

2nd

Kaggle Score

Boosting (Full Model)

26.43%

MSE

1st

Kaggle Score

Boosting (Subsetted Model)

28.17%

MSE

3rd

Kaggle Score

03 Results

Final Model Analysis

Final Model

Boosting*

The final model we chose was boosting with all predictors after using MICE to impute missing or NA values. The misclassification rate is the lowest at 28.17%, which is lower than the rates of the other models. The Kaggle score was the highest, at 0.73086.

According to this model, the most important predictors are: sex, Smoking.Status, and age.

Thus, if you knew someone's sex, age, and their smoking status, you could be fairly confident in being able to accurately predict if they were alcoholics or not.

Analysis of Model

Boosting Model with MICE Imputation

MICE imputes data on variable by variable basis and is capable of handling different types of variables. It is also more versatile in the sense that it can manage imputation of variables defined on a subset of data (as versus being limited to only dealing with the full dataset). Other methods like MVN lags behind because it uses a joint modeling approach based on multivariate normal distribution – MICE does not need to go through any sort of transformation to bring data close to normality. Boosting is an ensemble method based on decision trees. It begins with training a decision tree, but instead of every tree it creates be random, each subsequent tree is trained using information from previously grown trees. This allows the boosting method to learn slower, which increases accuracy and allows for a more complex model.

04 Limitations

Modeling Drawbacks

Final Model *Limitations*

Data Cleaning

When data are not Missing At Random, using MICE may result in biased estimates. Imputed values may also be very variable if there are not enough observations in the dataset.

Variable Selection

Chose full model, but potentially different combination of predictors would have been a better model.

Interpretability

Difficult to interpret due to the complexity of boosting

05
Conclusion

Summary of Model Findings

Conclusions

Our final model achieved a fairly good prediction, earning 25th rank in the Kaggle Competition. However, our model's limitations give us poor interpretations whereas it does not give a clear understanding of how each variable affects the outcome.

Additional research would be beneficial for improving the accuracy of our model. However, we believe our model provided great insight in predicting an individual's Alcoholic Status.

References

na.roughfix function - RDocumentation
https://www.rdocumentation.org/packages/randomForest/versions/4.7-1.1/topics/na.roughfix

Imputing Missing Data with R; MICE package
https://datascienceplus.com/imputing-missing-data-with-r-mice-package/

Prediction of Alcoholic Status
Kaggle Competition Project Overview & Background File

Dealing with Missing Data: Imputation
https://rpubs.com/minhtri/968586

Multiple imputation by chained equations: what is it and how does it work?
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/

Thanks!

Please let us know if you have any questions!

Thank you Professor Almohalwas for a great Fall quarter!