Predicting Alcoholic Status Using Various ML Classification Methods

Ahyoung Ju, Ashton Chung, Emily Pham, Johan Chua, Nathan Lim

December 17, 2023

1 Abstract

This paper investigates different statistical algorithms and classification models to predict the alcoholic status of individuals based on their vitals and other information for a Kaggle competition. In our paper, we investigate the dataset from the National Health Insurance Service in Korea. Specifically, we performed exploratory data analysis to understand the data's features, built predictive models using a host of machine learning techniques to classify individuals' alcoholic status, and evaluated our models using confusion matrices and accuracy rates from Kaggle. We ultimately found that our best performing model was the Gradient Boosting Machines (GBM) model that used Hmisc-imputed data, which yielded a top score of 0.73136 in the Kaggle competition.

2 Introduction

Alcoholism is a very serious and pervasive public health concern facing society today. It has a wide range of negative consequences for individuals afflicted with it, their families and their communities, including tolls to physical and mental health (liver/heart disease and cancer), relationships, and financial burdens. In the US, more than half of all American adults have a family history of alcohol addiction and it has become the third-leading cause of preventable death in the nation. Thus, alcoholism is an urgent issue that needs to be solved, and it's one that can be solved with data—specifically machine learning classification models.

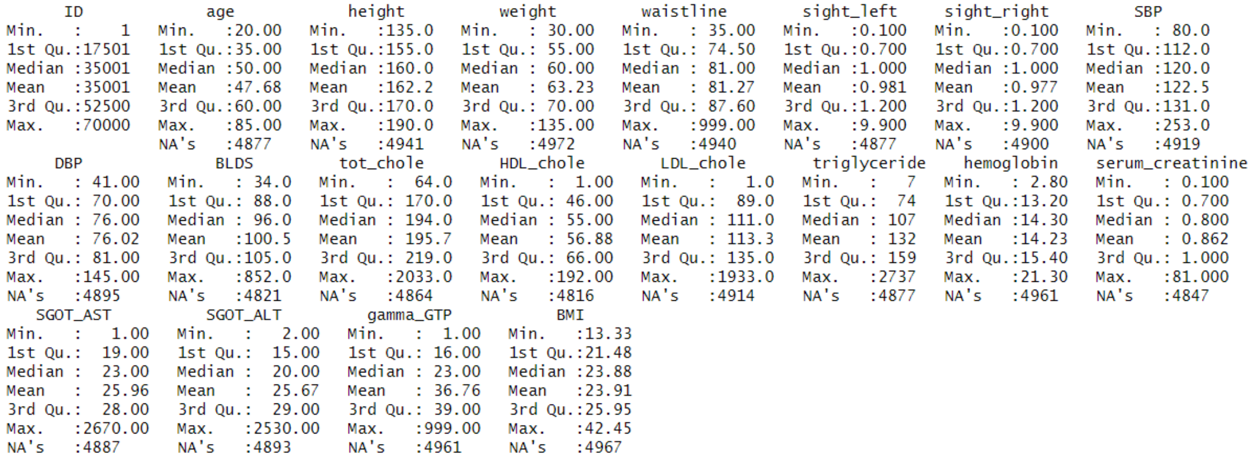
Our team was tasked with developing a reliable and accurate classification model for predicting alcoholic status using data collected from the National Health Insurance Service in Korea. Our intended response variable, Alcoholic Status, is a categorical variable that takes on either "Y" for Alcoholic or "N" for Not Alcoholic. We were provided a training dataset containing 70000 observations and 27 variables, and a testing dataset with 30000 observations and 26 variables (the testing dataset did not contain the response variable, "Alcoholic.Status"). We will first perform exploratory data analysis on our dataset to understand our variables. We then will impute any missing values in order to maximize the predictive power of our data before building various prediction models that will classify alcoholic status. Classification methods we will test and explore include K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines (SVM), and XGBoost among others. Each of these techniques offers unique strengths in analyzing data and predicting outcomes. Lastly, we will evaluate our different methods and determine our best overall model using confusion matrices, testing accuracies, and Kaggle scores.

Thus, this endeavor holds significant importance, as accurately identifying individuals with alcohol dependence can lead to early intervention and better health outcomes.

3 Data Analysis

3.1 Exploratory Data Analysis

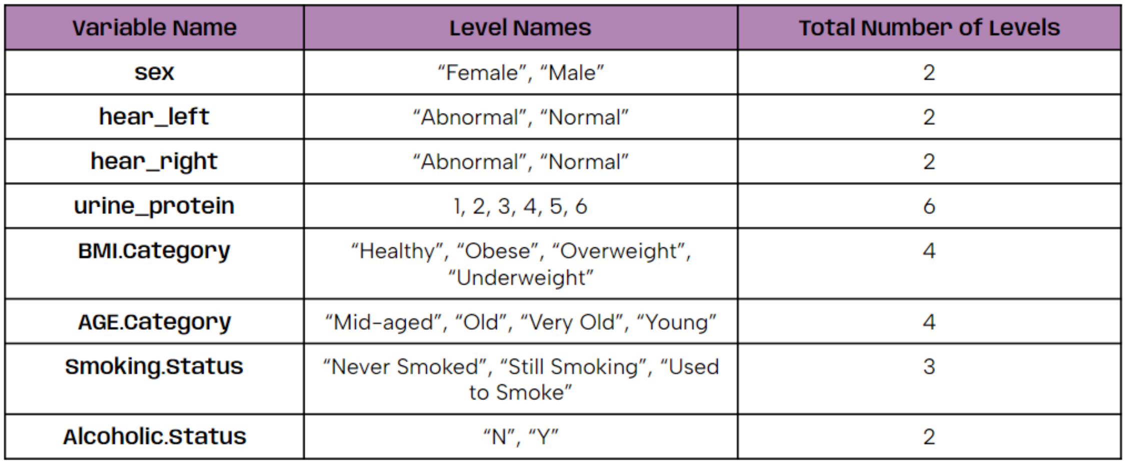
We initially performed exploratory data analysis on the given datasets to uncover any trends that may exist in the data as well as assist us in developing a strategy as to which models would be best to evaluate. We determined that there were 20 quantitative variables and 8 categorical variables.



```
       ID             age            height          weight         waistline        sight_left      sight_right          SBP       
 Min.   :    1   Min.   :20.00   Min.   :135.0   Min.   : 30.00   Min.   : 35.00   Min.   :0.100   Min.   :0.100   Min.   : 80.0  
 1st Qu.:17501   1st Qu.:35.00   1st Qu.:155.0   1st Qu.: 55.00   1st Qu.: 74.50   1st Qu.:0.700   1st Qu.:0.700   1st Qu.:112.0  
 Median :35001   Median :50.00   Median :160.0   Median : 60.00   Median : 81.00   Median :1.000   Median :1.000   Median :120.0  
 Mean   :35001   Mean   :47.68   Mean   :162.2   Mean   : 63.23   Mean   : 81.27   Mean   :0.981   Mean   :0.977   Mean   :122.5  
 3rd Qu.:52500   3rd Qu.:60.00   3rd Qu.:170.0   3rd Qu.: 70.00   3rd Qu.: 87.60   3rd Qu.:1.200   3rd Qu.:1.200   3rd Qu.:131.0  
 Max.   :70000   Max.   :85.00   Max.   :190.0   Max.   :135.00   Max.   :999.00   Max.   :9.900   Max.   :9.900   Max.   :253.0  
                 NA's   :4877    NA's   :4941    NA's   :4972     NA's   :4940     NA's   :4877    NA's   :4900    NA's   :4919   
      DBP             BLDS           tot_chole         HDL_chole         LDL_chole       triglyceride     hemoglobin    serum_creatinine
 Min.   : 41.00   Min.   : 34.0   Min.   :  64.0   Min.   :  1.00   Min.   :   1.0   Min.   :   7   Min.   : 2.80   Min.   : 0.100  
 1st Qu.: 70.00   1st Qu.: 88.0   1st Qu.: 170.0   1st Qu.: 46.00   1st Qu.:  89.0   1st Qu.:  74   1st Qu.:13.20   1st Qu.: 0.700  
 Median : 76.00   Median : 96.0   Median : 194.0   Median : 55.00   Median : 111.0   Median : 107   Median :14.30   Median : 0.800  
 Mean   : 76.02   Mean   :100.5   Mean   : 195.7   Mean   : 56.88   Mean   : 113.3   Mean   : 132   Mean   :14.23   Mean   : 0.862  
 3rd Qu.: 81.00   3rd Qu.:105.0   3rd Qu.: 219.0   3rd Qu.: 66.00   3rd Qu.: 135.0   3rd Qu.: 159   3rd Qu.:15.40   3rd Qu.: 1.000  
 Max.   :145.00   Max.   :852.0   Max.   :2033.0   Max.   :192.00   Max.   :1933.0   Max.   :2737   Max.   :21.30   Max.   :81.000  
 NA's   :4895     NA's   :4821    NA's   :4864     NA's   :4816     NA's   :4914     NA's   :4877   NA's   :4961    NA's   :4847    
    SGOT_AST           SGOT_ALT          gamma_GTP           BMI       
 Min.   :   1.00   Min.   :   2.00   Min.   :  1.00   Min.   :13.33  
 1st Qu.:  19.00   1st Qu.:  15.00   1st Qu.: 16.00   1st Qu.:21.48  
 Median :  23.00   Median :  20.00   Median : 23.00   Median :23.88  
 Mean   :  25.96   Mean   :  25.67   Mean   : 36.76   Mean   :23.91  
 3rd Qu.:  28.00   3rd Qu.:  29.00   3rd Qu.: 39.00   3rd Qu.:25.95  
 Max.   :2670.00   Max.   :2530.00   Max.   :999.00   Max.   :42.45  
 NA's   :4887      NA's   :4893      NA's   :4961     NA's   :4967   
```

Figure 1: Summary Statistics for Numerical Variables

Above are the summary statistics for all our quantitative variables. A preliminary look at the minimum and maximum values for each variable reveals evidence of outliers since many variables have a wide range. For example, the maximum values for waistline, LDL_chole, and triglyceride appeared especially high relative to the mean values.



Variable Name	Level Names	Total Number of Levels
sex	"Female", "Male"	2
hear_left	"Abnormal", "Normal"	2
hear_right	"Abnormal", "Normal"	2
urine_protein	1, 2, 3, 4, 5, 6	6
BMI.Category	"Healthy", "Obese", "Overweight", "Underweight"	4
AGE.Category	"Mid-aged", "Old", "Very Old", "Young"	4
Smoking.Status	"Never Smoked", "Still Smoking", "Used to Smoke"	3
Alcoholic.Status	"N", "Y"	2

Figure 2: Levels for Categorical Variables

Above is a summary table containing the levels for all our categorical variables. The testing dataset did not contain Alcoholic.Status. We observed that BMI.Category and AGE.Category variables are correlated with their age and BMI numerical counterparts.

3.2 Imputing Missing Values

Our preliminary examination of the datasets revealed that every column contained missing values except the variable “ID” and the response variable “Alcoholic.Status”. Overall, roughly 8% of values were missing across all variables, as shown in the bar charts below.

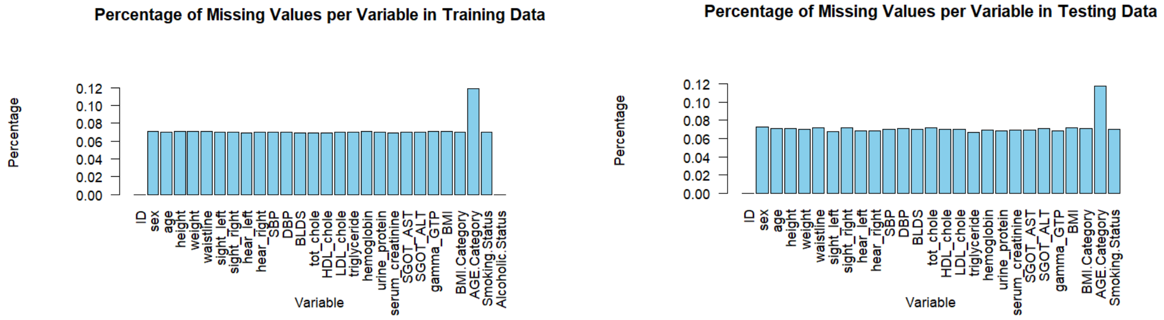


Figure 3: Percentage of Values Missing for each Variable

This posed an issue to our team, since not all statistical models and R functions are capable of handling missing values. As a result, our team faced two choices: data deletion or data imputation. With our data, deleting observations with missing values was not practical because 86% of observations in both datasets contained at least one missing value. Deleting all these observations would have deprived us of most of our training data. Additionally, deleting variables that had missing values was also unreasonable because every useful variable contained missing values. Thus, we decided that imputing was the best way forward, and we did so using a host of imputation techniques. This allowed us to compare imputation methods and determine the best imputed training dataset to use.

3.2.1 Using Logic, Medians and Modes

Our team first proceeded by determining which data could be imputed logically. For observations that contained values for at least two of height, weight, or BMI, we were able to derive the missing third value by the BMI formula. Moreover, we could obtain BMI.Category and AGE.Category for observations that had values for BMI and age respectively. Thereafter, we performed a median impute for all other missing quantitative data and mode impute for all other missing categorical data. Although very simple, a major advantage of this method is its computation efficiency on large, high-dimensional data like the one being used in this project.

3.2.3 Multivariate Imputation by Chained Equations (MICE Library)

However, simple techniques like the one above can only be so effective; thus, we also tried more advanced imputation techniques. One of which was Multiple Imputation by Chained Equations—a robust procedure that imputes missing values using a series of predictive models. For each variable, the algorithm uses other variables in the dataset to predict its missing values, and continues to do so until a convergence is achieved. We used the mice() function from the "mice" library to conduct this imputation. Specifically, we used the "pmm" method to predict numerical predictors and the "polyreg" method for categorical variables.

3.2.4 Multiple Imputation using Additive Regression, Bootstrapping, and Predictive Mean Matching (Hmisc Library)

Another method we employed to impute missing values was Multiple Imputation using Additive Regression, Bootstrapping, and Predictive Mean Matching. This additive method uses bootstrapping to approximate predicted values from a Bayesian distribution as well as predictive mean matching to predict missing values. In order to carry out this imputation method, we used the Hmisc library's aregImpute() function. While this method proved to be highly effective, it was also extremely time and computationally intensive. One major challenge with this technique was the fact that it returned numerical values—that corresponded with factor levels—as predictions for our categorical data. This meant that we had to recode all our imputed categorical data manually, which took quite some time.

3.3 Variable Selection

Given that we had many overlapping variables by the dataset's design (eg. both numerical and categorical versions for Age and BMI), it was imperative that we chose an optimal subset of variables from our training data to use as predictors. Not doing so, and simply using all the variables provided in our data to predict Alcohol Status, would lead to issues such as multicollinearity that can severely hinder our different models' performances. In order to determine the optimal subset, we used two methods:

1) The varImp() function from the Caret library, which calculates variable importance for regression and classification models.
2) Density Plots, looking for variables with dissimilar density distributions

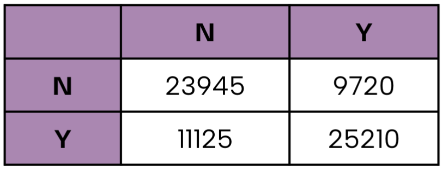
After applying both methods to our various models, we determined that our best performing subset included: sex, height, waistline, SBP, BLDS, tot_chole, HDL_chole, LDL_chole, triglyceride, hemoglobin, SGOT_AST, SGOT_ALT, gamma_GTP, BMI.Category, AGE.Category, and Smoking.Status.

4 Methods and Models

When building our models, we split the training data set into 80% new training data and 20% new testing data using random sampling. We trained each model with the new training data and made predictions on the new testing data in order to estimate the accuracy rate on the actual testing data. Logically, the accuracy rate of the testing data we created and that of the actual testing data should be similar, and this assumption did indeed hold when compared to our Kaggle scores.

4.1 K-Nearest Neighbors (KNN)

KNN finds K-nearest neighbors of training points to make predictions based on how many K-neighbors we set. Before utilizing this method, we first scaled the necessary non-categorical variables in our model to prevent predictors with large magnitudes from dominating the distance calculations. Using the library “class” and its knn() function alongside a multitude of K-values, I found that K = 100 provided the best Kaggle accuracy of 0.71083 despite the high number potentially leading to over-generalizing.

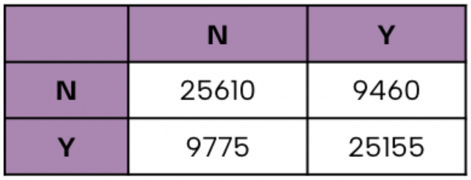


	N	Y
N	23945	9720
Y	11125	25210

Figure 4: Confusion Matrix for KNN

4.2 Logistic Regression (GLM)

Logistic Regression utilizes a binary outcome to predict a model. Normally, logistic regression is efficient when the number of predictors is outnumbered by the number of observations, which is easily true for our data set. We used caret library’s glm() function to achieve an accuracy of 0.7211 when submitted to Kaggle.



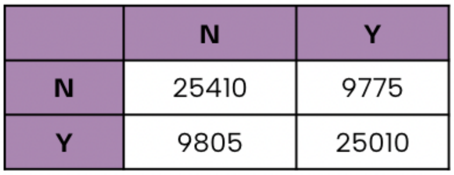
	N	Y
N	25610	9460
Y	9775	25155

Figure 5: Confusion Matrix for Logistic Regression

4.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a machine learning algorithm that finds a hyperplane that best separates data points of different classes. It works both in scenarios in which the hyperplane is linear and non-linear, though we chose to use a linear kernel for the sake of computation efficiency. The algorithm also takes a gamma parameter, which determines the radius of influence of a single data point. We tested gamma values from 0 to 1 using a step size of 0.1 and determined a gamma of 0.8 yielded the highest accuracy rate. We were careful to not test greater values for gamma because although increasing gamma does increase accuracy, it also may lead to overfitting.

We used the svm() function within the e1071 library in R to construct our SVM model. The highest accuracy rate we achieved using the new testing data we created was 0.7202857 and the highest accuracy rate we achieved on the actual testing data from Kaggle was 0.71813. Below is the confusion matrix for the predictions on the new testing data we created.

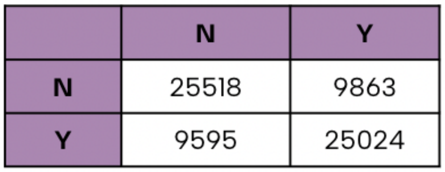


	N	Y
N	25410	9775
Y	9805	25010

Figure 6: Confusion Matrix for SVM

4.4 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis is a form of discriminant analysis that identifies the k-dimensional projection that creates the greatest between-group separation based on Y. It assumes and works best when the data contains a linear boundary. In order to implement this model, we used the lda() function from the MASS library. This method yielded an accuracy rate of 0.7220286.



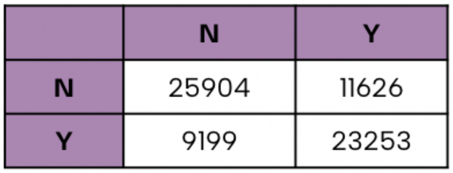
	N	Y
N	25518	9863
Y	9595	25024

Figure 7: Confusion Matrix for LDA

4.5 Quadratic Discriminant Analysis (QDA)

Another model that we implemented to predict Alcoholic Status was the Quadratic Discriminant Analysis, which is a type of discriminant analysis that assumes that the features of different classes have different covariances. The QDA learning method is similar to that of the Linear Discriminant Analysis (LDA) model in that it deals with multiclass classification in addition to performing a discriminant analysis, but the QDA method allows for different variances for each class.

To perform a QDA learning model to our training model, we used the qda() function in R by specifying our model to treat Alcoholic.Status as the Response Variable and including all our selected variables as the input variables. After training this model and testing for its accuracy, we found that the accuracy rate is 0.702423.



	N	Y
N	25904	11626
Y	9199	23253

Figure 8: Confusion Matrix for QDA

4.6 Recursive Partitioning & Decision Trees

Decision trees are another classification of machine learning models that we were interested in implementing with our research of predicting alcoholic status. Decision Trees is a non-parametric learning model, which utilizes a hierarchical structure (hence the "trees" in the name) that uses nodes, branches, and leaves to assist decision-making of outcomes.

In our instance, we used recursive partitioning, a method that is associated with decision trees. In recursive partitioning, a decision is made about which feature to split on and what threshold to use for the split at each node of a tree. We used rpart() function in R to perform a recursive partitioning method model while using the same Response Variable and input variables. Upon running this model, we found that the accuracy rate is 0.4999429, which is a relatively poor performance to our models. This may suggest that we had limitations regarding the variables we inputted to our tree.

4.7 Boosting (XGBoost)

We also leveraged the power of Boosted Classification Tree models in R to predict alcoholic status. The Boosting method is a powerful, scalable machine-learning algorithm that is particularly effective for large and complex datasets. First, the dataset was prepared by converting categorical variables into numeric factors, which is crucial to ensure the model could interpret these variables correctly. Then, we trained the model using the train() function from the XGBoost package in R. Specifically, we used the 'xgbTree' method with 10-fold cross validation. This method yielded an accuracy rate of 0.7262857.

4.8 Gradient Boosting Machines (GBM)

Gradient Boosting Machines is another machine learning method under the Boosting family of models. In GBM, the model uses decision trees, similar to previous methods we employed. What makes it unique is that it uses gradient descent optimization to minimize the chosen loss function (difference between the actual and predicted values), hence the term "gradient" in its name. This method also assigns weights to all models to optimally minimize the loss function—in doing so, it identifies weaker learners and reduces their influence on our final prediction.

In R, we use the gbm() function from the gbm library. In terms of model specification, we used a Gaussian SSE Loss Function, trained the model with 5000 trees and used 10-fold cross validation. This yielded an overall accuracy rate of 0.7313623, and proved to be our best performing model.

5 Results and Final Recommendations

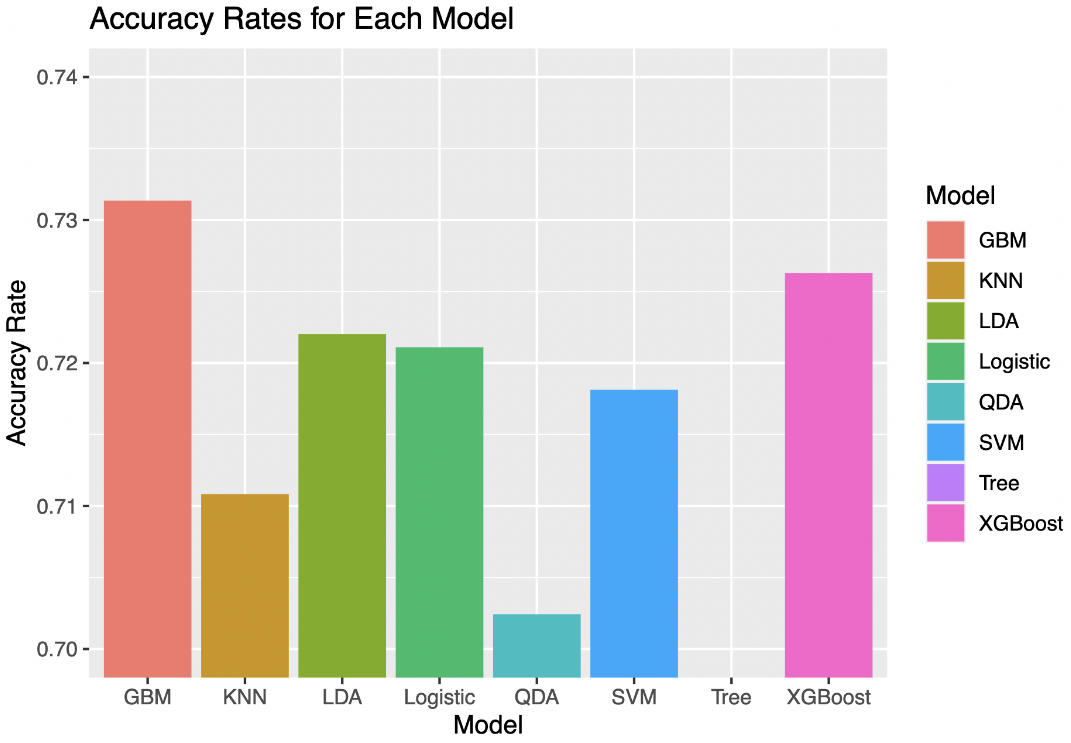


Figure 9: Bar Chart of Model Accuracy Rates

Rank	Model	Accuracy.Rate
1	GBM	0.73136
2	XGBoost	0.72628
3	LDA	0.72202
4	Logistic	0.72110
5	SVM	0.71813
6	KNN	0.71083
7	QDA	0.70242
8	Tree	0.49994

Figure 10: Table of Model Accuracy Rates (Ranked by Descending Order)

We summarize our findings through the bar chart and table above. Overall, we concluded that our best performing model was the Gradient Boosting Machines model based on accuracy rate. The XGBoost model was a close second with an accuracy rate 0.00508 worse than GBM. The remaining models are ranked as follows (in descending order): LDA, Logistic, SVM, KNN, QDA, and Tree.

6 Limitations and Further Investigations

It is important to acknowledge the limitations of our study and suggest improvements that could be applied to potential future investigations.

From our results, we've learned that there is much room for improvement when it comes to maximizing our predictive power. In the realm of K-Nearest Neighbors (KNN), it's evident that a more exhaustive search for an optimal K value could have substantially enhanced the accuracy of our model. The lack of an optimal K value potentially impacted the predictive power of our KNN model.

Similarly, within logistic regression, the model might have been adversely affected by the presence of complex or non-linear relationships inherent in our dataset. These complexities were further apparent in Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) since these techniques are sensitive to specific situations and underlying assumptions.

Scrutinizing the accuracies obtained, the tree model's performance was notably concerning, with an accuracy of 0.49 (comparable odds to simply guessing or flipping a coin). This low accuracy relative to our other models highlights a significant limitation, indicating substantial room for improvement and optimization within our tree model architecture. If we had more time, we would look into how we could improve the performance of that class of models.

Finally, since all our models fell between one to two percentage points of each other in terms of accuracy rate (excluding the Tree model), we can infer that our dataset itself may be the limiting factor in our predictions. Thus, another potential further investigation would be to explore better imputation techniques to see if that improves our accuracy rate more than if we just continued to hone our models. Through this work, we hope to improve our predictive abilities in order to help reduce alcoholism and its negative effects on our society.

7 References

1) IBM. "What is a Decision Tree." IBM, www.ibm.com/topics/decision-trees.
2) Harrell Jr, F. E. (2021). Hmisc: Harrell Miscellaneous. https://cran.r-project.org/web/packages/Hmisc/index.html
3) Mitchell, J & Collen, Jacob & Petteys, S & Holley, Aaron. (2011). A simple reminder system improves venous thromboembolism prophylaxis rates and reduces thrombotic events for hospitalized patients. Journal of thrombosis and haemostasis : JTH. 10. 236-43. 10.1111/j.1538-7836.2011.04599.x.
4) Natekin, A., &amp; Knoll, A. (2013, October 21). Gradient Boosting Machines, a tutorial. Frontiers. https://www.frontiersin.org/articles/10.3389/fnbot.2013.00021/full
5) Stenetorp, P & Pyysalo, Sampo & Topic, Goran & Ohta, Tomoko & Ananiadou, Sophia & Tsujii, Jun'ichi. (2012). brat: a Web-based Tool for NLP-Assisted Text Annotation.The 3th Conference of the European Chapter of the Association for Computational Linguistics; Avignon, France. 102-107.
6) Van Buuren, S., Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. https://cran.r-project.org/web/packages/mice/index.html
7) Van Buuren, S., Groothuis-Oudshoorn, K. (2021). Multiple Imputation by Chained Equations with miceRanger: An iterative series of predictive models. miceRanger: Imputation with Random Forests in R. Retrieved from https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html#:~:text=Multiple%20Imputation%20by%20Chained%20Equations,iterative%20series%20of%20predictive%20models.
8) World Health Organization. (n.d.). Harmful use of alcohol. World Health Organization. https://www.who.int/health-topics/alcohol#tab=tab_1
9) Zach. "Introduction to Quadratic Discriminant Analysis." Statology, 2 Nov. 2020, www.statology.org/quadratic-discriminant-analysis/