Predicting Alcoholic Status Using Person's Vitals

Loretta Hu, Yanle Lyu, Yiwen Zhang

Abstract

This Kaggle project aims to utilize statistical learning techniques to predict alcoholic status from the provided alcohol-drinking dataset. This document comprehensively outlines our methodology for constructing a classification model. It encompasses an overview, data exploration, visualization and cleaning, selecting features, building the model, result analysis, and discussing limitations. Our final model employs the neural network method, giving us a Kaggle score of 0.72996 and ranking at 33rd overall.

I. Introduction

This project explores utilizing vital signs to predict alcoholism conditions. Via a data-driven approach, it is possible to identify early signs and prevent long-term health consequences. This research aims to improve alcoholism prediction and intervention strategies.1

Alcoholism, or alcohol use disorder, is a widespread issue that significantly impacts individuals, families, and communities across the globe. It is characterized by an inability to manage drinking habits and is often associated with a range of physical, psychological, and social problems. Globally, alcohol misuse is a major cause of mortality and morbidity, with an estimated 95,000 deaths in the United States alone attributed to alcohol-related causes annually. The risks of alcoholism include liver disease, cardiovascular problems, mental health issues, and a higher likelihood of accidents and violence. Early detection and intervention are crucial for effective treatment and prevention of long-term damage.

In this kaggle project, the data set is from the National Health Insurance Service in Korea. The training dataset provided contains 30000 observations, and the testing dataset contains 70000 observations, both numerical and categorical. The dataset contains information on various aspects of analysis on body signals, and the classification based on drinking habits (Alcoholic status)This data consists of training and testing data sets; both have 26 predictors, except that for the testing data set, there is no response variable (Alcoholic.Status).Our mission was to introduce a classification model to predict the target variable Alcoholic Status “Y(Yes)” and “N(No) ” in testing data by selecting key variables.

II. Data Analysis

2.a Variables Exploratory Data Analysis (EDA)

Instead of directly fitting a model and utilizing the model for prediction purposes, we have completed some exploratory data analysis (EDA) upon the numerical and categorical variables. This allows us to gain a better understanding of the data set and potentially become more aware of the underlying relationships and patterns between and within the variables.

2.a.a Numerical EDA

Because the target variable Alcoholic.Status is binary, and is evenly distributed across the two categories, N and Y, we have decided to utilize density plots to obtain an overview of the numerical variables. Specifically for the density plots, the area of overlap indicates the range of values shared by both categories. Thus, a low degree of overlapping between the two categories implies a clear distinction, and thus the variable could be significant within a model.

The following are two iconic density plots we have observed with the mentioned pattern.

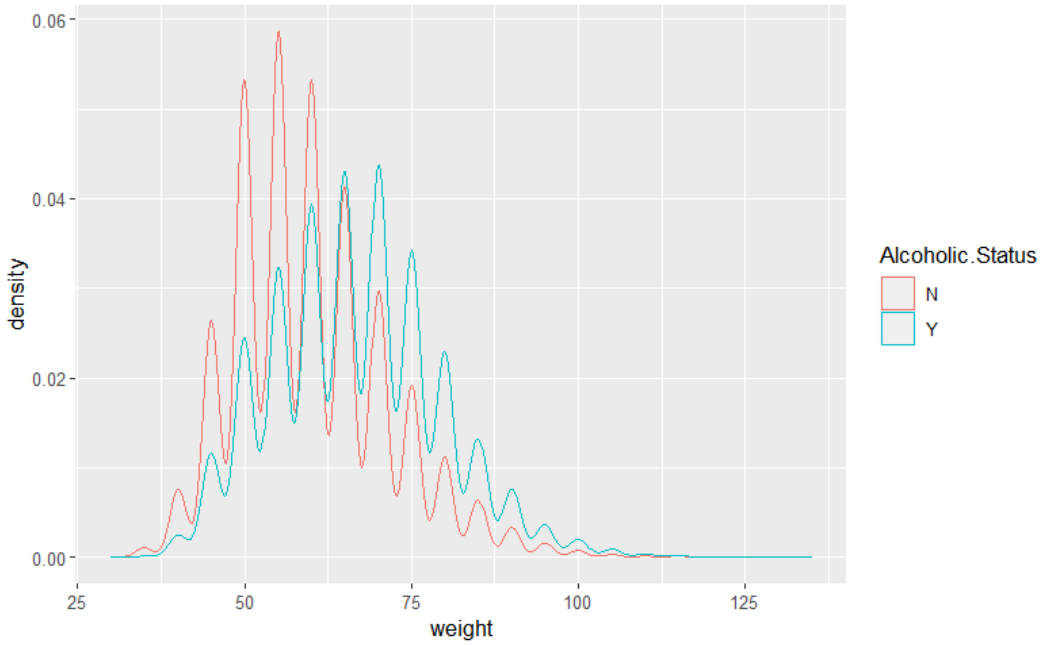

Figure 1: The density of weight vs. Alcoholic.Status

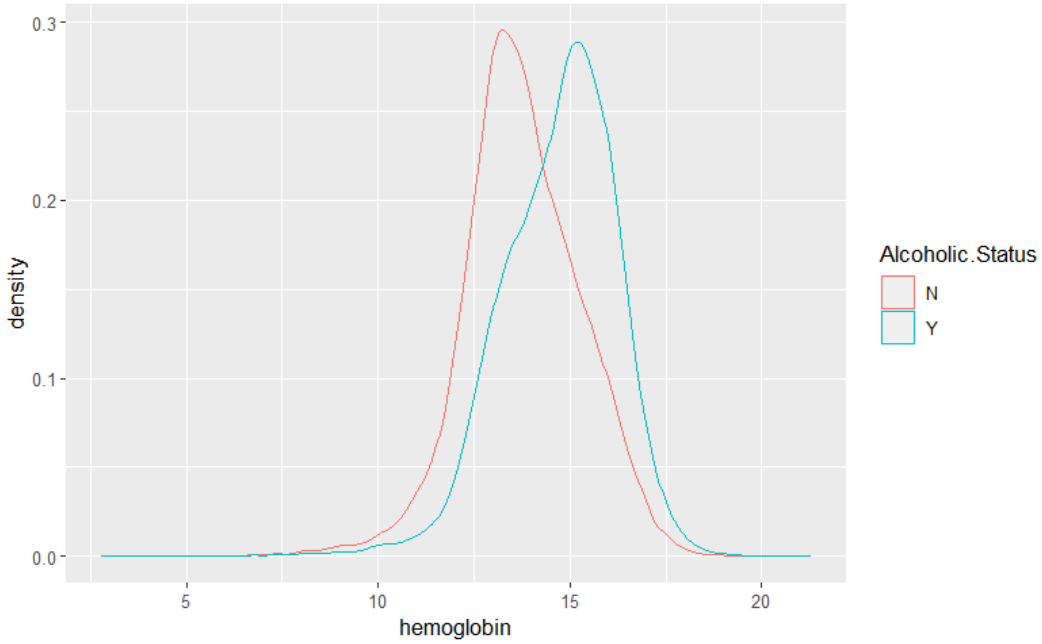

Figure 2: The density of hemoglobin vs. Alcoholic.Status

2.a.b Categorical EDA

Similarly, we have utilized stacked bar plots to visualize the inner distribution of Alcoholic.Status across the levels of categorical variables. Specifically for stacked bar plots, if we observe the Alcoholic.Status is highly unevenly distributed across the levels, then it is probable that the categorical variable is predictive for the Alcoholic.Status.

The following are two bar plots we have developed to be potentially helpful for establishing models.

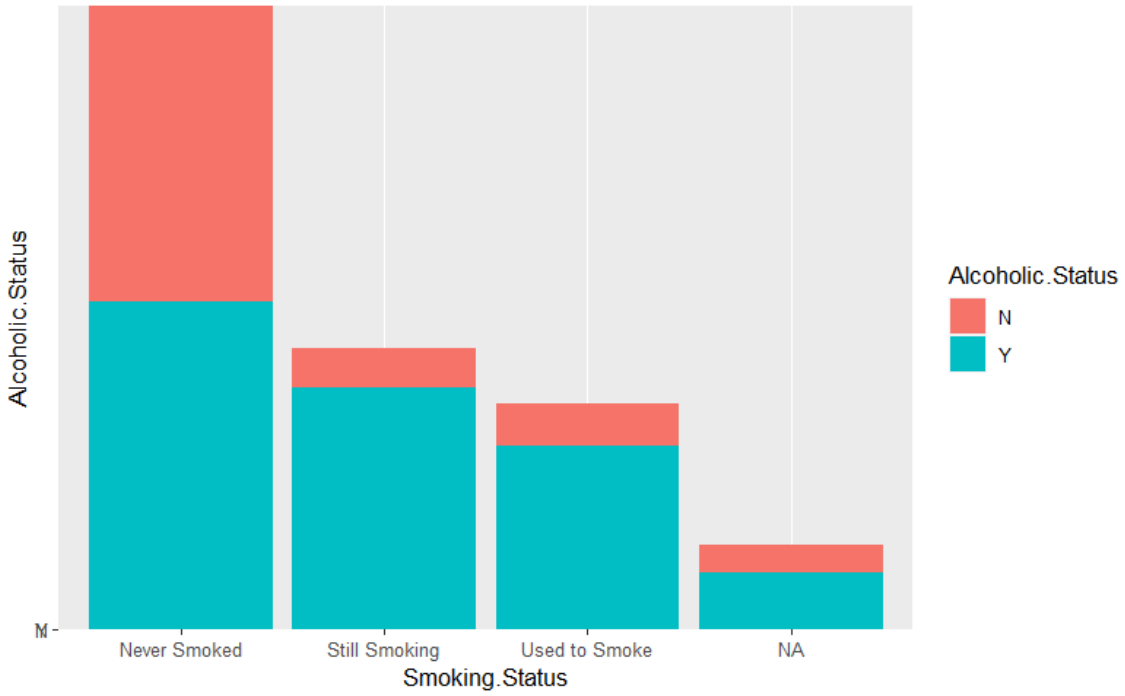

Figure 3: The distribution of Smoking.Status vs. Alcoholic.Status

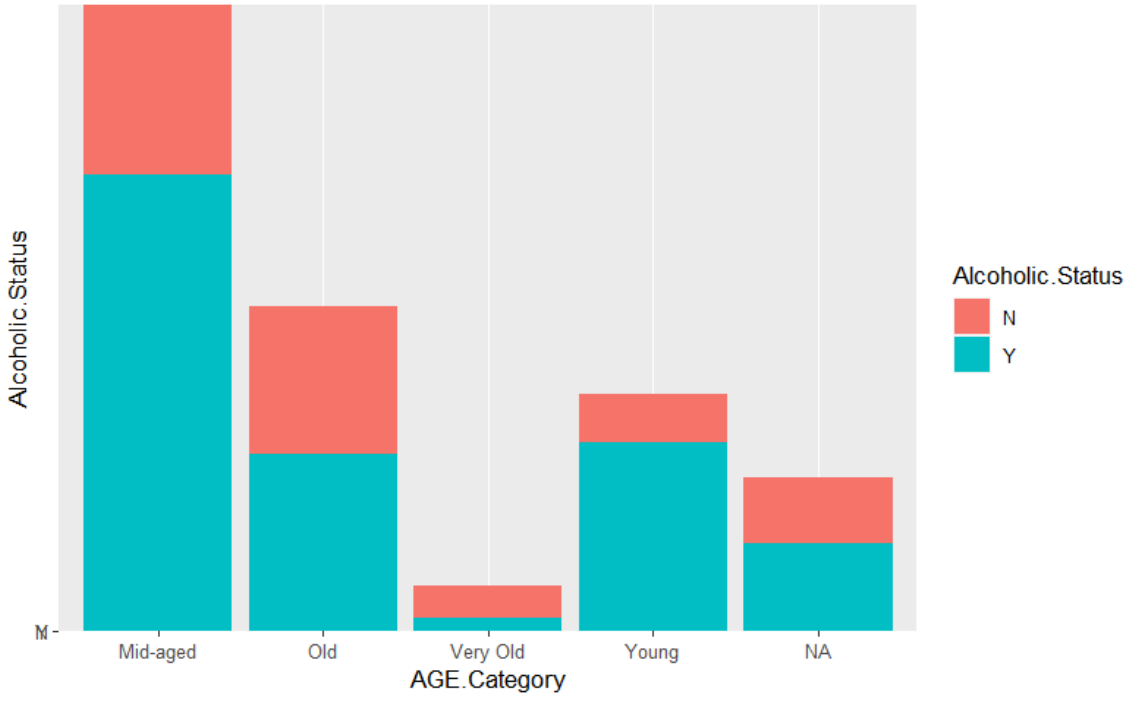

Figure 4: The distribution of AGE.Category vs. Alcoholic.Status

The levels of all categorical variables have also been identified.

Variable	Levels	Variable	Levels
AGE. Category	4	BMI.Category	4
sex	2	Smoking.Status	3
hear_left	2	urine_protein	6
hear_right	2		

Table 1: List of categorical variables and corresponding levels

2.b Clean data -- finding and dealing with missing values

The data set contains missing values encoded by NA values across all of the variables in both training and test sets, excepting the target variable, Alcoholic.Status. Because most of the machine learning methods adopted in this project or introduced in the class require non-NA values for all the variables used, imputing missing values becomes a necessity.

Initially, in the exploration phase, we attempted to use simplistic imputation methods, which entails utilizing mode in replace of NA values for the categorical variables and utilizing mean in replace of NA values for the numerical variables. Due to the limited performance, we later switched to using the amelia package to impute all variables. Nevertheless, the amelia package also performs poorly when imputing categorical variables. So we have ultimately

Lastly, we have settled with the method of using KNNImputer to impute the missing numerical values, and missForest for the missing categorical values. This merged use of methods is a compromise to the long imputation time with missForest and the potential memory issues.

2.c Variables Selection

After exploratory data analysis and necessary data cleaning, we started the feature selection process in dedication to optimize the model performance and improve the prediction

accuracy, especially considering that there exist 26 predictors. The feature selection is essentially completed with backward elimination with Bayesian Information Criterion (bBIC). We have also adopted bBIC upon all the variables, regardless of the class being numerical or categorical.

We choose to use bBIC for feature selection mainly because it offers an insight of a more simplified model, which is achieved from iteratively removing the least significant features based on the likelihood function. The simplified model is beneficial for large datasets as it penalizes harshly on the models with more parameters, and thus improves the computational efficiency, allows interpretation, and prevents overfitting.

Nonetheless, choosing bBIC for feature selection also obtains its limitations. As mentioned previously, the method penalizes harshly upon models with more parameters, which leads to possible underfitting scenarios. The decision of removing a feature is also based upon an arbitrary threshold, which may not necessarily align with the underlying data structure.

In conclusion, the finalized model includes the following variables. The categorical variables include Smoking.Status, AGE.Category, sex; and the numerical variables include HDL_chole, LDL_chole, triglyceride, SGOT_ALT, gamma_GTP, age, height, weight, waistline, hemoglobin, DBP.

III. Methods & Models

3.a Logistic Regression

Logistic Regression's main advantage is its simplicity and speed, making it a suitable choice for a preliminary analysis. The model is easy to interpret and implement, which aids in understanding the relationship between features and the target variable.

However, a significant limitation of Logistic Regression is its tendency to overfit, especially when dealing with complex or large datasets. This can lead to poor generalization on unseen data. Furthermore, it assumes linearity between independent variables and the log odds, which may not always be true in real-world scenarios.

In terms of performance on this dataset, Logistic Regression showed a **CV score of 0.7283 (+/- 0.0049) and true accuracy of 0.7165**. It served as a baseline for comparison with more complex models. While it could identify some patterns in the data, its overall predictive accuracy was not as high as some of the other models, like Random Forest or XGBoost.

3.b Random Forest

Random Forest operates by creating a multitude of decision trees at training time and outputting the class that is the mode of the classes from individual trees. This ensemble learning method is particularly effective for classification and regression tasks, offering advantages in handling large datasets with higher dimensionality.

One of the main strengths of Random Forest is its robustness against overfitting, a common issue in complex models. It achieves this by constructing each tree on a different subset of the data, using random feature selection. This diversity among trees reduces variance without significantly increasing bias, leading to better generalization of unseen data.

Random Forest also handles mixed data types (numerical and categorical) effectively, and can automatically handle missing values. Moreover, it provides insights into feature importance, aiding in understanding which variables most significantly impact the prediction.

However, Random Forest has its limitations. It can be computationally intensive, especially with large datasets and a high number of trees. The model's complexity can also make it less interpretable than simpler models, like Logistic Regression.

In terms of performance on the dataset, Random Forest obtained a **true accuracy of 0.72256** and better handling of the mixed data types, demonstrating its effectiveness in this application.

3.c LDA

Linear Discriminant Analysis (LDA) is a classification method that seeks to separate classes through linear combinations of features, maximizing the ratio of between-class variance to within-class variance.

It excels in simplicity and low computational cost, making it suitable for straightforward classification tasks. LDA's assumption of Gaussian distribution and equal covariance across classes, however, limits its flexibility, especially in datasets with complex structures.

Despite these limitations, LDA performed not too badly in the project. We ultimately obtained a **CV score of 0.7247 (+/- 0.0044) and true accuracy of around 0.70**.

3.d QDA

Quadratic Discriminant Analysis (QDA) extends Linear Discriminant Analysis (LDA) by allowing each class to have its covariance matrix, making it more flexible. This feature enables QDA to model complex relationships and capture interactions between variables better than LDA.

However, QDA's increased flexibility can also be a drawback, particularly with smaller datasets, as it might overfit. Moreover, QDA assumes that the predictors are normally distributed, which may not hold in all cases.

In the project, QDA showed less accuracy compared to models like Random Forest or XGBoost. Its performance was hindered by its strict assumptions and the complexity of the data.

CV score of 0.6925 (+/- 0.0066) and true accuracy of around 0.65 means that a quadratic function is not the best choice for boundaries.

3.e XGBoost

XGBoost is an advanced implementation of gradient boosting algorithms. It stands out for its efficiency, flexibility, and portability. XGBoost improves upon the standard gradient boosting method by introducing regularization to prevent overfitting, making it more effective on complex datasets.

It's particularly adept at handling large datasets and offers parallel processing, tree pruning, and cross-validation capabilities, enhancing speed and performance. XGBoost's ability to manage missing values and support various objective functions makes it versatile for various applications.

In this project, XGBoost demonstrated superior performance, with **CV score of 0.7328 (+/- 0.0028) and true accuracy of around 0.727**. It outclassed simpler models like Logistic Regression and even complex ones like Random Forest. Its ability to handle the dataset's intricacies efficiently made it a powerful tool for predicting alcoholic status. However, its complexity can be a drawback, requiring careful tuning of parameters and a good understanding of its functioning to achieve optimal results.

3.f Simple Neural Network

The Simple Neural Network used in the project is a multi-layer perceptron model designed for the classification task. Its architecture comprises:

Input Layer: Processes features with a fully connected linear layer (nn.Linear). The number of input nodes matches the number of features in X_train_scaled.

Hidden Layers: First hidden layer with 12 nodes, followed by a LeakyReLU activation function. The second hidden layer with 8 nodes, also uses LeakyReLU for non-linearity.

Dropout Layer: A dropout rate of 50% (nn.Dropout(0.5)) is applied after the second hidden layer to reduce overfitting.

Output Layer: The final layer has 2 nodes (for binary classification) and applies a log softmax function for the output.

The architecture of the model is the following:

```
class NeuraNetwork(nn.Module):
        def __init__(self):
            super(NeuraNetwork, self).__init__()
            self.layer1 = nn.Linear(X_train_scaled.shape[1],
12)
            self.layer2 = nn.Linear(12, 8)
            self.dropout1 = nn.Dropout(0.5)
            self.layer3 = nn.Linear(8, 2)

        def forward(self, x):
            x = nn.LeakyReLU()(self.layer1(x))
            x = nn.LeakyReLU()(self.layer2(x))
            x = self.dropout1(x)
            x = torch.log_softmax(self.layer3(x), dim=1)
            return x
```

This architecture leverages non-linear activation (LeakyReLU) to capture complex patterns in the data, and dropout for regularization. The choice of a softmax function in the output layer is appropriate for classification tasks, as it helps in predicting probability distributions over classes.

In terms of performance, this simple neural network model demonstrated a **CV score of 0.75 with a true accuracy of 0.72996**. While it was more complex than models like Logistic Regression, it provided moderate predictive accuracy, emphasizing the potential of neural networks in handling intricate relationships in data.

IV. Discussion and Limitation

The approach to handling missing data in our dataset plays a critical role in influencing the performance of our models. Various imputation strategies, such as mean/median imputation, k-NN imputation, or more advanced methods like MICE (Multiple Imputation by Chained Equations), offer different results. An important aspect to consider is the pattern of missing data. If the missingness is not random, it necessitates a tailored approach, as it could significantly affect the study's outcomes.

Furthermore, our model's foundation is the backward BIC method for feature selection. However, the justification for considering this method as the most optimized choice is not clear-cut. Due to the nature of BIC, which tends to favor simpler models, there is a risk of underfitting, potentially overlooking important variables. The variation in feature selection methods can considerably impact the model's final performance, highlighting the need for a thorough evaluation of alternative selection strategies.

Additionally, our exploration into neural networks was a calculated decision, given their capability to model complex relationships. Nevertheless, neural networks demand significant data and computational resources. Their performance, especially in contexts with smaller or less complex datasets, may not surpass that of simpler models. This phenomenon was evident in our findings, underscoring the necessity to weigh the advantages of complex models against their requirements and suitability for the dataset at hand.

In the end, the complexity of our dataset poses a substantial challenge in fully deciphering the relationships between predictors and the response variable. This complexity limits the potential for enhancing model performance, as it is difficult to capture the intricate interactions among variables using our current modeling approaches.

Due to time constraints, extensive hyperparameter tuning is often limited. Future studies should employ automated methods like grid search, random search, or Bayesian optimization for efficient exploration of parameter spaces. These techniques allow for more thorough tuning within restricted time frames. Also, the design of neural networks requires a careful balance between complexity, available data, and computational resources. Future research should experiment with varying the number of layers and nodes, activation functions, and regularization methods. It's crucial to recognize that more complex models do not always yield better results and can lead to overfitting, particularly with limited datasets.

V. Conclusion

Our project's endeavor to predict alcoholic status using persons' vitals, leveraging a dataset from Korea's National Health Insurance Service, has yielded significant insights. Key strategies like effective data imputation methods (KNNImputer and missForest) and Bayesian

Information Criterion (bBIC) for feature selection stood out for their impact on the models' performance. The study meticulously navigated through both numerical and categorical data, ensuring a comprehensive understanding of the dataset.

Among the various models evaluated, Logistic Regression provided a baseline understanding, while Random Forest and XGBoost excelled in handling the data's complexity and dimensionality. Notably, the simple neural network, with its multi-layer perceptron architecture, demonstrated the potential for sophisticated deep learning techniques in predictive analytics.

The project's findings accentuate the necessity of a nuanced approach in predictive modeling, particularly in the healthcare sector. It highlighted the importance of model selection and parameter tuning by the dataset's characteristics. Future directions could involve exploring more intricate neural network configurations and further refining data preprocessing and feature selection techniques.

VI. References

1. Gandhi, Roshi. “Support Vector Machine — Introduction to Machine Learning Algorithms.” June, 2018.
2. Ye, Andre. “MissForest: The Best Missing Data Imputation Algorithm?” August, 2020. https://towardsdatascience.com/missforest-the-best-missing-data-imputation-algorithm-4d01182aed3