P8106 Midterm Project Report

Shuchen Dong(sd3731)

Contents

1 Exploratory Analysis and Data Visualization

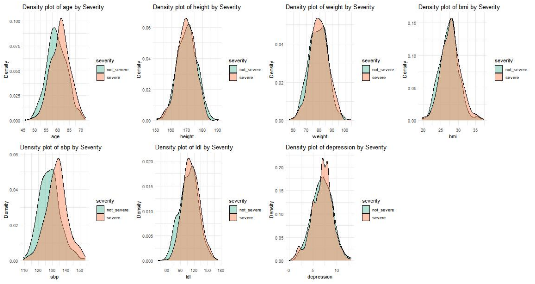
Our dataset includes COVID-19 information for 1000 participants, split into training (80%) and testing (20%) sets. Exploratory analysis is then performed on the training set.

The dataset without *id* contains 13 predictors: seven of them are continuous variables, and six of them are categorical variables. The *severity* is the response which is also a catergorical variable. First, use the **skim()** function from the **skimr** package is used to generate summaries of the trainData and testData datasets, offering structured information like column names, data types, counts and percentages of missing values, mean, median, and other statistical information.

Continuous variables: Figure 1 show density plots by *severity* for all continuous variables by **lapply**, providing insights into how different variables vary between *non_severe* and *severe* COVID-19 cases by **ggplot2**. And we can conclude that most continuous variables follow a normal distribution.

Categorical variables: Figures 2 show box plots for all categorical variables, which depict the distribution of the response across each level by severity.

From the correlation plot in **Figure 3**, the first visualization uses the **GGally** package with ggpairs function, while the second plot utilizes the **corrplot** package with **corrplot** function. We can see that among the continuous variables, BMI and weight have a strong positive correlation, while BMI and height exhibit a negative correlation. SBP, LDL, and age are moderately correlated with one another. Strongly correlated predictors may lead to multicollinearity, affecting the model's predictive performance. Thus, additional model training with cross-validation (CV) is necessary.



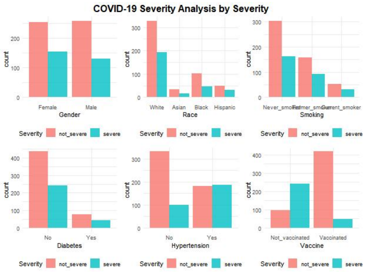


Figure1. Density plot of continuous variable by Severity

Figure2. Box plot of catergorical variable

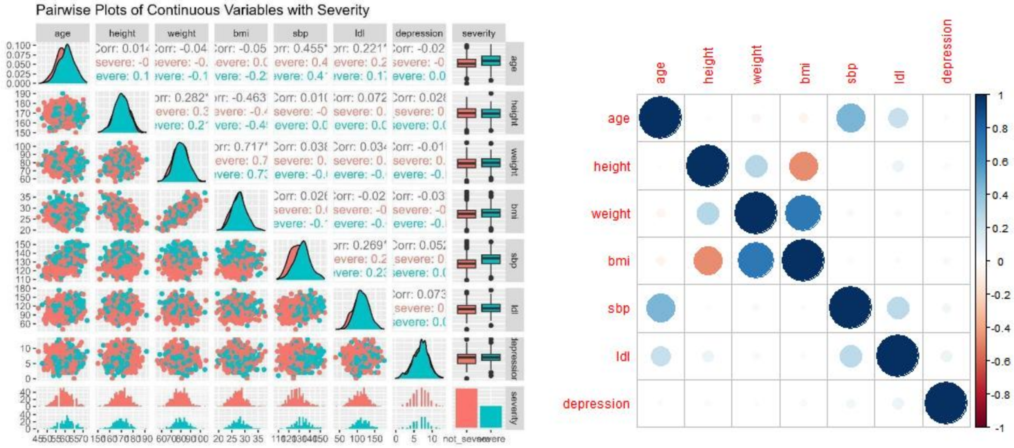


Figure3. Correlation plot

2 Model Training

The analysis employs classification methods well-suited for predicting binary outcomes(*severity*), utilizing 13 regression models trained via the **caret** package. Each model is rigorously evaluated through repeated 10-fold cross-validation, initiated with a consistent seed (3731) for reproducibility, and the tuning process is visualized to enhance parameter selection.

2.1 Logistic Regression

It assumes linearity in the logit, independence of observations, no multicollinearity, and no influential outliers. The model is fitted with the **train** function using the **glm** method for generalized linear models.

2.2 Penalized logistic regression

It combines L1 and L2 regularization to prevent overfitting, maintaining assumptions similar to logistic regression with the **glmnet** method. For training, it explores alpha (0 to 1) and lambda (exp(-13) to exp(-3)) values through a tuning grid.

2.3 Elastic Net

Like penalized regression, balances L1 and L2 penalties and handles correlated variables without assuming overall linearity using the **glmnet** method. The tuning grid spans alpha (0 to 1 in 11 steps) and lambda (exp(2) to exp(-8) over 50 points).

2.4 Generalized Additive Model (GAM)

It enhances model flexibility by incorporating non-linear smooth functions, requiring independent observations without predetermined form for these relationships.

2.5 Multivariate Adaptive Regression Splines (MARS)

It uses hinge functions for piecewise linear modeling, allowing non-linearity within defined regions. The tuning grid ranges from 1 to 5 for degree and 2 to 14 for nprune.

2.6 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is employed to find a linear combination of features that best separates two or more classes of events. It assumes that each class has normally distributed data with identical covariance matrices and that observations are independent. The **lda** method in R is used to fit the model, optimized through cross-validation to enhance the classification performance, with the area under the ROC curve as the performance metric.

2.7 Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis (QDA) allows for each class to have its own covariance matrix, providing a more flexible classification boundary than LDA. Like LDA, it assumes classes are normally distributed and observations are independent. The **qda** method is applied for fitting the model, utilizing the area under the ROC curve to measure model effectiveness, guided by a robust cross-validation process.

2.8 Naive Bayes (NB)

It simplifies computation by assuming feature independence within classes. The tuning grid adjusts kernel usage (TRUE/FALSE), fL (fixed at 1), and smoothing (0.1 to 5 in increments).

2.9 Random Forest

Random Forest models build numerous decision trees and merge their results to improve accuracy and stability. They assume that the individual trees are independently constructed and robust against noise with enough ensemble members. In this setup, the tuning grid explores the **mtry** parameter (number of variables sampled at each split, ranging from 1 to the total feature count), **splitrule** (using "gini" for classification), and **min.node.size** (ranging from 2 to 16 in increments of 2) to control node size and tree complexity.

2.10 Classification Trees(rpart)

Classification Trees classify data by splitting it according to branching rules based on the value ranges of input variables. The **rpart** method uses the **cp** (complexity parameter) to prune insignificant branches. The tuning grid **expand.grid** explores a range of **cp** values (from exp(-6) to exp(-4) over 50 intervals) to optimize tree complexity, ensuring a good balance between simplicity and predictive power.

2.11 Adaboost

Adaboost combines multiple weak learners into a weighted ensemble to improve predictive accuracy. It assumes that even weak classifiers can achieve high accuracy through strategic weighting. The tuning grid for Adaboost in the **gbm** method uses **n.trees** (2000 to 5000), **interaction.depth** (1 to 10), **shrinkage** (0.001 to 0.003), and **n.minobsinnode** (fixed at 1) to explore parameter combinations, optimizing the number and depth of boosting stages.

2.12 Support Vector Machine (SVM)

Linear SVM finds straight-line boundaries in feature space, while Radial SVM uses an RBF kernel to create separability in transformed space. Both tune C (exp(-3) to exp(6) for linear and similar plus sigma (exp(-4) to exp(1)) for radial.

After training all the models, the function **resamples()** is used to assess model performance, and the summary plots are shown below. The **Adaboost** model shows the highest median ROC value among all the models and was chosen as the final model.

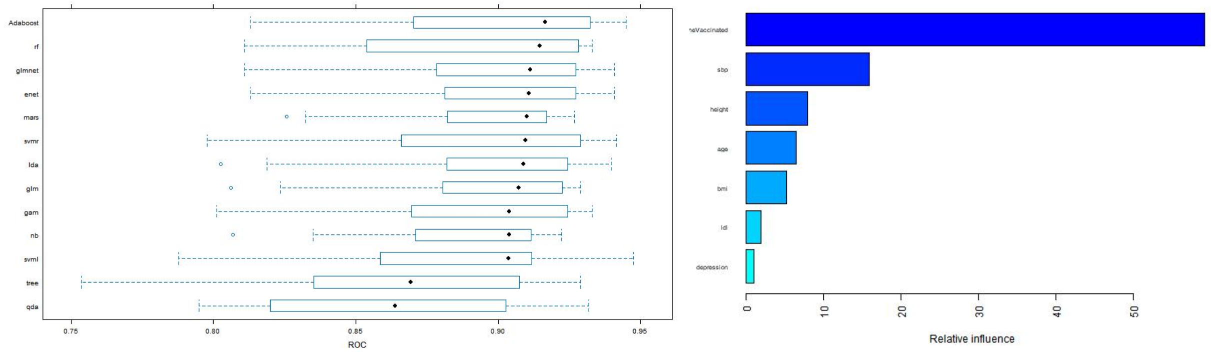


Figure4. Resample plot

3 Results

Based on the analysis using gradient boosting with AdaBoost, the optimal hyperparameters selected were 4000 trees, an interaction depth of 2, a shrinkage of 0.001, and a minimum number of observations in a node of 1. The most influential predictors for COVID-19 severity, as determined by variable importance analysis, were "vaccineVaccinated," "sbp" (systolic blood pressure), "height," "age," and "bmi" (body mass index), with relative influence percentages of 59.24%, 15.98%, 7.93%, 6.45%, and 5.27%, respectively.

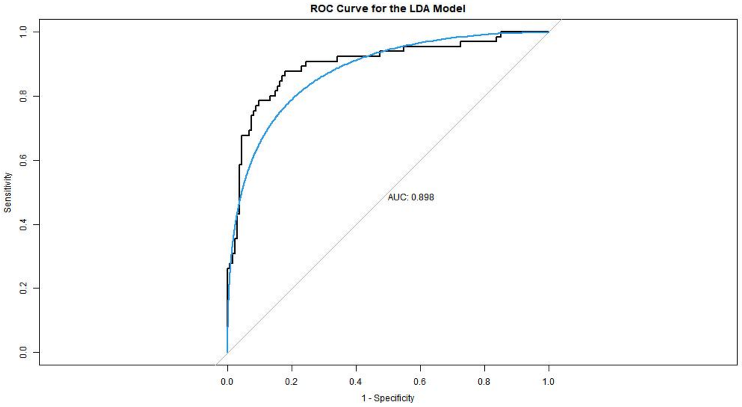
Based on the AdaBoost model selected, the test results showed an accuracy of 86.5% and a kappa coefficient of 0.6809. The confusion matrix indicated a sensitivity of 72.31% and a specificity of 93.33% for predicting severe cases of COVID-19. The positive predictive value (PPV) was calculated at 83.93%. These metrics suggest that the AdaBoost model performs well in distinguishing severe cases of COVID-19 from non-severe ones, demonstrating its effectiveness in clinical prediction tasks.

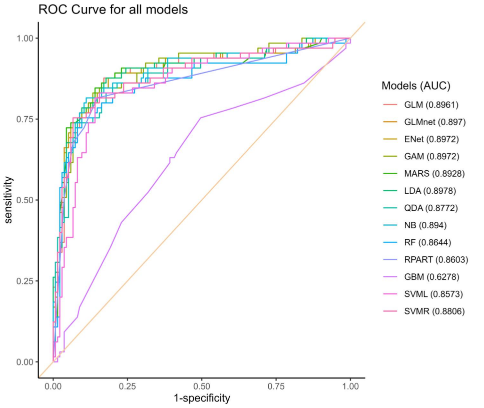
4 Conclusion & Discussion

In conclusion, the AdaBoost model, with its optimized hyperparameters, demonstrated strong predictive performance in distinguishing severe cases of COVID-19 from non-severe ones. Key predictors such as vaccination status, systolic blood pressure, height, age, and body mass index emerged as influential factors in determining disease severity. These findings underscore the importance of leveraging advanced machine learning techniques, like AdaBoost, in healthcare settings for accurate risk stratification and clinical decision-making. Moving forward, continued refinement and validation of predictive models will be essential for enhancing our understanding of COVID-19 outcomes and improving patient care.

After evaluating model performance on the test data, the LDA model stood out with the highest AUC value of 0.8977778 among all considered models, indicating strong discriminatory power in identifying severe COVID-19 cases. Additionally, the LDA model demonstrated an accuracy of 83.5% and a kappa coefficient of 0.6341, underscoring its predictive capability. Despite a misclassification error rate of 16.5%, the overall performance metrics support the effectiveness of the LDA model in predicting COVID-19 severity.

Considering these results, the LDA model appears promising for COVID-19 severity prediction. However, model selection should also weigh factors like interpretability, computational efficiency, and practical applicability in clinical settings. Future research could explore ensemble methods or hybrid approaches to further enhance predictive accuracy and robustness in COVID-19 severity prediction models.


ROC Curve for the LDA Model
Sensitivity
AUC: 0.898
1 - Specificity


ROC Curve for all models
sensitivity
1-specificity
Models (AUC)
GLM (0.8961)
GLMnet (0.897)
ENet (0.8972)
GAM (0.8972)
MARS (0.8928)
LDA (0.8978)
QDA (0.8772)
NB (0.894)
RF (0.8644)
RPART (0.8603)
GBM (0.6278)
SVML (0.8573)
SVMR (0.8806)