Midterm Report

Predicting Recovery Time of Patients with Covid-19

Yi Huang (yh3554)

2023-04-05

Contents

Abstract

This project applies varies regression methods to predict the recovery time of participants with Covid-19 to estimate the recovery time of these participants. The ultimate goal is to develop a prediction model for recovery time and identify important risk factors for long recovery time. The dataset is from a study that combines three existing cohort studies that have been tracking participants for several years.

1 Introduction

1.1 Background

To gain a better understanding of the factors that predict recovery time from COVID-19 illness, a study was designed to combine three existing cohort studies that have been tracking participants for several years. The study collects recovery information through questionnaires and medical records, and leverages existing data on personal characteristics prior to the pandemic. The ultimate goal is to develop a prediction model for recovery time and identify important risk factors for long recovery time.

1.2 Data Description

The dataset is a random sample of 2000 participants draw from `"recovery.RData` includes 16 variables and 2000 observations.
Description of each variable:
- Gender (gender): 1 = Male, 0 = Female;
- Race/ethnicity (race): 1 = White, 2 = Asian, 3 = Black, 4 = Hispanic;
- Smoking (smoking): Smoking status; 0 = Never smoked, 1 = Former smoker, 2 = Current smoker;
- Height (height): Height (in centimeters);
- Weight (weight): Weight (in kilograms);
- BMI (bmi): Body Mass Index; BMI = weight (in kilograms) / height (in meters) squared;
- Hypertension (hypertension): 0 = No, 1 = Yes;
- Diabetes (diabetes): 0 = No, 1 = Yes;
- Systolic blood pressure (SBP): Systolic blood pressure (in mm/Hg);
- (10)LDL cholesterol (LDL): LDL (low-density lipoprotein) cholesterol (in mg/dL);
- Vaccination status at the time of infection (vaccine): 0 = Not vaccinated, 1 = Vaccinated;
- Severity of COVID-19 infection (severity): 0 = Not severe, 1= Severe;
- Study (study): The study (A/B/C) that the participant belongs to;
- Age: age of participants;
- Time to recovery (tt_recovery_time): Time from COVID-19 infection to recovery in days;
- ID: unique id of each participant.

1.3 Data Cleaning

Package `tidyverse` and `dplyr` are used for data cleaning and wrangling. Variable `ID` is omitted before data wrangling. Table 1 includes the summary of all variables including 6 continuous and 9 categorical predictors and 1 continuous response. Based on the summary, there is no missing data. Then, factor all the categorical data before data partition. Apply `caret` package to split the data into 70% train data and 30% test data. Based on the Table 1, the distribution of predictors are quite similar across different studies.

Table 1: **Summary of Dataset by Study Group**

Characteristic	**A**, N = 402	**B**, N = 1,209	**C**, N = 389
age	60.0 (57.0, 63.0)	60.0 (57.0, 63.0)	60.0 (57.0, 63.0)
height	170 (166, 173)	170 (166, 174)	170 (166, 174)
weight	79 (75, 84)	80 (76, 85)	80 (75, 85)

Characteristic	**A**, N = 402	**B**, N = 1,209	**C**, N = 389
bmi	27.60 (25.60, 29.30)	27.70 (25.80, 29.50)	27.60 (25.60, 29.40)
SBP	131 (126, 136)	130 (125, 136)	129 (124, 135)
LDL	112 (98, 125)	110 (97, 123)	108 (96, 123)
recovery_time	40 (33, 47)	37 (23, 56)	39 (33, 45)
gender			
0	211 (52%)	613 (51%)	191 (49%)
1	191 (48%)	596 (49%)	198 (51%)
race			
1	271 (67%)	779 (64%)	249 (64%)
2	21 (5.2%)	60 (5.0%)	15 (3.9%)
3	82 (20%)	257 (21%)	83 (21%)
4	28 (7.0%)	113 (9.3%)	42 (11%)
smoking			
0	242 (60%)	724 (60%)	249 (64%)
1	114 (28%)	357 (30%)	109 (28%)
2	46 (11%)	128 (11%)	31 (8.0%)
hypertension			
0	196 (49%)	629 (52%)	227 (58%)
1	206 (51%)	580 (48%)	162 (42%)
diabetes			
0	327 (81%)	1,017 (84%)	326 (84%)
1	75 (19%)	192 (16%)	63 (16%)
vaccine			
0	164 (41%)	471 (39%)	150 (39%)
1	238 (59%)	738 (61%)	239 (61%)
severity			
0	365 (91%)	1,091 (90%)	353 (91%)
1	37 (9.2%)	118 (9.8%)	36 (9.3%)

2 Exploratory analysis and data visualization:

In this section, I use several function to explore the data and identify the patterns by creating data visualizations. The function `ggpairs()` from `GGALLY` package allows to build scatterplot matrix, where the variable distribution is displayed on the diagonal, scatterplot of each pair of continuous variale on the left part of the figure, and pearson correlation is on the right figure. The function `ggplot()` from `ggplot2` generate density plot and boxlpots. In function `featureplot()` from `caret` can be used to plot the correlation of variables. Figure 1 to 3 shows various observations related to COVID-19 recovery time and its association with different variables. Highlights that males generally have a longer recovery time than females, white patients have a shorter recovery time compared to all other races, and never smokers recover faster than former and current smokers. Patients with lower weight have a slightly shorter recovery time, and a U-shaped relationship is observed between patient BMI and recovery time. The presence of hypertension and diabetes, as well as severe infections, is associated with a longer recovery time. On the other hand, vaccinated patients tend to recover faster. Additionally, the study identifies certain variables such as patient age, SBP, and LDL cholesterol that do not seem to have a clear association with COVID-19 recovery time. Figure 4 to 6, are a closer look at the distribution of age variable across different studies, which follows almost the same pattern.

3 Model training

This section describes the models used for predicting time to recovery from COVID-19. State the assumptions made by using the models and detailed description of the model training procedure and how to obtained the final model.

This project performs varies regression models on the dataset, including linear regression, K-Nearest Neighbors (KNN), ridge, lasso, lasso 1se, elastic net, partial least squares (pls), generalized additive model (gam), Multivariate Adaptive Regression (mars), random forest, and boosting using cross-validation results on train data. The model with the minimum mean of train RMSE from cross-validation results will be the best model. All models are implemented in `train()` function from `caret` package.

1. Linear model: Linear model assumes a linear relationship between the predictor and response variables (linearity), and assumes that the errors are normally distributed (normality) and have constant variance (homoscedasticity),and that the observations are independent of each other (independence). The linear model is the most basic and assumes a linear relationship between the variables, while the other models allow for more flexible relationships. A method = "lm" argument was used within the `train()` function to specify a linear model fit.

2. KNN (K-Nearest Neighbors): KNN is a non-parametric machine learning algorithm that can be used for classification or regression tasks. The algorithm works by finding the K closest data points to a new input, and using their output values to predict the output value for the new input. It assume data points which exist in close proximity to each other are highly similar, while if a data point is far away from another group it's dissimilar to those data points. With `train()` function to specify the KNN model and use `expand.grid()` with sequence from 1 to 40 by 1.

3. Ridge, lasso, lasso 1se, elastic net regression: Ridge regression uses L2 regularization to prevent overfitting in the model. The method works by adding a penalty term to the loss function of the model that is proportional to the sum of the squares of the model coefficients. Lasso regression uses L1 regularization to prevent overfitting in the model by adding proportional to the sum of the absolute values of the model coefficients as penalty term. Lasso 1 se selecting the optimal regularization parameter for the Lasso regression model. Elastic Net combines the L1 and L2 regularization methods to prevent overfitting in the model by adding L1 and L2 penalties. Ridge, lasso, and elastic net regression models have the same assumption as linear model. A method = "glmnet" argument was used within the `train()`function to specify model fit. Additionally, a grid of tuning parameters for these models were set using the tuneGrid argument within the train() function. The grid contains 100 values of the lambda parameter, ranging from exp(-20) to exp(10) with equal intervals on the log scale for ridge and lasso. The grid contains 100 values of the lambda parameter, ranging from exp(-2) to exp(2) with equal intervals on the log scale for elastic net.

4. Partial Least Squares (PLS): PLS is a multivariate regression method that is used to model the relationship between two sets of variables. The method works by finding the linear combination of the input variables that best explains the variation in the output variables. A method = "pls" argument was used within the `train()` function to specify a PLS model fit. Additionally, a tuning parameter for the model were set using the tuneGrid argument within the `train()` function. The tuneGrid object includes a data frame with a single column ncomp that ranges from 1 to 18, representing the number of components used in the model. It has the same assumptions as the linear model. The preProcess argument is set to "center" and "scale", which means that the training data will be centered and scaled prior to model fitting.

5. Generalized Additive Model (GAM): GAM is a type of regression model that can capture non-linear relationships between the input and output variables. The method works by modeling the input variables as a sum of smooth functions. It has the same assumptions as the linear model. A method = "gam" argument was used within the `train()` function to specify a GAM fit.

6. Multivariate Adaptive Regression Splines (MARS): MARS can model non-linear relationships between the input and output variables. The method works by fitting piecewise linear or nonlinear functions to the data. It has the same assumptions as the linear model. A method = "earth" argument was used within the `train()` function to specify a MARS model fit. Additionally, the `expand.grid()` function is used to generate a grid of tuning parameters. The mars_grid object includes two arguments: degree is set to 1, 2, and 3, representing the number of possible product hinge functions in a single term, and nprune is set to integers between 2 and 17, representing the upper bound on the number of terms in the model. The

tuneGrid argument in the `train()` function uses the mars_grid object to specify the parameters for model tuning.

7. Bagging, random forest, boosting regression: Bagging is a machine learning techniques that used to reduce variance and improve the accuracy of a model. It works by training multiple models on different subsets of the training data and then combining their predictions. Random Forest used for classification and regression tasks. The algorithm works by training an ensemble of decision trees on different subsets of the training data and then combining their predictions. Boosting can be used to improve the accuracy of a model by combining weak learners into a strong learner. The method works by training a sequence of models on the training data, with each model focusing on the instances that were misclassified by the previous model. Use`randomForest()` function to fit Bagging and random forest models, and `train()` function for the boosting regression with `expand.grid()` function for generating a grid of tuning parameters.

4 Results and discussion

In this section, report the final model that you built for predicting time to recovery from COVID-19. Interpret the results. Assess the model's training/test performance.

Training performance

All models performance were evaluated by using 10-folds cross-validation on the training set. The cross-validation results are shown in Figure 7 and Figure 8. Based on the cross-validation results, Random forest has the smallest mean RMSE 20.8315. Thus random forest is the optimal model to predict the recovery time from Covid-19. MARS model has the second smallest mean RMSE 21.5259.

Final model with variable importance

The Random forest model shows that the relative variable importance follows this order (Figure 9): BMI, weight, height, vaccine, gender(male=1), diabetes (yes=1), study, hypertension, LDL, SBP, severity, age, smoking, and race(White=1).

Testing performance

Finally, we estimate the test error for the best and second best models obtained from resampling cross-validation results to evaluate the test performance. Random forest test RMSE is 22.3937, and MARS is 23.1509.

5 Conclusions

This section will summarize findings from the model analysis and discuss the insights gained into predicting time to recovery from COVID-19.

After comparing different methods, the final model implies that male has diabetes and vaccinated with high value of bmi, weight, and height are more likely to recovery soon from Covid-19. The difference in immune system function, and stress endurance level between males and females could be important determinants to explain the significant difference in recovery time between females and male. This project could be improved if we have the information about what types of diabetes they have. Many research paper revealed that people who do not have Type I diabetes are more likely to recovery from Covid compare to people has Type I diabetes.

References

CDC Covid Data Tracker. (2021), Centers for Disease Control and Prevention.

James, Gareth, e. a. (2021), An Introduction to Statistical Learning: With Applications in R., Springer.

Pradhan, A. and Olsson., P.-E. (2021), 'Sex differences in severity and mortality from covid- 19: are males more vulnerable?', Biology of sex differences 11(1), 53.

A Appendix: Figures

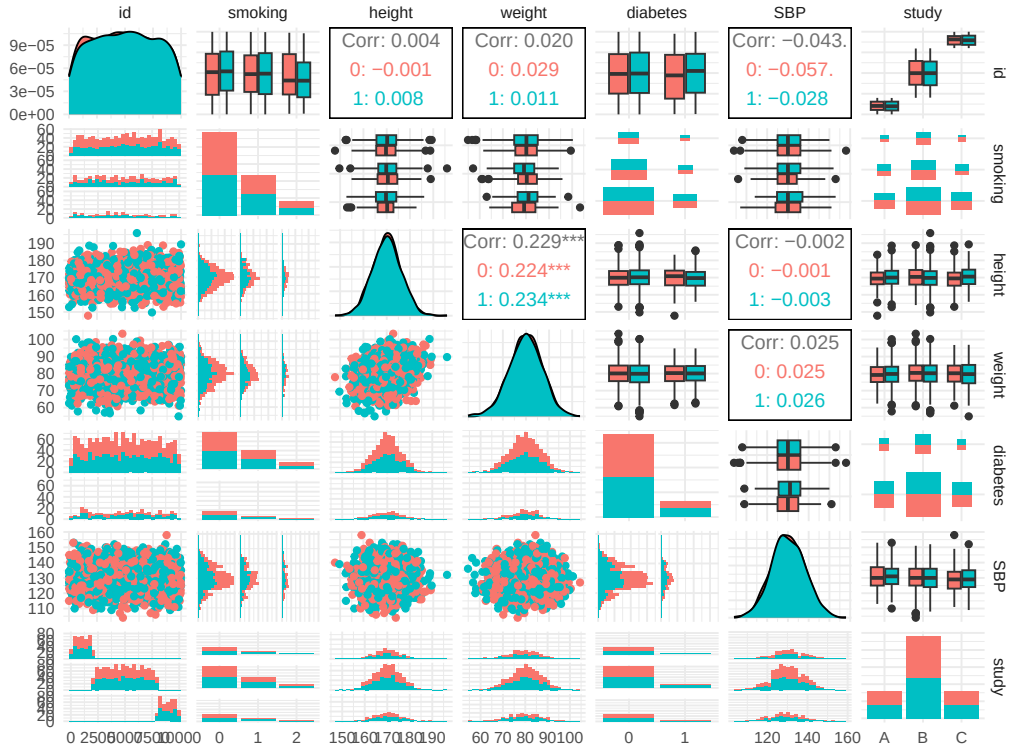

Figure 1: Scatterplot Matrix of Continuous

B Appendix: Code

The code used in this project are in the P8106_Code.pdf and P8106_Code.rmd files. Including data processing methods, modeling methods, tuning parameters selection, model comparison, etc. See Appendix

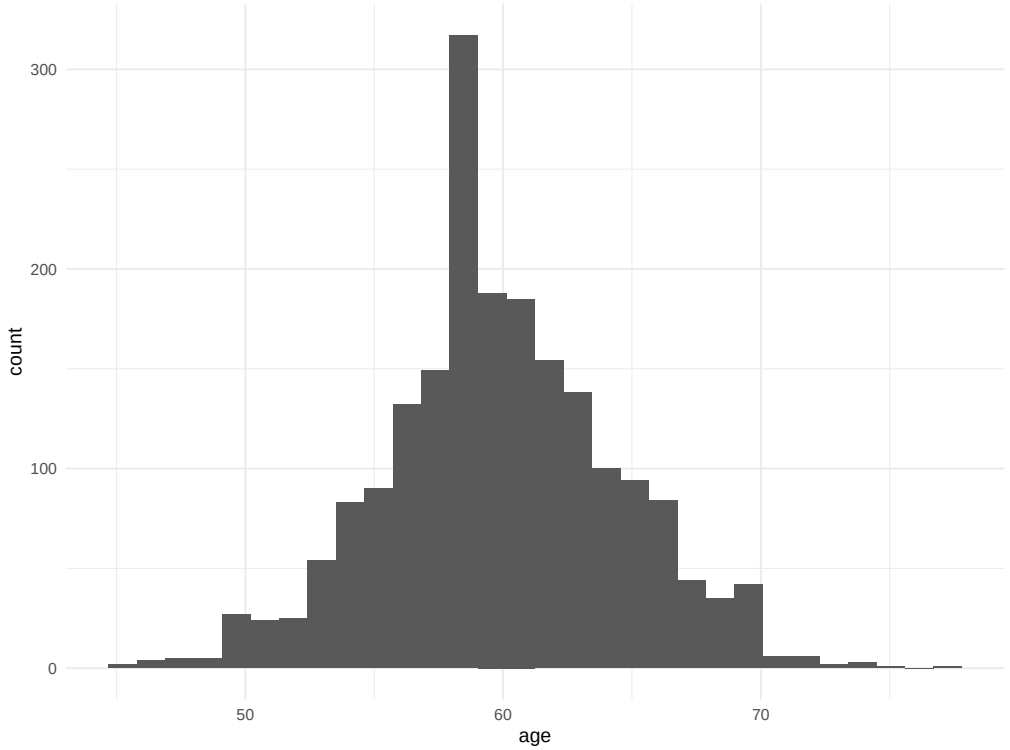

Figure 2: Density of age

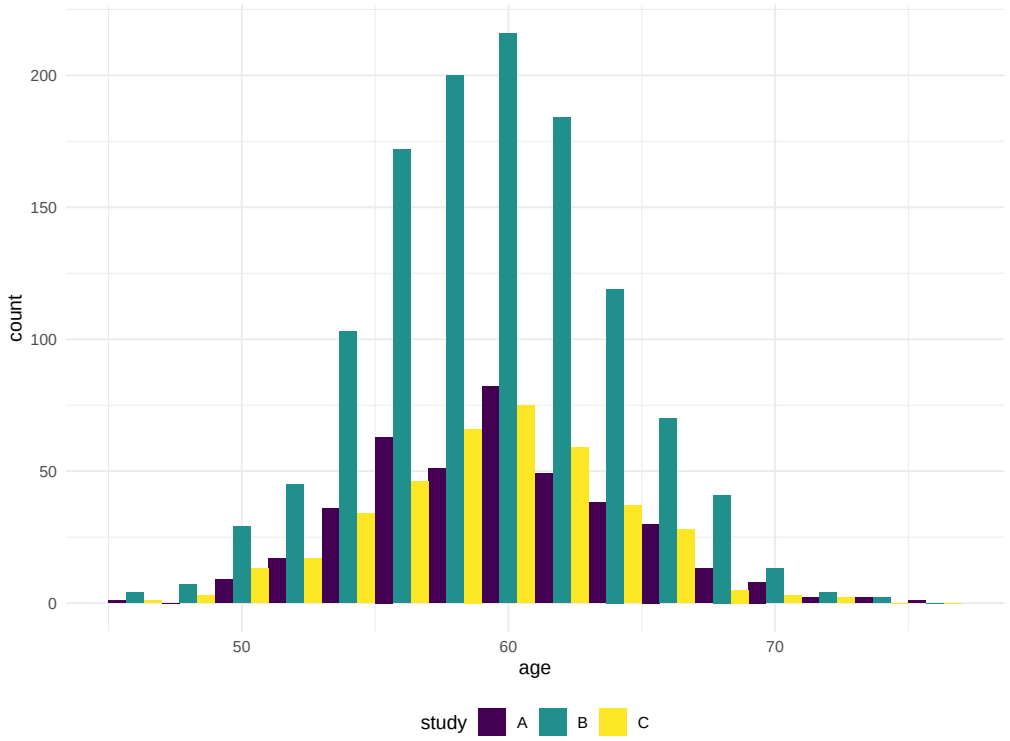

Figure 3: Distribution of age by studies

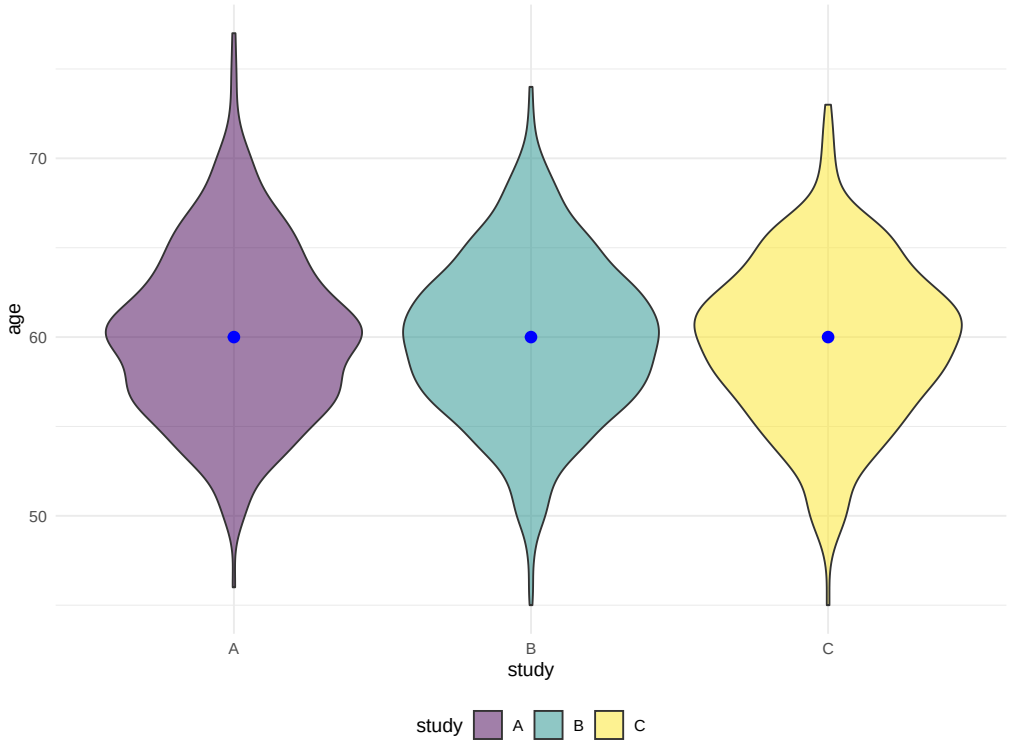

Figure 4: Volin plot of age by studies

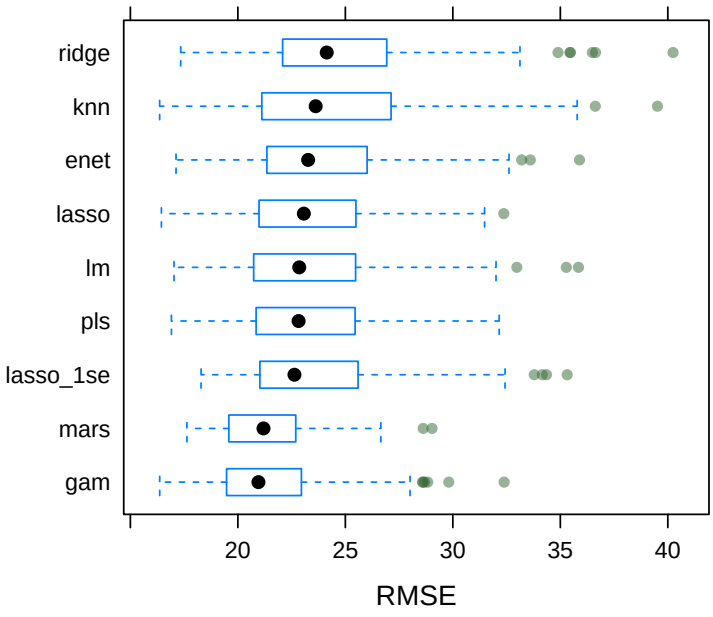

Figure 5: Model Comparison Boxplot

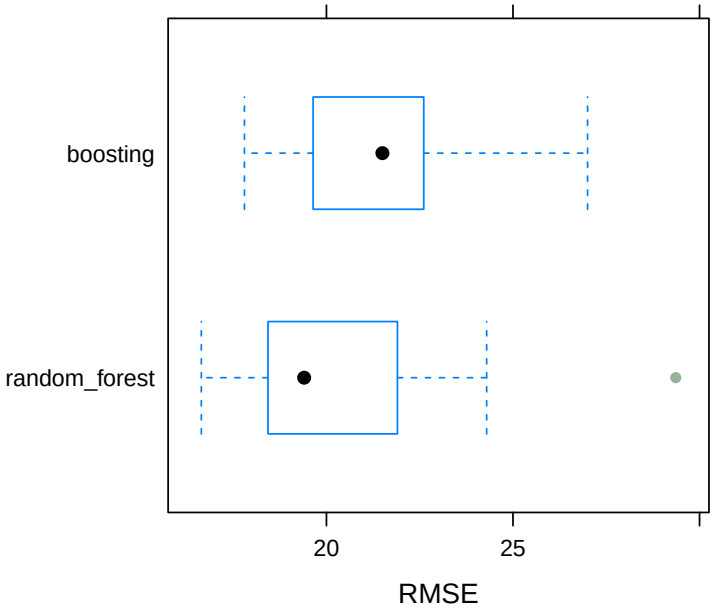

Figure 6: Model Comparison Boxplot 2

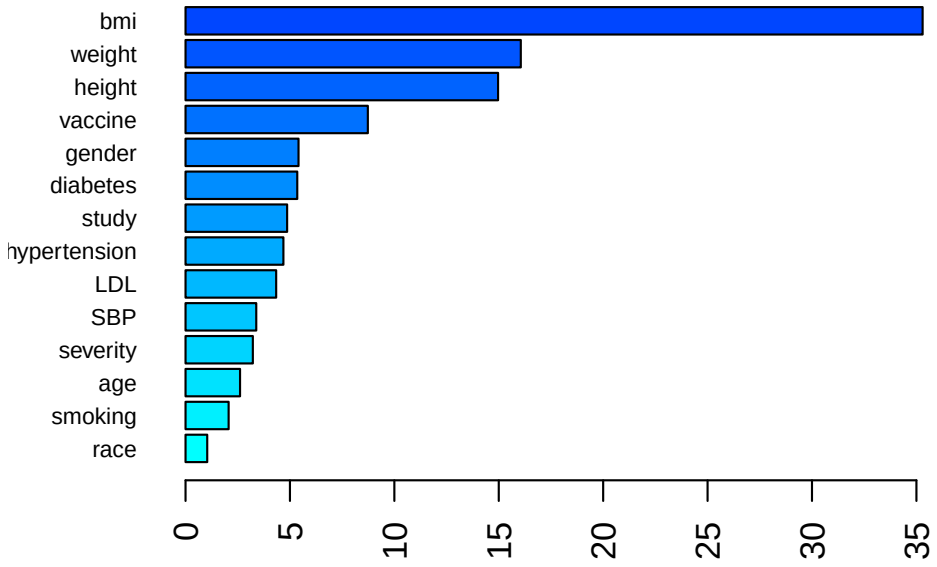

Figure 7: Scatterplot Variable importance