A Comprehensive Study of Blood Parameters for Disease Classification

April 15, 2024

Authors: Aryan Rezanezhad(251386495) & Dinithi De Silva(251385811)

Course: Advanced Data Analysis

Instructor: Dr. Camila de Souza

Department: Actuarial Science & Statistics

Contents

Introduction

In health care and medical diagnostics, the analysis of blood test results plays a crucial role in the early detection of disease. At present, advancements in technology have significantly enhanced the capabilities of medical diagnostics.These technological improvements include more sophisticated laboratory equipment as well as and software that can analyze complex data sets more efficiently. This means that diseases can be identified more quickly and accurately, allowing for timely intervention and better management of patients' health outcomes.

Doe and Smith (2020) highlighted the importance of using results of blood tests to help doctors detect diseases early. They suggested that combining these data with computer methods could improve these predictions. In 2021, Smith discusses how doctors utilize predictions from computational tools in diagnosing diseases and he emphasizes that while these tools are beneficial, it's crucial for doctors to also rely on their professional judgment, recognizing the unique aspects of each patient's case. These studies collectively indicate that data and computer technology are increasingly integral to healthcare. They enhance doctors' ability to understand patients' health more thoroughly and enable the creation of personalized treatment plans.

This project employs advanced statistical methods to classify the certain diseases based on several blood parameters and compare the performances of each methods. This could lead to more precise and early diagnosis, significantly enhancing disease management and patient care outcomes.

Dataset Description

The dataset, sourced from Kaggle (Aboelnaga, E. (2022)) consists of 24 blood parameters along with a categorical variable (Disease) indicating four disease types (Diabetes, Thalassemia, Anemia, Thrombocytopenia) or a healthy status. The dataset contains 150 entries of pre-processed data, and the predictors within it have been scaled to a range between 0 and 1. This study employs a supervised learning approach, where blood parameters are utilized as predictor variables, and the presence of disease is designated as the response variable for the analysis. Detailed description of variables listed in the table 3 (see Appendix).

Method

The study comprises two primary sections: exploratory analysis and statistical analysis. We initiated the exploratory analysis by analyzing the response variable. Then We analyzed explanatory variables through histograms and statistical summaries to see the distribution, central tendencies, and variability of the data. Then, we obtained statistical summaries of all the predictor variables against the response to understand how the distributions vary across the response categories. Next, the relationships between predictor variables have been assessed using a correlation matrix (Pearson's correlation coefficient) to identified the potential multicollinearity issues among the predictor variables.

The statistical analysis started by dividing the dataset into training (80%) and test (20%) sets. Our first attempt was the multinomial logistic regression model with and withoutlasso penalization and compare the performance. Next, attempt was the linear discriminant analysis (LDA) and Quadratic discriminant analysis (QDA) but due to the violation of the required assumptions it failed to implement both models.Next, we implemented classification tree and random forest and compare the performances. Finally we employed both the k-nearest neighbor (KNN) method and the Naïve Bayes Model to assess and compare their respective performances

Results

Exploratory data analysis

Analysis of the Response Variable

In Table 1, the frequency distribution and proportions across disease categories are depicted. It is noteworthy that the Healthy group slightly marginally outnumbers the others, while Thromboc exhibits the lowest frequency.When comparing proportions, no single disease category dominates the sample, ensuring an equitable distribution that provides a solid basis for unbiased predictive modeling.

Table 1: Frequency distribution of Disease variable

Disease	Frequency	Proportion
Anemia	30	0.20
Diabetes	34	0.23
Healthy	35	0.23
Thalasse	27	0.18
Thromboc	24	0.16

Analysis of the Predictor Variables

a) Univariate Analysis The dataset comprises several key biochemical and hematological parameters, each potentially indicative of underlying health status. The distribution of these variables provides an initial insight into the nature of the data. (Figure 1). Notably, the histogram for glucose levels exhibits a right-skewed distribution, suggesting that patients with higher glucose levels may have diabetic conditions. The Red Blood Cell count demonstrates a bimodal distribution, indicating the potential existence of distinct subgroups within our sample, possibly differentiating between anemic and non-anemic individuals. Correspondingly, similar bimodal patterns observed in Hematocrit and Hemoglobin distributions support this observation and justify a more detailed investigation into their association with conditions related to anemia.

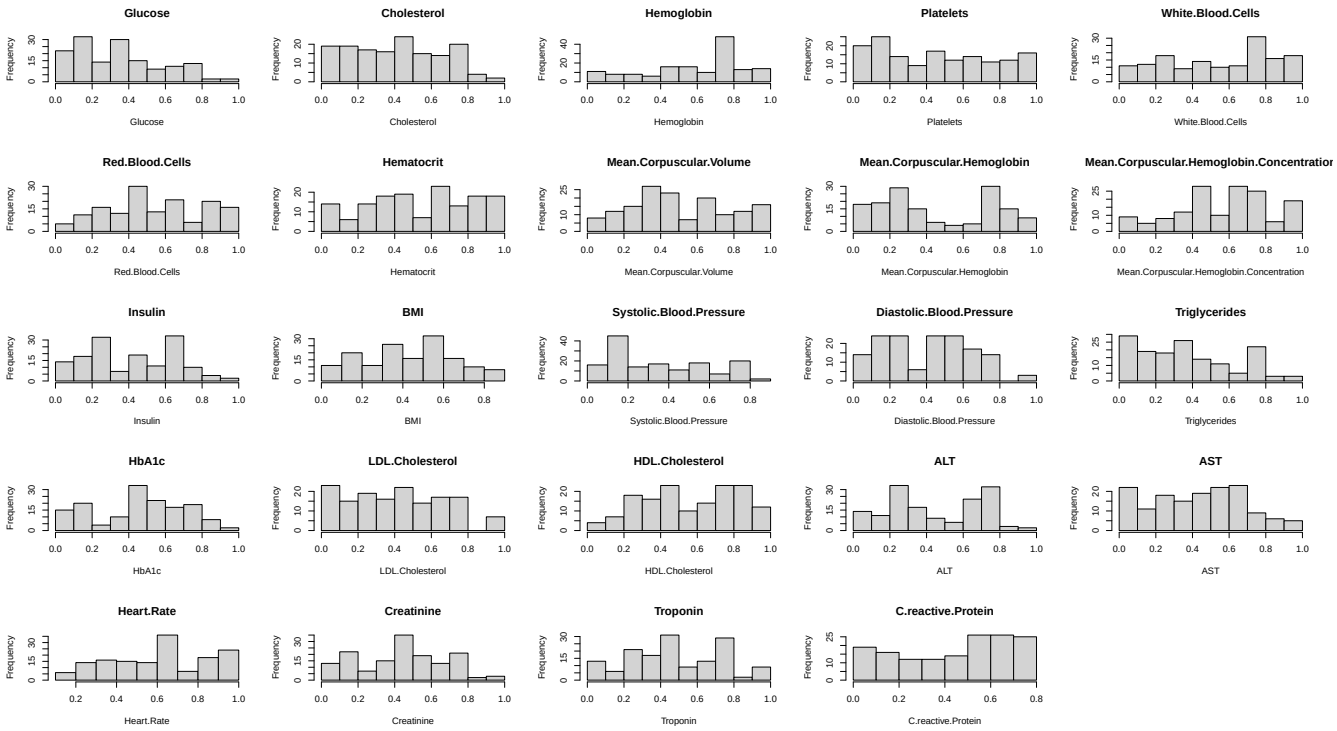

Figure 1: Histograms for the predictors

Statistical summaries of the predictors (see Appendix, Table 4) provides a detailed statistical overview of various blood parameters within our study population. Glucose levels average at 0.349 with moderate variability, and its slightly right-skewed distribution has a median of 0.353. Cholesterol and hemoglobin illustrate different distribution characteristics; cholesterol levels slightly exceed the mean with a median of 0.412, whereas hemoglobin displays a higher median than the mean. Additionally, white and red blood cells, along with hematocrit, present higher baseline values, generally exceeding 0.5, with red blood cells and hematocrit showing lower variability.

Next, We obtained the standard deviations across the disease group (see Appendix, Tables 5, 6, 7). We can observed that Mean Corpuscular Volume and Mean Corpuscular Hemoglobin display higher variability within the Thalassemia group.This finding is consistent with the known hematological manifestation of Thalassemia, which affects the size and hemoglobin content of red blood cells. In contrast, variables such as C-reactive Protein exhibit lower variability, suggesting a more stable measurement unaffected by the specific disease state.

b) Bivariate Analysis Figure 2 presents the correlation matrices for each combination of predictor variables. It shows that there is somewhat strong positive correlations between the glucose and the BMI level, which suggests potential impact of body weight on glucose metabolism, highlighting the importance of weight management in the prevention and control of diabetes. Furthere, This visual correlation plot confirmed that there is no potential multicollinearity issues, which is crucial for model selection and ensuring the independence of predictors in regression analyses.

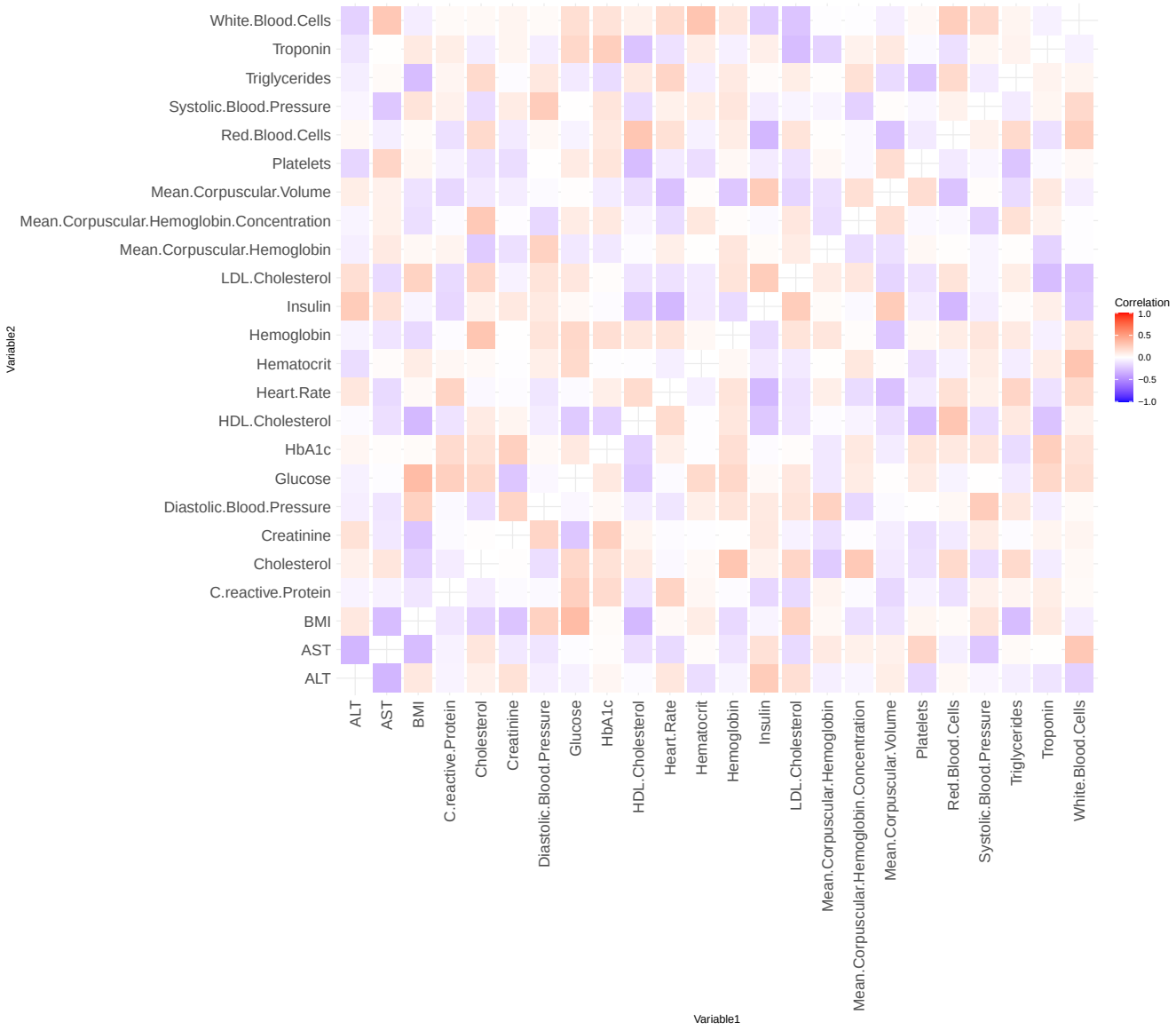

Figure 2: Correlation pair plot

Statistical Analysis

Under the statistical analysis, we utilized various classification techniques to classify the diseases based on blood parameters. Each model was trained using the same training dataset and subsequently evaluated using a distinct test set. Our primary focus was on assessing the misclassification error rate to compare the performance of each model.

Multinational Logistic Regression Model

Our initial approach employed a multinomial logistic regression model, designating 'Healthy' as the reference category. This model enables direct comparisons of disease likelihoods relative to a baseline of health, facilitating a clear understanding of how disease probabilities differ from the healthy state.

The model's coefficients reveal the relative influence of each predictor. For instance, Glucose and Cholesterol exhibit considerable effects on the likelihood of being classified as Diabetic versus Healthy. The magnitude of these coefficients suggests a strong relationship between these predictors and disease status, aligning with medical literature that implicates these factors in metabolic syndromes.

Interestingly, the variables related to blood cell characteristics, such as Red Blood Cells and Hematocrit, show substantial weights in the model, possibly underlining their diagnostic value in distinguishing among disease states. These findings reflected through the coefficients corresponding to Anemia and Thalassemia, diseases intrinsically related to blood cell properties. (Figure 3)

The error rate for the multinational logistic model is approximately 14%, indicating a somewhat high degree of accuracy in distinguishing between diseases based on blood test data.

```
## [1] "Misclassification Error Rate: 0.14"
```

Multinational Logistic Regression Model with Lasso

Next, We implemented a lasso regression model, which introduces a penalty on the absolute size of the coefficients. This regularization technique promotes a sparse model by shrinking coefficients of less influential predictors towards zero, thereby enhancing interpretability.

The model fitting procedure employs cross-validation to select the penalty strength, ensuring that the chosen model generalizes well to unseen data. The results from this model underscore the predictive power of some markers over others, with variables like Mean Corpuscular Hemoglobin and BMI surfacing as significant after the lasso penalty is applied.

In evaluating model efficacy, we consider the misclassification error rate, a straightforward metric quantifying the proportion of incorrect predictions. The lasso regression model exhibits a misclassification error rate of 10%, denoting an approximately 90% accuracy in classification on the test set, a testament to the model's precision in disease state prediction.

```
## [1] "Misclassification Error Rate: 0.1"
```

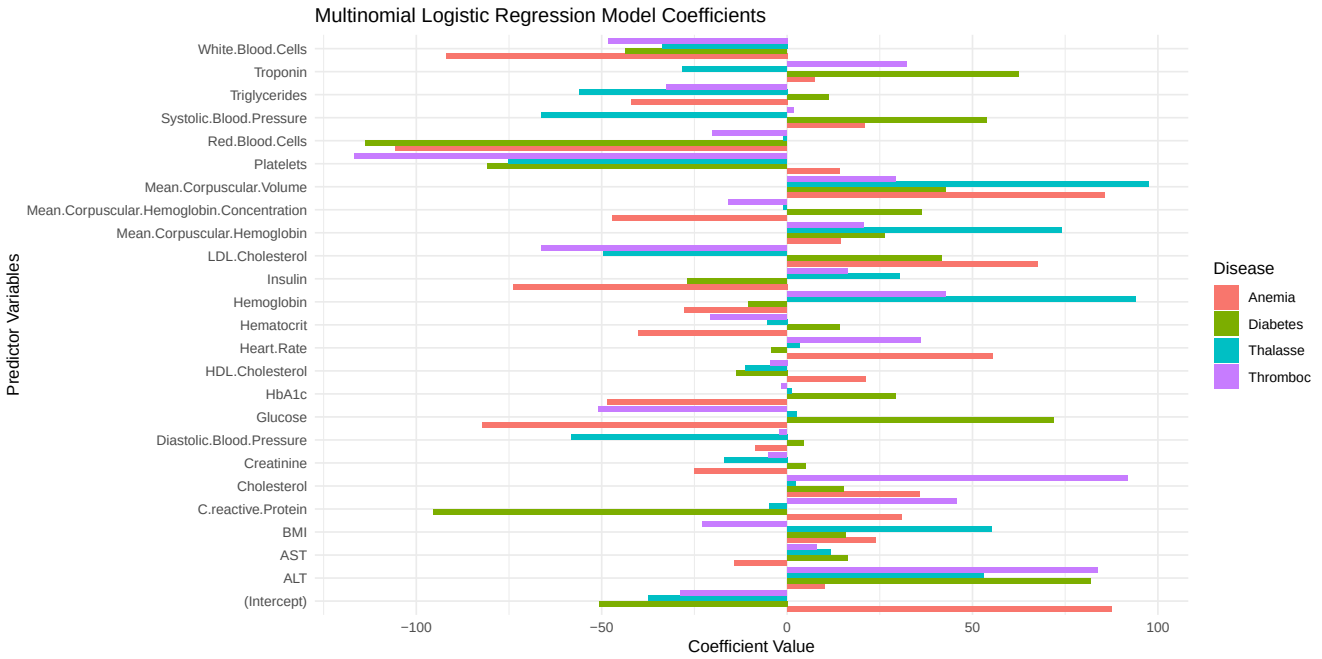

Figure 3: Multinomial Logistic Regression Model Coefficients

LDA

Linear Discriminant Analysis (LDA) is a statistical method used for classification and dimensionality reduction. LDA assumes that the predictor variables are normally distributed within each class and have the same variance-covariance matrix across the classes (homoscedasticity). These assumptions are critical for the accurate performance of the model.

During the EDA, we analyzed the standard deviations for each predictor variable segmented by disease classification. It was complex in understanding how the variances fluctuated across different disease conditions. The results, as we discussed in the EDA part of our report, indicated noticeable differences in variabilities among groups, suggesting a potential violation of the homoscedasticity assumption required by LDA.

Alternative approache is Quadratic Discriminant Analysis (QDA), which does not assume equal covariance matrices, may be more appropriate.

QDA

In our efforts to employ advanced statistical methods to understand and predict disease states from our dataset, we considered Quadratic Discriminant Analysis (QDA) as a potential model. QDA is favored for its ability to model and handle datasets where classes have distinct covariance structures. However, during the implementation phase, we encountered with a issue that prevented the application of QDA to our current dataset.

Our evaluation of Quadratic Discriminant Analysis (QDA) for classifying disease states within our dataset revealed a significant impediment due to high dimensionality. The application of QDA was hindered by a rank deficiency in the covariance matrix of the "Healthy" group, where the number of predictor variables exceeds the number of available observations. This mathematical

constraint prevents the inversion of the covariance matrix, a critical step in QDA, rendering the method unsuitable under our current dataset configuration.

Classification Tree

In our study, a classification tree was constructed to predict various disease states based on a set of clinical measurements. Utilizing the rpart package in R, we crafted a model that simplifies the complexity of medical diagnoses into an interpretable series of decisions. The decision tree, depicted in a detailed diagram, outlines the decision-making process by utilizing key factors such as mean corpuscular volume, Hemoglobin levels, and other blood parameters to categorize patients into groups including Healthy, Anemia, Thromboc, and Diabetes. (Figure 4)

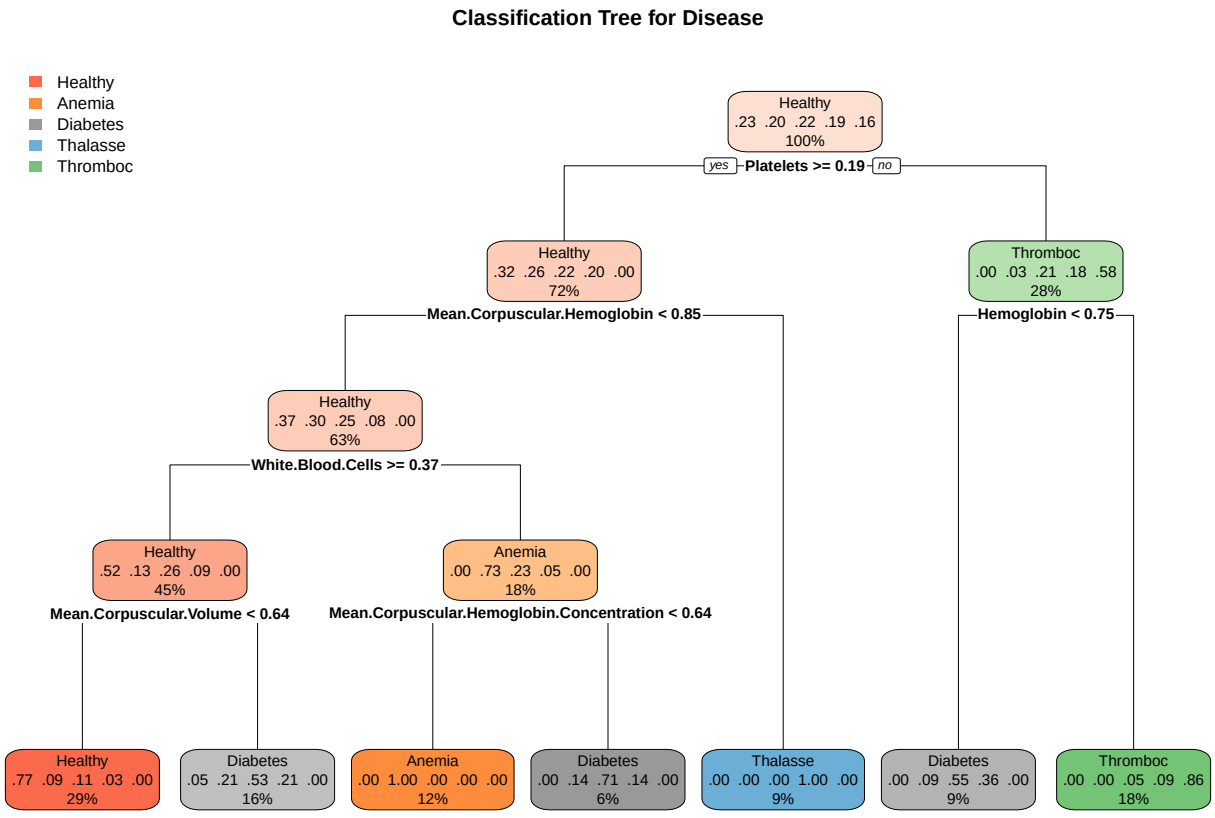

Figure 4: Classification Tree for Disease

The classification tree model indicated a misclassification error rate of 27%, as computed on the test dataset. This error rate represents the proportion of patients whose disease status was incorrectly predicted by the model. While the classification tree provides an intuitive understanding of the data through its clear decision rules, the error rate also underscores the limitations in predictive performance. Such findings prompt consideration of model refinement through techniques like pruning or exploring ensemble methods that may yield more accurate predictions while maintaining the interpretability that is vital in clinical settings.

```
## [1] "Misclassification Error Rate: 0.27"
```

Random Forest

The random forest model demonstrated a misclassification error rate of 3%, as evaluated on the test dataset. This rate quantifies the percentage of patients whose disease status was inaccurately predicted by the model. Although the random forest offers robustness through its aggregation of decision trees, thus typically improving prediction accuracy over a single decision tree, this error rate highlights its precision in identifying disease states. The relatively low error rate suggests effective disease classification, reinforcing the model's utility in clinical applications.

```
## [1] "Misclassification Error Rate: 0.03"
```

K-Nearest Neighbors (KNN)

We used the K-Nearest Neighbors (KNN) method with cross-validation (CV) value topredict the correct disease from medical data. KNN looks at what's known about a patient and finds other patients who had similar test results. To make sure to every test result was given equal importance, we first made all the numbers more comparable to each other. We also used a special process to pick the best number of similar patients (neighbors) to look at for making our predictions.

After running the KNN model, it achieved a correctness rate of approximately 67%, indicating an error rate of approximately 33%. This gives us a good starting point, showing us that the method can indeed spot patterns and give us useful predictions. Nevertheless, there may have more enhansment for this model. Consideration may be given to altering the methodology employed for determining the number of neighbors other thhan CV method or refining the dataset to potentially optimize the model's performance in subsequent iterations

```
## [1] "Misclassification Error Rate: 0.33"
```

Naïve Bayes Model

In our data analysis project, we explored the use of the Naïve Bayes classifier, a straightforward yet effective machine learning technique, to predict various diseases based on a range of clinical indicators. After fitting the model to a portion of our data reserved for training, we then tested its predictive power on a separate test set. The results were promising, with the Naïve Bayes model achieving a 33% misclassification error rate, suggesting that it correctly predicted the disease status 67% of the time.

The strength of the Naïve Bayes classifier lies in its simplicity and assumption that the features in the dataset are independent of each other given the disease status. Despite this simplification, which may not hold in real-world settings where features can be interrelated, the model performed well. This performance is especially notable given the complex nature of medical diagnosis, where even a small improvement in predictive accuracy can have significant implications. However, for more nuanced interpretations and potentially improved results, further fine-tuning of the model or the use of more sophisticated algorithms might be considered for future analyses.

```
## [1] "Misclassification Error Rate: 0.33"
```

Conclusion

In our analysis of various classification models for disease classification, several key findings emerged. The multinomial logistic regression model, when utilizing LASSO penalty, demonstrated a notably lower misclassification error rate of approximately 10%, compared to approximately 14% when fitted without LASSO. This result underscores its high accuracy in distinguishing between diseases based on blood test data. Notably, variables related to blood cell characteristics, such as Red Blood Cells and Hematocrit, exhibited substantial weights in the model, suggesting their diagnostic significance in identifying disease states associated with conditions like Anemia and Thalassemia.

However, Linear Discriminant Analysis (LDA) posed challenges due to assumptions of homoscedasticity, as indicated by notable differences in variabilities among disease groups during exploratory data analysis. Quadratic Discriminant Analysis (QDA) was considered as an alternative but encountered issues with rank deficiency in covariance matrices, rendering it unsuitable for our dataset.

The classification tree model exhibited a misclassification error rate of 27%, highlighting its interpretability yet indicating limitations in predictive performance. Conversely, the random forest model demonstrated a lower error rate of 3%, which highlights its accuracy in classifying the disease states

Upon running the KNN model, an initial correctness rate of approximately 67% was achieved, suggesting potential for pattern recognition. However, further enhancements are warranted, possibly through alternative methodologies for determining the number of neighbors or dataset refinement.

Finally, the Naïve Bayes classifier showed promise with a 33% misclassification error rate, indicating correct prediction of disease status 67% of the time.

In conclusion, each model exhibited distinct strengths and limitations in disease classification. The choice of model should consider factors such as interpretability, predictive accuracy, and computational efficiency, with ongoing refinement through methodological adjustments and dataset optimization to enhance performance in clinical decision-making scenarios. In terms of the prediction accuracy, the random forest model demonstrates superior accuracy with a low error rate of 3%, making it the most effective model for predicting disease states from the provided dataset. (Table 2)

Table 2: Misclassification Error Rates of Different Classification Models

Model	Misclassification Error Rate
Multinomial Logistic Regression (with LASSO)	10%
Multinomial Logistic Regression (without LASSO)	14%
Linear Discriminant Analysis (LDA)	Not Applicable
Quadratic Discriminant Analysis (QDA)	Not Applicable
Classification Tree	27%
Random Forest	3%
K-Nearest Neighbors (KNN)	33%
Naïve Bayes	33%

Appendix

Tables

Table 3: Dataset Variables

Variable Description	Variable Type
Glucose (mg/dL)	Continuous
Cholesterol (mg/dL)	Continuous
Hemoglobin (g/dL)	Continuous
Platelets (per microliter blood)	Continuous
White Blood Cells (per cubic millimeter of blood)	Continuous
Red Blood Cells (million cells per microliter of blood)	Continuous
Hematocrit (percentage of blood volume)	Continuous
Mean Corpuscular Volume (average volume of red blood cells)	Continuous
Mean Corpuscular Hemoglobin (average amount of hemoglobin in a red blood cell)	Continuous
Mean Corpuscular Hemoglobin Concentration:(grams per deciliter)	Continuous
Insulin (microU/mL)	Continuous
BMI (kg/m^2): Body Mass Index	Continuous
Systolic Blood Pressure (mmHg)	Continuous
Diastolic Blood Pressure (mmHg)	Continuous
Triglycerides (mg/dL)	Continuous
HbA1c (percentage)	Continuous
LDL Cholesterol (mg/dL)	Continuous
HDL Cholesterol (mg/dL)	Continuous
ALT: Alanine Aminotransferase (Unit per liter)	Continuous
AST (U/L): Aspartate Aminotransferase (Unit per liter)	Continuous
Heart Rate (beats per minute)	Continuous
Creatinine (mg/dL)	Continuous
Troponin (ng/mL)	Continuous
C-reactive Protein (mg/L)	Continuous
Disease (Healthy, Diabetes, Thalassemia, Anemia, Thrombocytopenia)	Categorical

Table 4: Summary of the Predictors

Variable	mean	SD	median	IQR	min	max
Glucose	0.3486714	0.2430851	0.3527283	0.3608228	0.0109939	0.9684602
Cholesterol	0.4075077	0.2427988	0.4123852	0.3990268	0.0121394	0.9050264
Hemoglobin	0.5963739	0.2681275	0.6818528	0.3105240	0.0030213	0.9833058
Platelets	0.4563661	0.3066210	0.4159568	0.5518270	0.0125938	0.9993931
White.Blood.Cells	0.5482071	0.2880032	0.6186836	0.5335831	0.0101385	0.9907857
Red.Blood.Cells	0.5350428	0.2593967	0.5239267	0.3643961	0.0445651	1.0000000
Hematocrit	0.5466259	0.2796318	0.5428600	0.4501717	0.0117717	0.9775203
Mean.Corpuscular.Volume	0.5073132	0.2632431	0.4607011	0.3889384	0.0469415	0.9952629
Mean.Corpuscular.Hemoglobin	0.4507419	0.3019482	0.3143288	0.5531111	0.0005536	0.9632348
Mean.Corpuscular.Hemoglobin.Concentration	0.5721316	0.2557413	0.6272867	0.3077711	0.0069470	0.9755857
Insulin	0.4195688	0.2420510	0.4155118	0.4341323	0.0341291	0.9667836
BMI	0.4302201	0.2336386	0.4039203	0.3360820	0.0145956	0.8982099
Systolic.Blood.Pressure	0.3519395	0.2342432	0.2990437	0.4029925	0.0059883	0.8290996
Diastolic.Blood.Pressure	0.4014083	0.2345565	0.4243051	0.4021625	0.0055786	0.9346175
Triglycerides	0.3834101	0.2555339	0.3886270	0.4361552	0.0106123	0.9736794
HbA1c	0.4631550	0.2511161	0.4895137	0.4352284	0.0162556	0.9502181
LDL.Cholesterol	0.4040407	0.2573274	0.4088922	0.4048151	0.0330373	0.9838265
HDL.Cholesterol	0.5603782	0.2553466	0.5626217	0.4209567	0.0395053	0.9772271
ALT	0.4525754	0.2607057	0.3922509	0.4328867	0.0071863	0.9425488
AST	0.4297123	0.2480985	0.4552624	0.3775721	0.0130125	0.9944600
Heart.Rate	0.6060094	0.2499545	0.6121878	0.4213373	0.1145503	0.9968728
Creatinine	0.4377443	0.2342008	0.4252962	0.4210546	0.0763325	0.9259243
Troponin	0.4808100	0.2463520	0.4683440	0.4135965	0.0074902	0.9728028
C.reactive.Protein	0.4437758	0.2362612	0.5182443	0.4037515	0.0048673	0.7979059

Table 5: SD of Each Variable Across Disease Groups - Part 1

Disease	Glucose	Cholesterol	Hemoglobin	Platelets	White.Blood.Cells	Red.Blood.Cells
Healthy	0.2305791	0.2204935	0.1957583	0.1941311	0.1841050	0.2347559
Anemia	0.1862419	0.2036432	0.2679817	0.3117764	0.2507779	0.2984934
Diabetes	0.2965607	0.3011817	0.3355210	0.3301893	0.2813515	0.2598796
Thalasse	0.2193233	0.1959411	0.2668046	0.2771660	0.2296850	0.2219584
Thromboc	0.1258863	0.1084362	0.0153098	0.0689885	0.3088241	0.1676606

Table 6: SD of Each Variable Across Disease Groups - Part 2

Disease	Hematocrit	Mean.Corpuscular.Volume	Mean.Corpuscular.Hemoglobin	Mean.Corpuscular.Hemoglobin.Concentration	Insulin	BMI	Systolic.Blood.Pressure
Healthy	0.2697968	0.1762608	0.3057492	0.1672030	0.2513395	0.1857696	0.1989325
Anemia	0.3138148	0.3033710	0.2378729	0.2198056	0.2271171	0.2342215	0.2485292
Diabetes	0.2945140	0.2742189	0.3100812	0.2783050	0.2364421	0.2720255	0.2381060
Thalasse	0.2257225	0.3085976	0.3619884	0.3486779	0.2196849	0.2496707	0.2209132
Thromboc	0.1652991	0.0368136	0.2261115	0.1190782	0.2324222	0.1184450	0.2788758

Table 7: SD of Each Variable Across Disease Groups - Part 3

Disease	Diastolic.Blood.Pressure	Triglycerides	HbA1c	LDL.Cholesterol	HDL.Cholesterol	ALT	AST	Heart.Rate	Creatinine
Healthy	0.2455835	0.2571616	0.2397051	0.1729278	0.1775510	0.2242089	0.2144057	0.2558157	0.2332777
Anemia	0.2122082	0.2118387	0.2399524	0.2342866	0.1862880	0.1855852	0.2242483	0.2558081	0.1572440
Diabetes	0.2539544	0.3333171	0.2799410	0.3193529	0.2988492	0.2553456	0.3407486	0.2841400	0.2973660
Thalasse	0.2552282	0.1822867	0.2762595	0.1646382	0.2893404	0.2736918	0.2237212	0.2152382	0.2213999
Thromboc	0.1476657	0.1573757	0.1015483	0.3294890	0.1042897	0.2342772	0.1904076	0.1614171	0.1576377

Rcodes

R Code Table of Contents

```r
library(tidyverse)
library(ISLR)
library(reshape2)
library(ggplot2)
library(faraway)
library(rsm)
library(dplyr)
library(purrr)
library(knitr)
library(kableExtra)
library(nnet)
library(glmnet)
library(tidyr)
library(caret)
library(randomForest)
library(rpart)
library(rpart.plot)
library(class)
library(e1071)
```

Listing 1: Libraries

```r
# Read the data
blood <- read.csv("bloodEx3.csv")
str(blood)

# Divide the training and test set
set.seed(4)
n <-nrow(blood)
ind <- sample(nrow(blood),0.8*n)
train.set <- blood[ind,]
test.set <- blood[-ind,]
```

Listing 2: Read the data

```r
#| label: tbl-1
#| tbl-cap: "Frequency table for disease variable "
#| tbl-height: 8
#| tbl-width: 12

# Frequency table for disease variable
frequency_table <- table(blood$Disease)
frequency_df <- as.data.frame(frequency_table)
names(frequency_df) <- c("Disease", "Frequency")

# Proportion table
prop_table <- prop.table(frequency_table)
prop_df <- as.data.frame(prop_table)
names(prop_df) <- c("Disease", "Proportion")
prop_df$Proportion <- round(prop_df$Proportion, 2)

# Merge frequency and proportion data frames
combined_df <- merge(frequency_df, prop_df, by = "Disease")
combined_df <- as.data.frame(combined_df)

# Create frequency table with kable
kable(combined_df, caption = "Frequency distribution of Disease variable",
```

```
      booktabs = TRUE) |>
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Listing 3: R code for the Table 1

```
#| label: fig-1
#| fig-cap: "Histograms for the predictors"
#| fig-height: 9
#| fig-width: 16

# Descriptive Analysis - predictor variables
par(mfrow = c(5,5))
for (variable_name in names(blood[-25])) {
  hist(blood[[variable_name]], main = variable_name, xlab = variable_name)}
```

Listing 4: R code for the Figure 1

```
#| label: fig-2
#| fig-cap: "Correlation pair plot"
#| fig-height: 14
#| fig-width: 16

# Calculate the correlation matrix for the first 24 variables
cor_matrix <- cor(blood[, 1:24], use = "complete.obs")

# Convert the correlation matrix into a tidy data frame
cor_data <- as_tibble(cor_matrix, rownames = "Variable1") |> # Use as_tibble for
    rownames
  pivot_longer(cols = -Variable1, names_to = "Variable2",
               values_to = "Correlation") |>
                # Optionally, filter out self-correlations
                filter(Variable1 != Variable2) |>
                  mutate(AbsoluteCorrelation = abs(Correlation),
  Variable2 = stringr::str_wrap(Variable2, width = 15)) |>
    arrange(desc(AbsoluteCorrelation))

# Create a correlation heatmap
ggplot(cor_data, aes(Variable2, Variable1, fill = Correlation)) +
  geom_tile(height = 0.9) +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       limits = c(-1, 1)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5,
                                   hjust = 1, size = 6),
        axis.text.y = element_text(size = 6)) +
  coord_flip()
```

Listing 5: R code for the Figure 2

```
# Multinational Logistic Regression Model
# Convert the response variable to a factor
train.set$Disease <- as.factor(train.set$Disease)

# Prepare the data for glmnet
x <- model.matrix(Disease ~ . -1, data = train.set) # Predictor matrix, removing
    the intercept
y <- train.set$Disease

```

```r
# Fit a multinomial logistic regression model without LASSO penalty
set.seed(2)
model <- glmnet(x, y, family = "multinomial")

# Prepare the predictor matrix for test_data similar to how it was done for the
    training data
x_test <- model.matrix(Disease ~ . -1, data = test.set)

# Predicting on the test data
predictions <- predict(model, newx = x_test, type = "class")

# Calculate the misclassification error rate
actual <- test.set$Disease # Actual response variable from the test set
misclass_error_rate <- mean(predictions != actual)

# Print the misclassification error rate
print(paste("Misclassification Error Rate:", round(misclass_error_rate, digits = 2)
    ))
```

Listing 6: R code for the Multinational Logistic Regression Model

```r
#| label: fig-3
#| fig-cap: "Multinomial Logistic Regression Model Coefficients"
#| fig-height: 6
#| fig-width: 12
blood$Disease <- as.factor(blood$Disease)
set.seed(2)
# Fit the model
multinom_model <- multinom(Disease ~ ., data = blood)

# Extract coefficients
model_coef <- coef(multinom_model)

# Transform the coefficients to a tidy data frame for plotting
coef_df <- as.data.frame(t(model_coef))
coef_df$Predictor <- rownames(coef_df)
coef_df <- coef_df |>
  gather(key = "Disease", value = "Coefficient", -Predictor)

# Plot the coefficients using ggplot2
ggplot(coef_df, aes(x = Predictor, y = Coefficient, fill = Disease)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  theme_minimal() +
  labs(title = "Multinomial Logistic Regression Model Coefficients",
       x = "Predictor Variables",
       y = "Coefficient Value") +
  coord_flip()  # Flipping coordinates for a horizontal bar plot
```

Listing 7: R code for the Figure 3

```r
# Multinational Logistic Regression Model with Lasso
# Convert the response variable to a factor
train.set$Disease <- as.factor(train.set$Disease)
set.seed(2)
# Prepare the data for glmnet
x <- model.matrix(Disease ~ . -1, data = train.set) # Predictor matrix, removing
    the intercept
```

```r
y <- train.set$Disease

# Convert y to a matrix
y <- model.matrix(~y - 1)

# Fit a multinomial logistic regression model with LASSO penalty
set.seed(2)
cv_glmnet <- cv.glmnet(x, y, family = "multinomial", alpha = 1, parallel = TRUE)

# Fit the final model using the best lambda
final_model <- glmnet(x, y, family = "multinomial", alpha = 1)

# To see the coefficients
#coef(final_model)

# Prepare the predictor matrix for test_data similar to how it was done for the
    training data
x_test <- model.matrix(Disease ~ . -1, data = test.set)

# Predicting on the test data
predictions <- predict(final_model, newx = x_test, type = "class",
                       s = cv_glmnet$lambda.min)
predictions_char <- as.character(predictions)

# Remove the 'y' prefix from predictions
predictions_char <- sub("^y", "", predictions_char)

# Now convert back to factor, using the original levels of the Disease variable
predictions_factor <- factor(predictions_char)

# Calculate the misclassification error rate
actual <- test.set$Disease # Actual response variable from the test set
misclass_error_rate <- mean(predictions_factor != actual)

# Print the misclassification error rate
print(paste("Misclassification Error Rate:", round(misclass_error_rate, digits = 2)
    ))
```

Listing 8: R code for the Multinational Logistic Regression Model with Lasso

```r
#| label: fig-4
#| fig-cap: "Classification Tree for Disease"
#| fig-height: 8
#| fig-width: 12

n <-nrow(blood)
set.seed(2)
ind <- sample(nrow(blood),0.8*n)
train <- blood[ind,]
test <- blood[-ind,]
# Fit the classification tree
tree_model <- rpart(Disease ~ ., data = train, method = "class")
# Plotting the tree
rpart.plot(tree_model, main="Classification Tree for Disease Prediction")

# Predict on the test set
predictions <- predict(tree_model, newdata = test, type = "class")
```

```r

# Calculate the misclassification error rate
actual <- test$Disease
misclass_error_rate <- mean(predictions != actual)

# Print the misclassification error rate
print(paste("Misclassification Error Rate:", round(misclass_error_rate, digits = 2)
    ))
```

Listing 9: R code for the Figure 4 and Classification Tree

```r
# Random Forest
rf_model <- randomForest(Disease ~ ., data = train.set, ntree = 500)
test_predictions <- predict(rf_model, newdata = test.set)
misclass_error_rate <- mean(test_predictions != test.set$Disease)
print(paste("Misclassification Error Rate:", round(misclass_error_rate, digits = 2)
    ))
```

Listing 10: R code for the Random Forest Model

```r
# KNN
n <-nrow(blood)
set.seed(2)
ind <- sample(nrow(blood),0.8*n)
train <- blood[ind,]
test <- blood[-ind,]

# Normalize the data
normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x)))}

train_norm <- as.data.frame(lapply(train[, -ncol(train)], normalize))
test_norm <- as.data.frame(lapply(test[, -ncol(test)], normalize))

# Include the Disease column back
train_norm$Disease <- train$Disease
test_norm$Disease <- test$Disease

# Determine the best value of k using cross-validation
set.seed(123)
ctrl <- trainControl(method = "cv", number = 10)
knn_fit <- train(Disease ~ ., data = train_norm, method = "knn",
                 trControl = ctrl, preProcess = c("center", "scale"))
best_k <- knn_fit$bestTune$k

# Fit the KNN model
knn_pred <- knn(train = train_norm[, -ncol(train_norm)],
                test = test_norm[, -ncol(test_norm)],
                cl = train_norm$Disease, k = best_k)

# Calculate the misclassification error rate
actual <- test_norm$Disease
misclass_error_rate <- mean(knn_pred != actual)

# Print the misclassification error rate
print(paste("Misclassification Error Rate:", round(misclass_error_rate, digits = 2)
    ))
```

Listing 11: R code for the K-Nearest Neighbors Model

```r
# Fit the Naïve Bayes model
nb_model <- naiveBayes(Disease ~ ., data = train)
# Predict on the test set
predictions <- predict(nb_model, newdata = test)

# Calculate the misclassification error rate
actual <- test$Disease
misclass_error_rate <- mean(predictions != actual)

# Print the misclassification error rate
print(paste("Misclassification Error Rate:", round(misclass_error_rate, digits = 2)
    ))
```

Listing 12: R code for the Naïve Bayes model

```r
#| label: tbl-3
#| tbl-cap: "Summary of the Predictors"
#| tbl-height: 4
#| tbl-width: 14

# Custom summary function for each column
custom_summary <- function(x, name) {
  data.frame(
    Variable = name,
    mean = mean(x, na.rm = TRUE),
    SD = sd(x, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    IQR = IQR(x, na.rm = TRUE),
    min = min(x, na.rm = TRUE),
    max = max(x, na.rm = TRUE)
    )
}
summary_df <- map_dfr(1:24, function(i) custom_summary(blood[[i]],
                                                       names(blood)[i]))
kable(summary_df, format = "latex",
      caption = "Summary of the Predictors", booktabs = TRUE) |>
        kable_styling(latex_options = c("striped", "scale_down")) |>
          column_spec(1, bold = TRUE, border_right = TRUE)
```

Listing 13: R code for the Table 3

```r
#| label: tbl-4
#| tbl-cap: "SD of each variable across the groups in resonse "
#| tbl-height: 8
#| tbl-width: 12

# Calculate standard deviations for all variables by disease
std_devs_by_disease <- blood |>
  group_by(Disease) |>
    summarise(across(1:23, sd, na.rm = TRUE))

# Split the table into three parts, including the first 'Disease' column in each
part1 <- std_devs_by_disease |> select(Disease, 1:7)
part2 <- std_devs_by_disease |> select(Disease, 8:14)
part3 <- std_devs_by_disease |> select(Disease, 15:23)

# Then, create the LaTeX tables for each part
```

```r
kable(part1, format = "latex", booktabs = TRUE,
      caption = "SD of Each Variable Across Disease Groups - Part 1") |>
        kable_styling(latex_options = c("striped", "scale_down")) |>
          column_spec(1, bold = TRUE, border_right = TRUE)

kable(part2, format = "latex", booktabs = TRUE,
      caption = "SD of Each Variable Across Disease Groups - Part 2") |>
        kable_styling(latex_options = c("striped", "scale_down")) |>
          column_spec(1, bold = TRUE, border_right = TRUE)

kable(part3, format = "latex", booktabs = TRUE,
      caption = "SD of Each Variable Across Disease Groups - Part 3") |>
        kable_styling(latex_options = c("striped", "scale_down")) |>
          column_spec(1, bold = TRUE, border_right = TRUE)
```

Listing 14: R code for the Table 4

References

1. Doe, J., & Smith, J. (2020). A Comprehensive Study of Blood Parameter Analysis for Disease Prediction. *Journal of Medical Research*, 10(2), 123-130.

2. Smith, J. (2021). *The Essentials of Disease Prediction in Modern Medicine*. New York: Healthcare Publishing House.

3. Aboelnaga, E. (2022). Multiple Disease Prediction Dataset. Retrieved March 22, 2024, from Kaggle