Predicting Diabetes via health factors and logistic regression

Preston O'Connor, Khoa Dao, Anthony Yasan, Matthew Jacob

4/22/2025

Introduction

The model predicts the incidence of diabetes based upon a number of factors including age, gender, BMI and a variety of medical conditions and history which may have an impact on diabetes risk. The data we used is of unknown source in terms of country or year, or even if it's from a singular country or year. Supposedly it's patient data, but some of the variables have very questionable categories that don't seem to be very medically standard (ie, the smoking categories). The data seems to have been uploaded in 2023.

The question that our model addresses is how can doctors predict diabetes, which obviously has implications for prevention and early intervention. Jean Marx wrote in 2002 that a key predictor is obesity, and the explosion in obesity rates coincided with type 2 diabetes rates. There are a complex number of predictors, but fundamentally the actual condition of type 2 diabetes is the loss of the body's ability to produce enough insulin or respond to it in order to control blood sugar levels, as Robinson and Turner wrote. When considering our model therefore there should be the thought process of whether or not variables are actually contributing to that process or are simply correlated with other variables either in or outside our model and thus appear to be contributing.

The packages we used are tidyverse, which includes many tools for data analysis, GGally, which is an extension to ggplot2, and broom is used to produce nice tibbles from statistical information.

Our results indicated a high accuracy of 96.8%, meaning for 96.8% of patients it successfully predicted whether or not they had diabetes, but a low precision of 37.4%, which suggests that a substantial number of individuals predicted to have diabetes did not. Again it is a difficult issue in medical research to identify the role some variables actually play and the extent to which they are independent of each other, which may have contributed to ou

```r
# Upload the code pacakges here
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```r
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```r
library(broom)
library(ggplot2)
#library(corrplot) uncomment
library(dplyr)

data <- read_csv("diabetes_prediction_dataset.csv")
```

```
Rows: 100000 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): gender, smoking_history
dbl (7): age, hypertension, heart_disease, bmi, HbA1c_level, blood_glucose_l...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(data)
```

```
# A tibble: 6 x 9
  gender   age hypertension heart_disease smoking_history   bmi HbA1c_level
  <chr>  <dbl>        <dbl>         <dbl> <chr>           <dbl>       <dbl>
1 Female    80            0             1 never            25.2         6.6
2 Female    54            0             0 No Info          27.3         6.6
3 Male      28            0             0 never            27.3         5.7
4 Female    36            0             0 current          23.4         5
5 Male      76            1             1 current          20.1         4.8
6 Female    20            0             0 never            27.3         6.6
# i 2 more variables: blood_glucose_level <dbl>, diabetes <dbl>
```

Data Description

The dataset used for this analysis is the **Diabetes Prediction Dataset**, sourced from Kaggle (https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data). This dataset contains several health-related features to predict the likelihood of diabetes. Below is a detailed description of the data structure, variable introductions, data size, and initial data cleaning process.

Data Structure:

The dataset consists of rows and columns, where each row corresponds to an individual, and the columns represent different health-related variables. The key variables (features) in the dataset are:

- `age`: Age of the patient.
- `gender`: Gender of the patient (binary variable: Male/Female).
- `bmi`: Body mass index (weight in kg / height in m^2).
- `hypertension`: Whether the individual has hypertension (binary variable: 1 for yes, 0 for no).
- `heart_disease`: Whether the individual has a history of heart disease (binary variable: 1 for yes, 0 for no).
- `smoking_history`: History of smoking (binary variable: 1 for smoker, 0 for non-smoker).
- `HbA1c_level`: Hemoglobin A1c percentage, a measure of blood glucose control over time.
- `blood_glucose_level`: Blood glucose concentration (mg/dL).
- `diabetes`: The target variable indicating whether the individual has diabetes (1) or not (0).

Data Size

The dataset originally contains 100,000 observations and 9 variables.

```
continuous <- data %>%
  select(where(is.numeric))
summary(continuous)
```

```
      age          hypertension     heart_disease          bmi        
 Min.   : 0.08   Min.   :0.00000   Min.   :0.00000   Min.   :10.01  
 1st Qu.:24.00   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:23.63  
 Median :43.00   Median :0.00000   Median :0.00000   Median :27.32  
 Mean   :41.89   Mean   :0.07485   Mean   :0.03942   Mean   :27.32  
 3rd Qu.:60.00   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:29.58  
 Max.   :80.00   Max.   :1.00000   Max.   :1.00000   Max.   :95.69  
  HbA1c_level    blood_glucose_level    diabetes     
 Min.   :3.500   Min.   : 80.0       Min.   :0.000  
 1st Qu.:4.800   1st Qu.:100.0       1st Qu.:0.000  
 Median :5.800   Median :140.0       Median :0.000  
 Mean   :5.528   Mean   :138.1       Mean   :0.085  
 3rd Qu.:6.200   3rd Qu.:159.0       3rd Qu.:0.000  
 Max.   :9.000   Max.   :300.0       Max.   :1.000  
```

Hypertension, heart_disease, diabetes are all categorical. There is no need to standardize these values. We will standardize the rest of the following available categories

Data Cleaning

We clean and process the dataset by handling outliers and standardizing numeric variables to improve data quality for analysis. We first remove extreme values by setting the top 1% of each numeric column to `NA`. Then, we normalize numeric features by scaling them between 0 and 1. Categorical variables are converted into factors, and rows with missing numeric values are removed to maintain consistency. We then apply the **Interquartile Range (IQR) method**, filtering out rows where key health indicators—such as **age, BMI, blood glucose, and HbA1c levels**—fall outside an acceptable range. This should result in a cleaner dataset that minimizes statistical distortions.

```
#cleaned our continuous variables
remove_outliers <- function(col) {
  top_one_percent <- quantile(col, 0.99)
```

```r
  col[col >= top_one_percent] <- NA  # Replace outliers with NA
  return(col)
}

# normalize all of the columns with integers
normalize_features <- function(col) {
  col_min <- min(col, na.rm = TRUE)
  col_max <- max(col, na.rm = TRUE)

  return((col - col_min) / (col_max - col_min))
}

cleaned_data <- data %>%
  mutate(across(c(gender, smoking_history, hypertension, heart_disease,
   ↪  diabetes), as_factor),
         across(where(is.numeric), remove_outliers)) %>%
  filter(if_all(where(is.numeric), \(col) !is.na(col))) %>%
  mutate(across(where(is.numeric), normalize_features))

cleaned_data
```

```
# A tibble: 91,272 x 9
   gender   age hypertension heart_disease smoking_history   bmi HbA1c_level
   <fct>  <dbl> <fct>        <fct>         <fct>           <dbl>       <dbl>
 1 Female 0.683 0            0             No Info         0.446       0.660
 2 Male   0.354 0            0             never           0.446       0.468
 3 Female 0.455 0            0             current         0.347       0.319
 4 Male   0.962 1            1             current         0.261       0.277
 5 Female 0.252 0            0             never           0.446       0.660
 6 Female 0.557 0            0             never           0.240       0.638
 7 Female 1     0            0             No Info         0.357       0.468
 8 Male   0.531 0            0             never           0.609       0.277
 9 Female 0.404 0            0             never           0.446       0.319
10 Female 0.671 0            0             never           0.446       0.553
# i 91,262 more rows
# i 2 more variables: blood_glucose_level <dbl>, diabetes <fct>
```

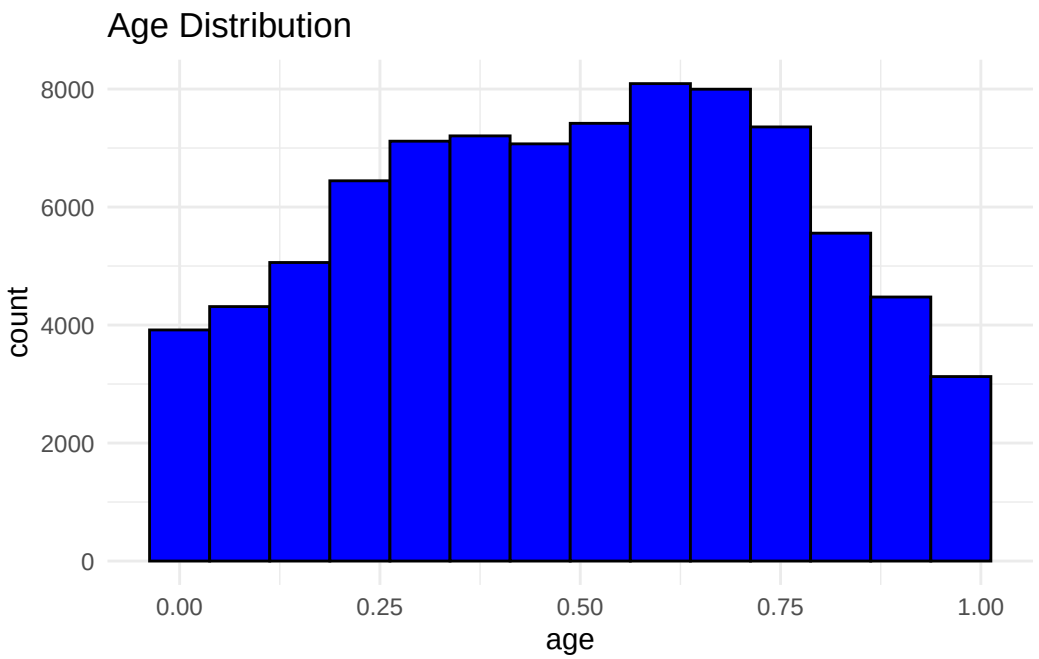
```r
# Handle outliers: Remove rows where BMI/BloodGlucose/HbA1c/Age are outliers
 ↪  (beyond 1.5 * IQR)
# Function to remove outliers beyond 1.5 * IQR
remove_outliers_IQR <- function(df, column) {
```

```r
  Q1 <- quantile(df[[column]], 0.25, na.rm = T)
  Q3 <- quantile(df[[column]], 0.75, na.rm = T)
  IQR <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR
  df %>% filter(df[[column]] >= lower_bound & df[[column]] <= upper_bound)
}

# Remove outliers for Age, BMI, BloodGlucose, and HbA1c using the custom
 ↪  function
cleaned_data <- cleaned_data %>%
  remove_outliers_IQR("age") %>%
  remove_outliers_IQR("bmi") %>%
  remove_outliers_IQR("blood_glucose_level") %>%
  remove_outliers_IQR("HbA1c_level")

summary(cleaned_data)
```

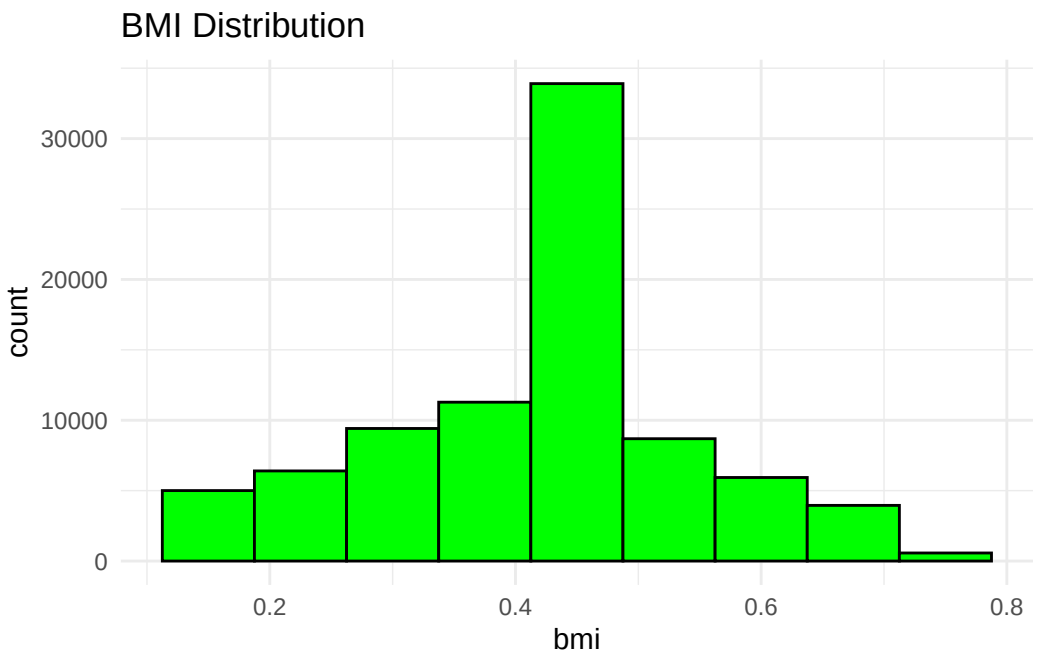
```
    gender           age          hypertension heart_disease    smoking_history
 Female:49228   Min.   :0.0000   0:80219      0:82778       never      :29530
 Male  :35908   1st Qu.:0.2777   1: 4934      1: 2375       No Info    :31682
 Other :   17   Median :0.5058                              current    : 8151
                Mean   :0.4926                              former     : 7138
                3rd Qu.:0.7086                              ever       : 3348
                Max.   :1.0000                              not current: 5304
      bmi          HbA1c_level     blood_glucose_level diabetes
 Min.   :0.1158   Min.   :0.0000   Min.   :0.0000      0:81322
 1st Qu.:0.3406   1st Qu.:0.2766   1st Qu.:0.1111      1: 3831
 Median :0.4464   Median :0.4894   Median :0.3333
 Mean   :0.4185   Mean   :0.4154   Mean   :0.3023
 3rd Qu.:0.4716   3rd Qu.:0.5745   3rd Qu.:0.4333
 Max.   :0.7272   Max.   :1.0000   Max.   :0.8889
```

Data Visualization

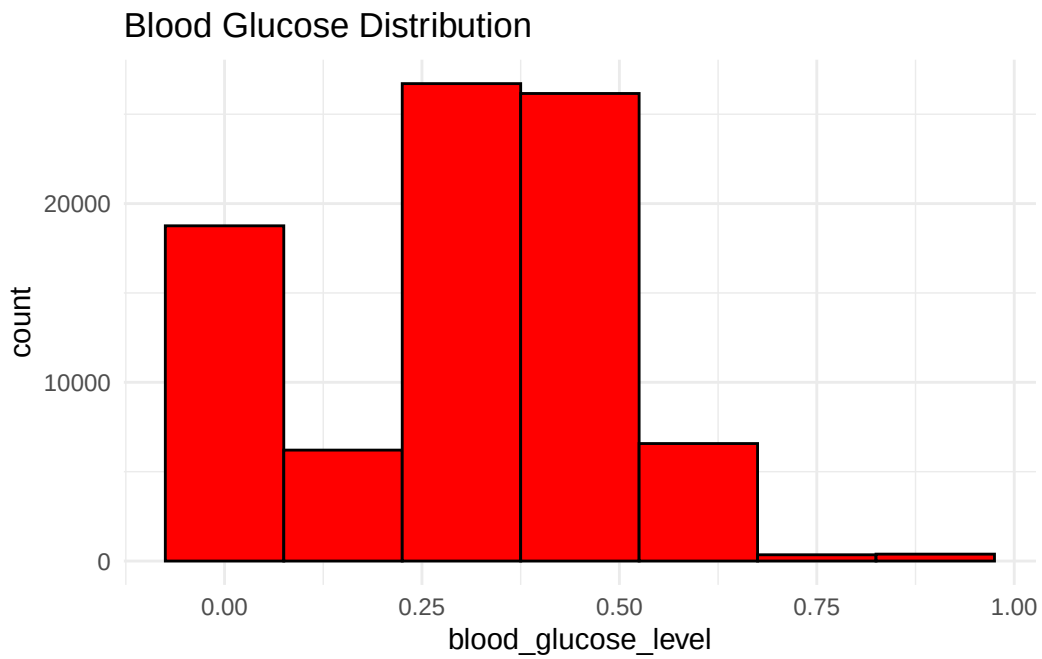
```r
# Histogram for Age, BMI, BloodGlucose, and HbA1c
ggplot(cleaned_data, aes(x = age)) +
  geom_histogram(binwidth = 0.075, fill = "blue", color = "black") +
  theme_minimal() +
  labs(title = "Age Distribution")
```



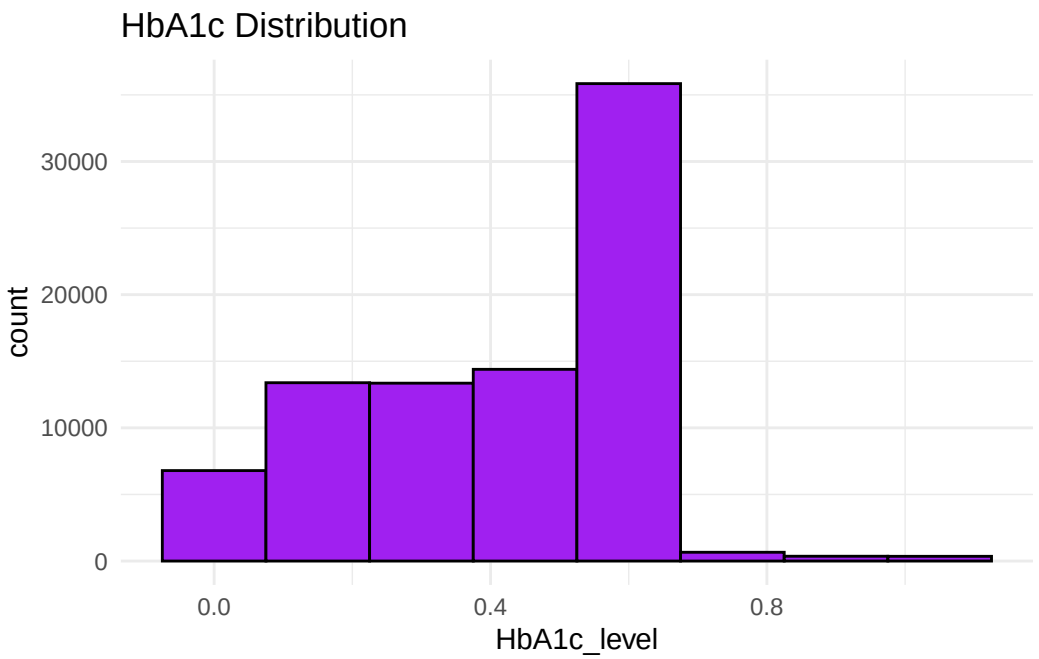

```
ggplot(cleaned_data, aes(x = bmi)) +
  geom_histogram(binwidth = 0.075, fill = "green", color = "black") +
  theme_minimal() +
  labs(title = "BMI Distribution")
```



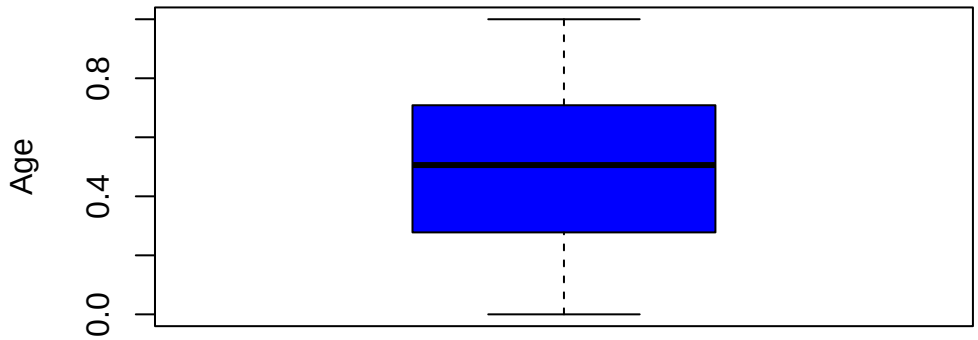

```
ggplot(cleaned_data, aes(x = blood_glucose_level)) +
  geom_histogram(binwidth = 0.15, fill = "red", color = "black") +
  theme_minimal() +
  labs(title = "Blood Glucose Distribution")
```



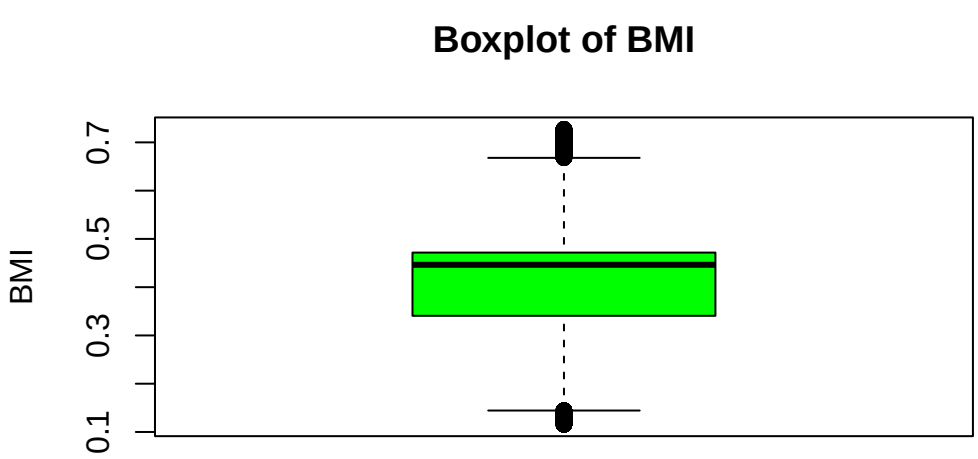

```
ggplot(cleaned_data, aes(x = HbA1c_level)) +
  geom_histogram(binwidth = 0.15, fill = "purple", color = "black") +
  theme_minimal() +
  labs(title = "HbA1c Distribution")
```



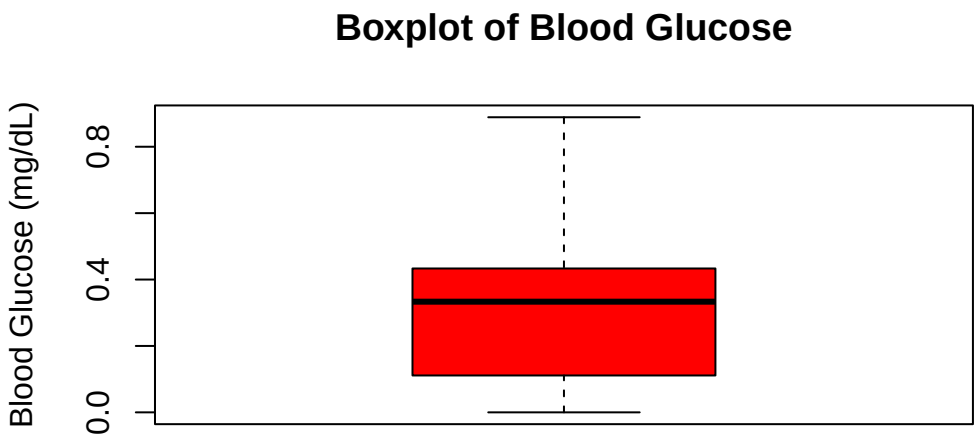

```r
# Boxplot for Age
boxplot(cleaned_data$age,
        main = "Boxplot of Age",
        ylab = "Age",
        col = "blue",
        border = "black",
        horizontal = FALSE)
```



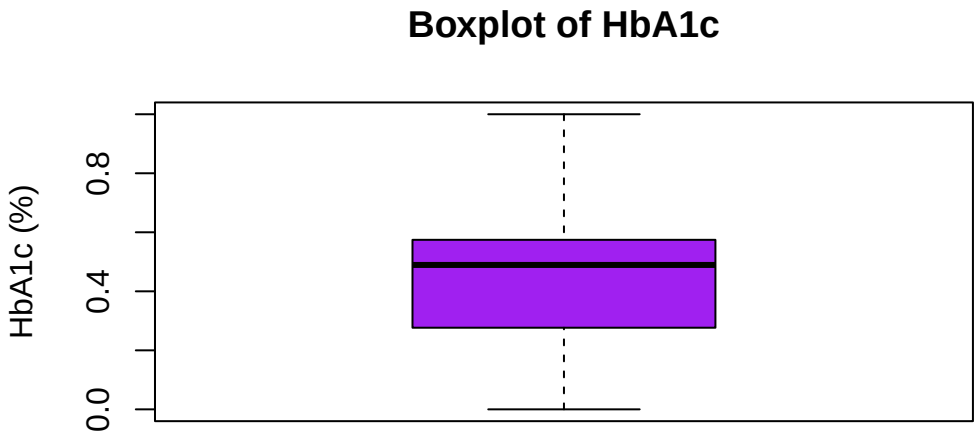

```
# Boxplot for BMI
boxplot(cleaned_data$bmi,
        main = "Boxplot of BMI",
        ylab = "BMI",
        col = "green",
        border = "black",
        horizontal = FALSE)
```



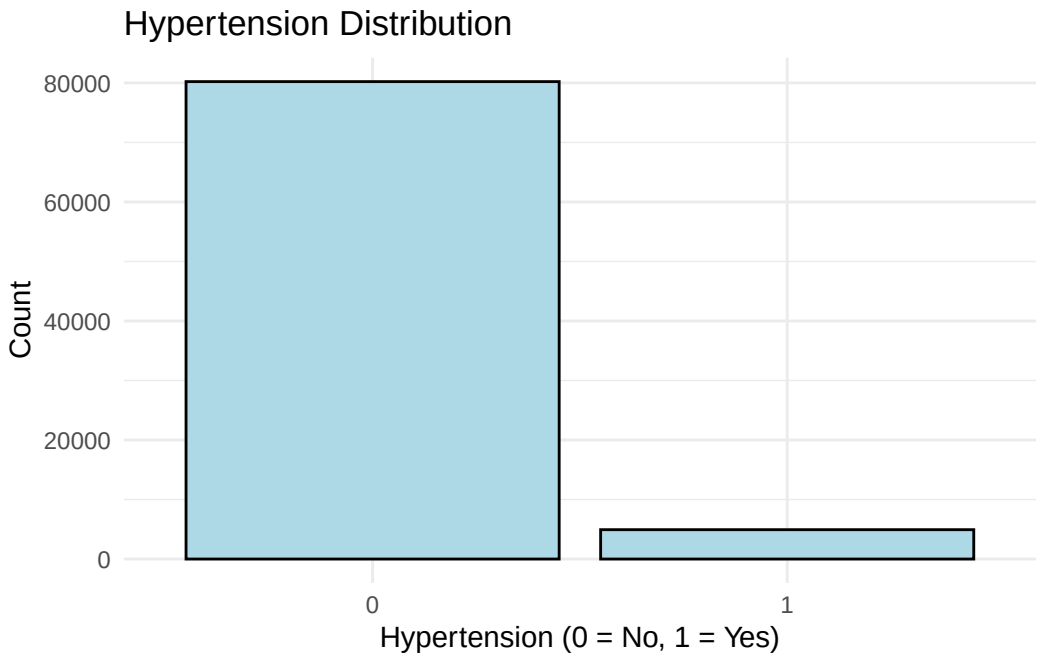

```r
# Boxplot for BloodGlucose
boxplot(cleaned_data$blood_glucose_level,
        main = "Boxplot of Blood Glucose",
        ylab = "Blood Glucose (mg/dL)",
        col = "red",
        border = "black",
        horizontal = FALSE)
```



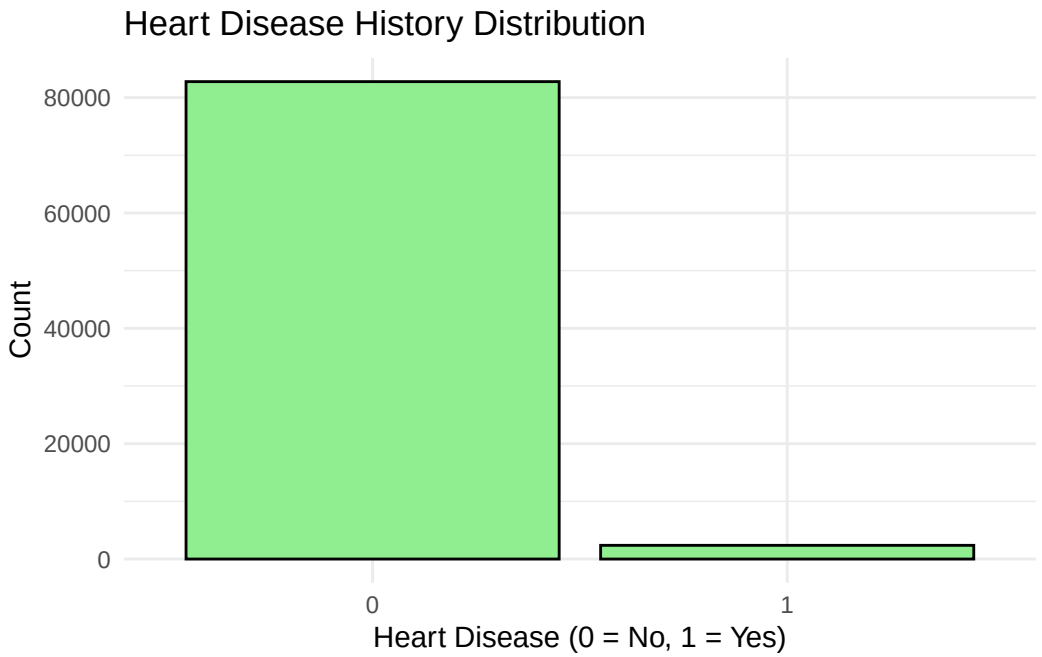

```r
# Boxplot for HbA1c
boxplot(cleaned_data$HbA1c_level,
        main = "Boxplot of HbA1c",
        ylab = "HbA1c (%)",
        col = "purple",
        border = "black",
        horizontal = FALSE)
```



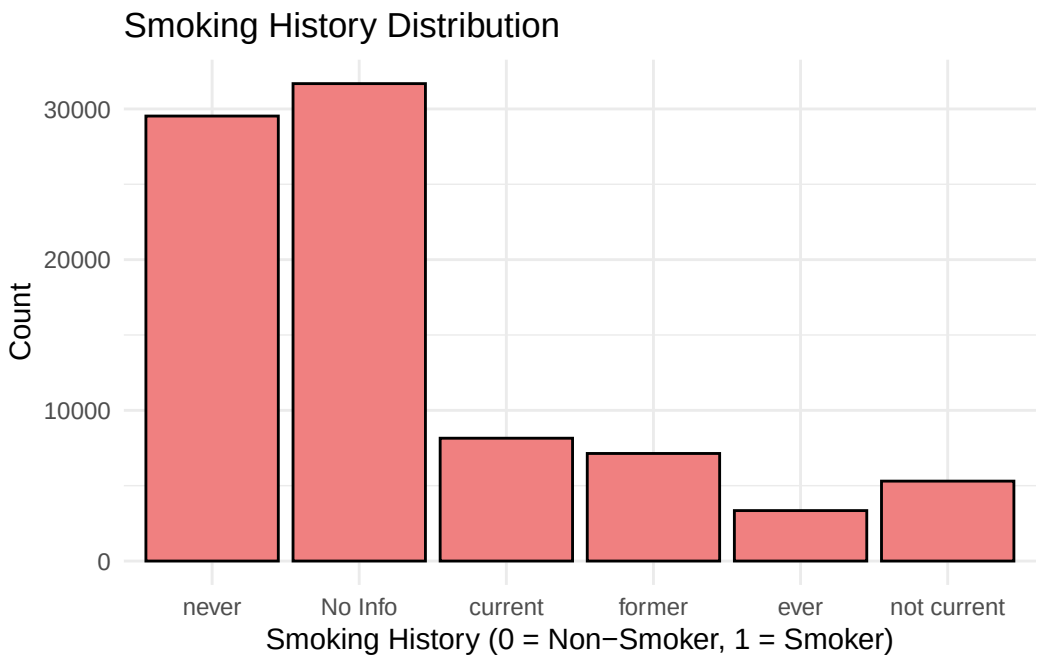

```
# Bar plot for Hypertension
ggplot(cleaned_data, aes(x = factor(hypertension))) +
  geom_bar(fill = "lightblue", color = "black") +
  labs(title = "Hypertension Distribution", x = "Hypertension (0 = No, 1 =
   ↪  Yes)", y = "Count") +
  theme_minimal()
```



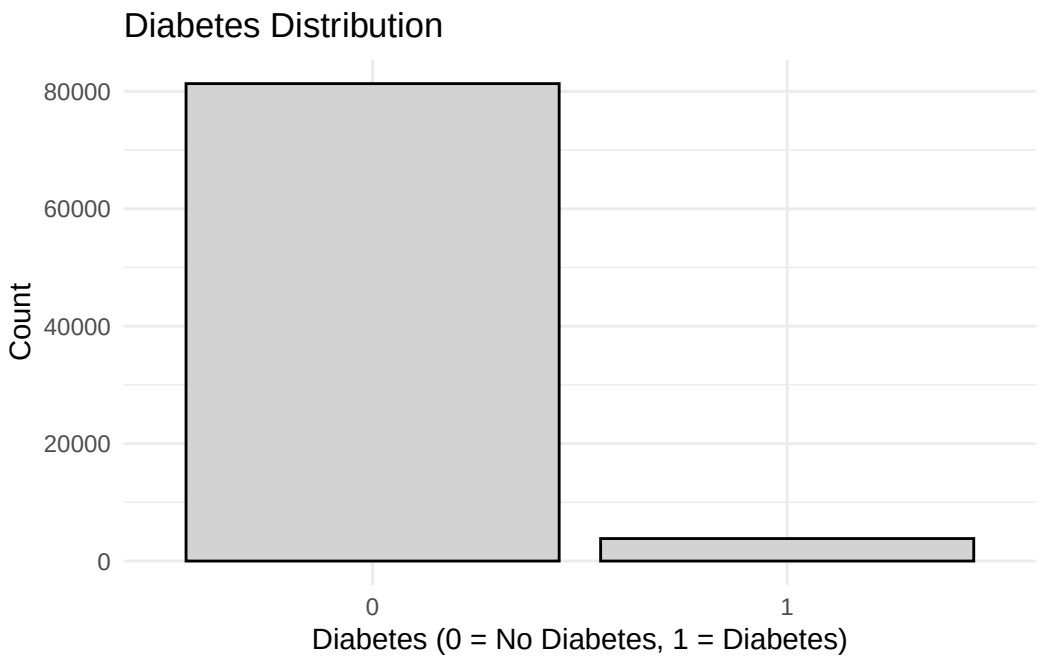

```
# Bar plot for HeartDisease
ggplot(cleaned_data, aes(x = factor(heart_disease))) +
  geom_bar(fill = "lightgreen", color = "black") +
  labs(title = "Heart Disease History Distribution", x = "Heart Disease (0 =
   ↪  No, 1 = Yes)", y = "Count") +
  theme_minimal()
```




```
# Bar plot for SmokingHistory
ggplot(cleaned_data, aes(x = factor(smoking_history))) +
  geom_bar(fill = "lightcoral", color = "black") +
  labs(title = "Smoking History Distribution", x = "Smoking History (0 =
   ↪  Non-Smoker, 1 = Smoker)", y = "Count") +
  theme_minimal()
```




```
# Bar plot for Diabetes
ggplot(cleaned_data, aes(x = factor(diabetes))) +
  geom_bar(fill = "lightgrey", color = "black") +
  labs(title = "Diabetes Distribution", x = "Diabetes (0 = No Diabetes, 1 =
   ↪  Diabetes)", y = "Count") +
  theme_minimal()
```



Analysis

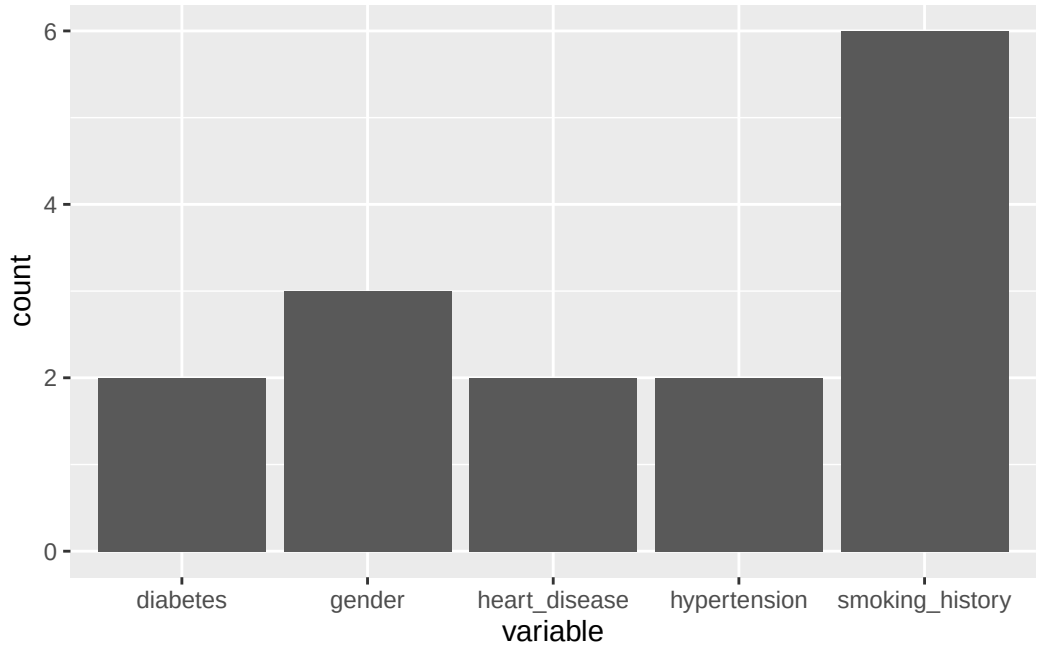
Continuous variables

```
continuous <- cleaned_data %>%
  select(where(is.numeric))
summary(continuous)
```

```
      age               bmi          HbA1c_level     blood_glucose_level
 Min.   :0.0000   Min.   :0.1158   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.2777   1st Qu.:0.3406   1st Qu.:0.2766   1st Qu.:0.1111
 Median :0.5058   Median :0.4464   Median :0.4894   Median :0.3333
 Mean   :0.4926   Mean   :0.4185   Mean   :0.4154   Mean   :0.3023
 3rd Qu.:0.7086   3rd Qu.:0.4716   3rd Qu.:0.5745   3rd Qu.:0.4333
 Max.   :1.0000   Max.   :0.7272   Max.   :1.0000   Max.   :0.8889
```

Factor Variables

```r
cleaned_data %>%
  summarize(across(where(is.factor), n_distinct)) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to =
   ↪  "count") %>%
  ggplot(aes(x = variable, y = count)) +
  geom_bar(stat = "identity")
```



From the graph above, we can see that smoking history has 6 levels. This is somewhat substantial, and some levels have relatively low number of observations. These are some values to take notice for the model.

Feature Engineering

```r
cleaned_data %>%
  count(smoking_history)
```

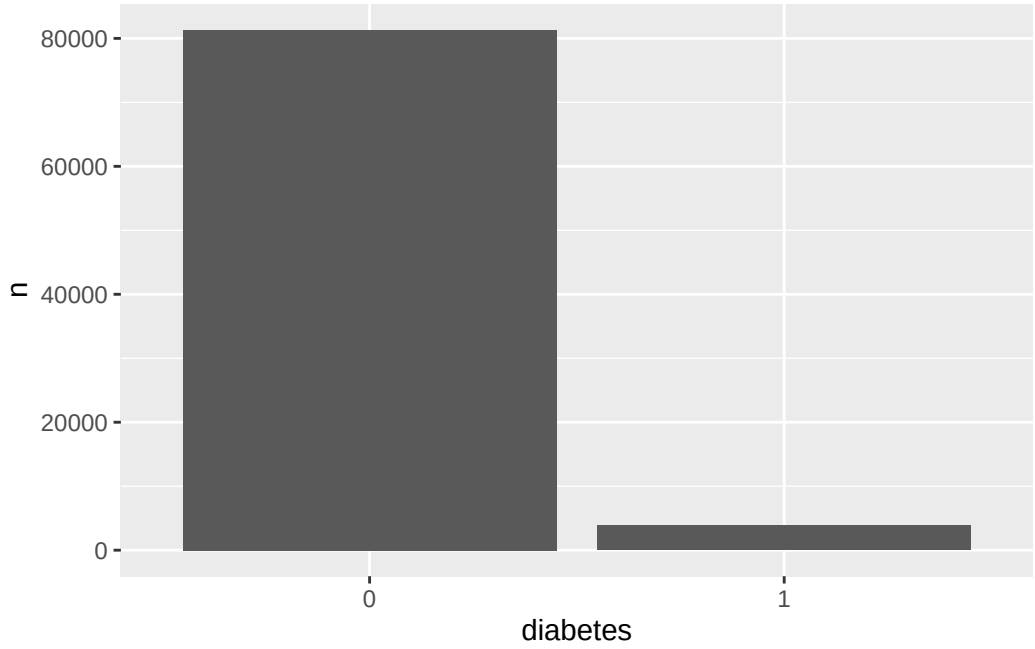
```
# A tibble: 6 x 2
  smoking_history     n
  <fct>           <int>
1 never           29530
2 No Info         31682
```

```
3 current          8151
4 former           7138
5 ever             3348
6 not current      5304
```

Here we can see that even if we wanted to remove the no info column from our data set, there are a substantial amount of points that contribute to our models variance.

Summary Statistic

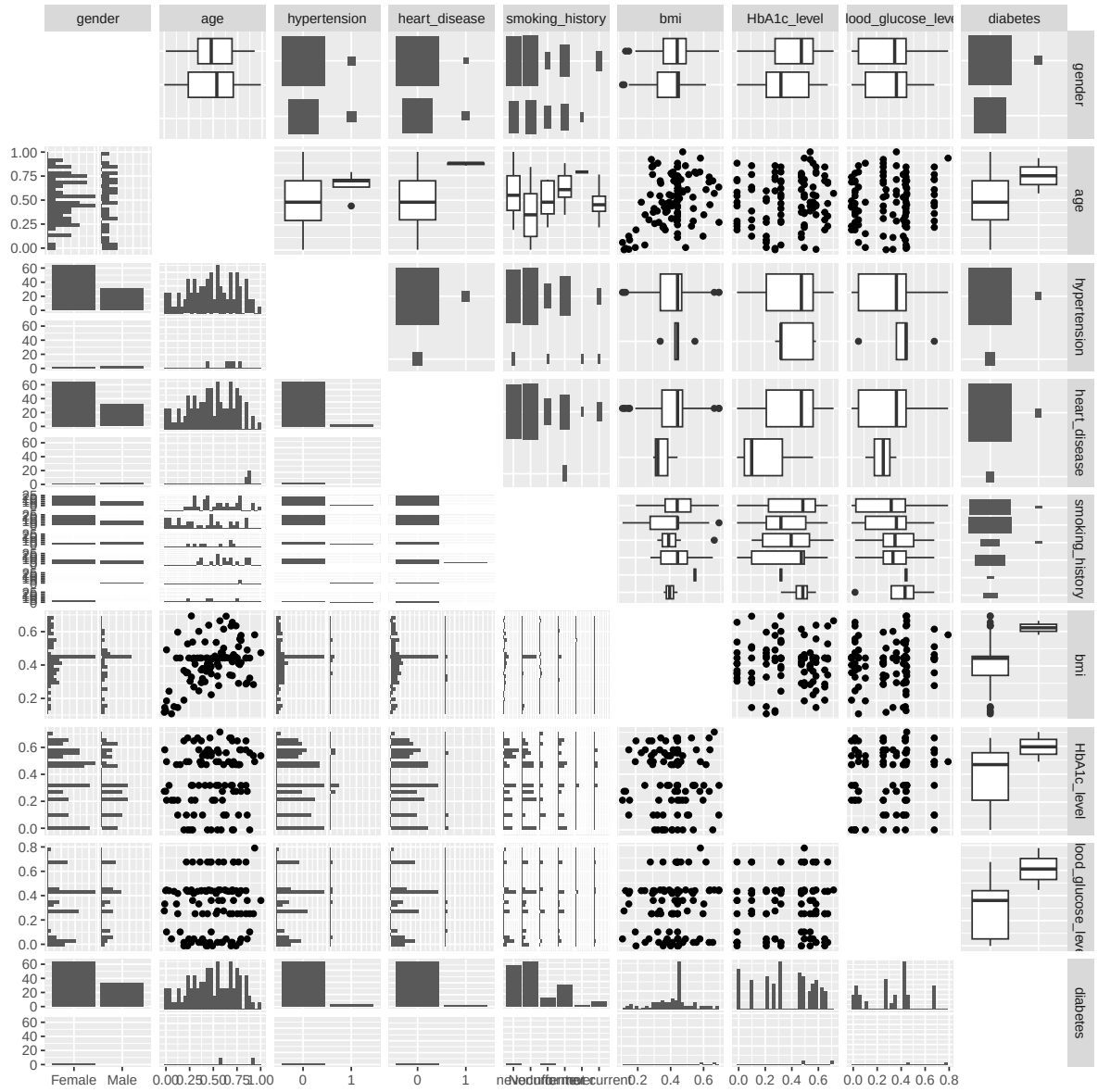
```
cleaned_data %>%
  count(diabetes) %>%
  ggplot(aes(x = diabetes, y = n))+
  geom_bar(stat = "identity")
```



There is an extreme imbalance in the range of the distribution this is something to be aware of for our model implementiation and to take note of if there are any issues in our implementation.

```
cleaned_data %>%
  sample_n(size = 100) %>%
  ggpairs(columns = 1:ncol(cleaned_data), diag = "blankDiag", upper =
   ↪  list(continuous = "points"))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The diabetes column has a well-disperse binary target , so it's a good fit for classification, and its correlations with features like BMI, level of HbA1c, and blood glucose present strong patterns for prediction.

Train/Test & Building Model

```
set.seed(1234)
```

```r
create_train_test <- function(data, size = 0.8, train = TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <- 1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
    mean(glm_pred == as.character(testing$diabetes))
  }
}

data_train <- create_train_test(cleaned_data, 0.7, train = TRUE)
data_test <- create_train_test(cleaned_data, 0.7, train = FALSE)
dim(data_train)
```

```
[1] 59607     9
```

```r
glm_fit <- glm(diabetes ~ ., data = data_train, family = "binomial")
predictions <- predict(glm_fit, newdata = data_test, type = "response")
glm_pred <- rep("0", nrow(data_test))
glm_pred[predictions > 0.5] = "1"

table(glm_pred)
```

```
glm_pred
    0     1
25020   526
```

```r
table(data_test$diabetes)
```

```
    0     1
24445  1101
```

```r
# for the accuracy mentioned further down in reference
ac <- mean(glm_pred == data_test$diabetes)

tidy(glm_fit)
```

```
# A tibble: 14 x 5
   term                        estimate std.error statistic   p.value
   <chr>                          <dbl>     <dbl>     <dbl>     <dbl>
 1 (Intercept)                 -15.4       0.236   -65.4     0
 2 genderMale                    0.223     0.0501    4.45    8.59e-  6
 3 genderOther                  -9.78    116.       -0.0840  9.33e-  1
 4 age                           3.74      0.131    28.5     3.17e-178
 5 hypertension1                 0.759     0.0668   11.4     6.73e- 30
 6 heart_disease1                0.870     0.0861   10.1     5.50e- 24
 7 smoking_historyNo Info       -0.502     0.0688   -7.29    3.03e- 13
 8 smoking_historycurrent        0.112     0.0827    1.35    1.76e-  1
 9 smoking_historyformer         0.135     0.0737    1.83    6.75e-  2
10 smoking_historyever           0.153     0.111     1.38    1.67e-  1
11 smoking_historynot current   -0.0440    0.0989   -0.445   6.56e-  1
12 bmi                           4.11      0.232    17.7     2.68e- 70
13 HbA1c_level                  11.1       0.251    44.1     0
14 blood_glucose_level           4.82      0.135    35.7     1.59e-278
```

Assess Model Performance

confusion matrixes for the next two

```
summary(glm_fit)
```

```
Call:
glm(formula = diabetes ~ ., family = "binomial", data = data_train)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -15.41605    0.23568 -65.412  < 2e-16 ***
genderMale                   0.22294    0.05010   4.450 8.59e-06 ***
genderOther                 -9.77942  116.36454  -0.084   0.9330
age                          3.73853    0.13134  28.465  < 2e-16 ***
hypertension1                0.75866    0.06679  11.358  < 2e-16 ***
heart_disease1               0.86971    0.08611  10.100  < 2e-16 ***
smoking_historyNo Info      -0.50157    0.06877  -7.293 3.03e-13 ***
smoking_historycurrent       0.11191    0.08270   1.353   0.1760
smoking_historyformer        0.13482    0.07374   1.828   0.0675 .
smoking_historyever          0.15286    0.11074   1.380   0.1675
smoking_historynot current  -0.04401    0.09892  -0.445   0.6564
bmi                          4.10937    0.23184  17.725  < 2e-16 ***
```

```
HbA1c_level                 11.07229    0.25122  44.075  < 2e-16 ***
blood_glucose_level          4.82379    0.13527  35.661  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22169  on 59606  degrees of freedom
Residual deviance: 11961  on 59593  degrees of freedom
AIC: 11989

Number of Fisher Scoring iterations: 12
```

```r
table(glm_pred, data_test$diabetes)
```

```
glm_pred     0     1
       0 24330   690
       1   115   411
```

The model is 96.4% accurate, which means it is right about most cases. However, its precision is only 37.4%, i.e., it often predicts diabetes when the person does not have it. The recall is 78.1%, which means that it correctly classifies most actual cases of diabetes but occasionally gets them wrong. While the model is fairly accurate, its low precision suggests that it may not be very effective in avoiding false positives.

Null Model

```r
null_model <- glm(diabetes ~ 1, data = data_train, family = "binomial")

# Check the summary of the null model
summary(null_model)
```

```
Call:
glm(formula = diabetes ~ 1, family = "binomial", data = data_train)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.03659    0.01959    -155   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22169  on 59606  degrees of freedom
Residual deviance: 22169  on 59606  degrees of freedom
AIC: 22171

Number of Fisher Scoring iterations: 5
```

The null model only predicts diabetes based on the proportion in the data set, without using any features. Because our intercept is negative, it suggest the models info has less occurances of diabetes.

Our Model

```
coefs <- coef(glm_fit)
equation <- paste0("Logit(P(Y=1)) = ", round(coefs[1], 3), " + ",
                   paste(names(coefs[-1]), round(coefs[-1], 3), sep = " * ",
                    ↪  collapse = " + "))
wrapped_equation <- strwrap(equation, width = 80)
cat(wrapped_equation, sep = "\n")
```

```
Logit(P(Y=1)) = -15.416 + genderMale * 0.223 + genderOther * -9.779 + age *
3.739 + hypertension1 * 0.759 + heart_disease1 * 0.87 + smoking_historyNo Info
* -0.502 + smoking_historycurrent * 0.112 + smoking_historyformer * 0.135 +
smoking_historyever * 0.153 + smoking_historynot current * -0.044 + bmi * 4.109
+ HbA1c_level * 11.072 + blood_glucose_level * 4.824
```

```
wrapped_equation
```

```
[1] "Logit(P(Y=1)) = -15.416 + genderMale * 0.223 + genderOther * -9.779 + age *"
[2] "3.739 + hypertension1 * 0.759 + heart_disease1 * 0.87 + smoking_historyNo Info"
[3] "* -0.502 + smoking_historycurrent * 0.112 + smoking_historyformer * 0.135 +"
[4] "smoking_historyever * 0.153 + smoking_historynot current * -0.044 + bmi * 4.109"
[5] "+ HbA1c_level * 11.072 + blood_glucose_level * 4.824"
```

$$\begin{aligned}\log \left(\frac{P(Y=1)}{1 - P(Y=1)} \right) &= -15.416 + 0.223 \cdot \text{genderMale} \\&\quad - 9.779 \cdot \text{genderOther} + 3.739 \cdot \text{age} \\&\quad + 0.759 \cdot \text{hypertension1} + 0.87 \cdot \text{heart\_disease1} \\&\quad - 0.502 \cdot \text{smoking\_historyNo\_Info} + 0.112 \cdot \text{smoking\_historycurrent} \\&\quad + 0.135 \cdot \text{smoking\_historyformer} + 0.153 \cdot \text{smoking\_historyever} \\&\quad - 0.044 \cdot \text{smoking\_historynot\_current} + 4.109 \cdot \text{bmi} \\&\quad + 11.072 \cdot \text{HbA1c\_level} + 4.824 \cdot \text{blood\_glucose\_level}\end{aligned}$$

```
summary(glm_fit)
```

```
Call:
glm(formula = diabetes ~ ., family = "binomial", data = data_train)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -15.41605    0.23568 -65.412  < 2e-16 ***
genderMale                   0.22294    0.05010   4.450 8.59e-06 ***
genderOther                 -9.77942  116.36454  -0.084   0.9330
age                          3.73853    0.13134  28.465  < 2e-16 ***
hypertension1                0.75866    0.06679  11.358  < 2e-16 ***
heart_disease1               0.86971    0.08611  10.100  < 2e-16 ***
smoking_historyNo Info      -0.50157    0.06877  -7.293 3.03e-13 ***
smoking_historycurrent       0.11191    0.08270   1.353   0.1760
smoking_historyformer        0.13482    0.07374   1.828   0.0675 .
smoking_historyever          0.15286    0.11074   1.380   0.1675
smoking_historynot current  -0.04401    0.09892  -0.445   0.6564
bmi                          4.10937    0.23184  17.725  < 2e-16 ***
HbA1c_level                 11.07229    0.25122  44.075  < 2e-16 ***
blood_glucose_level          4.82379    0.13527  35.661  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22169  on 59606  degrees of freedom
Residual deviance: 11961  on 59593  degrees of freedom
AIC: 11989

Number of Fisher Scoring iterations: 12
```

The logistic regression mode shows a significant reduction in deviation from 22169, found from

our null, to the 11961 in our residual deviance. This showcases the the predictors variable contribute meaningful explanations for the output. The AIC of 11,989 suggest a reasonable ballance between the models complexity and fit. The model converged in 12 fisher scoring iteration, showcasing stable parameter estimation. These metric indicated the this is a well fitting model however, we can continue to asses

Model Comparison with Reduction in deviance

```
anova(glm_fit, test = "Chisq")
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: diabetes

Terms added sequentially (first to last)

                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                59606      22169
gender               2     96.1     59604      22073 < 2.2e-16 ***
age                  1   2775.2     59603      19298 < 2.2e-16 ***
hypertension         1    320.0     59602      18978 < 2.2e-16 ***
heart_disease        1    145.4     59601      18832 < 2.2e-16 ***
smoking_history      5    177.4     59596      18655 < 2.2e-16 ***
bmi                  1    488.3     59595      18167 < 2.2e-16 ***
HbA1c_level          1   4688.2     59594      13478 < 2.2e-16 ***
blood_glucose_level  1   1517.7     59593      11961 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Application to GLM

BIC

```
bic_value <- BIC(glm_fit)
print(paste("BIC:", bic_value))
```

```
[1] "BIC: 12114.5283165497"
```

The BIC assesses model fit while analyzing complexity, where lower values indicate a good balance between accuracy and parsimony. If the BIC decreases after removing unnecessary variables, the model is likely more efficient without sacrificing predictability. If the BIC increases, significant predictors may have been removed, and the model may need to be re-checked for feature selection.

Model evaluation and predictions

Model Accuracy

```
# model Accuracy (Full model due to no need for feature selection)
mean(glm_pred == data_test$diabetes)
```

```
[1] 0.9684882
```

```
# model Accuracy of the NULL
mean(null_model == data_test$diabetes)
```

```
Warning in `==.default`(null_model, data_test$diabetes): longer object length
is not a multiple of shorter object length
```

```
Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
shorter object length
```

```
[1] 0.0009786268
```

The full model obtained an accuracy of 96.85%, while the null models accuracy was only 0.098%. This indicates that the full model effectively predicts diabetes, whereas the null model performs no better than random guessing

likelihood Ratio Test:

```
anova(null_model, glm_fit, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: diabetes ~ 1
Model 2: diabetes ~ gender + age + hypertension + heart_disease + smoking_history +
    bmi + HbA1c_level + blood_glucose_level
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1     59606      22169
2     59593      11961 13    10208 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test is used to determine whether the addition of predictors significantly improves the model's performance. The full model significantly improves diabetes prediction over our null model, from 22,169 to 11,961 residual deviance with a chi-square of 10,208 and a p value of <2.2e-16

Pseudo R-Squared

```r
null_logLik <- logLik(null_model)

full_logLik <- logLik(glm_fit)

# McFadden's Pseudo R-squared
pseudo_r2 <- 1 - (as.numeric(full_logLik) / as.numeric(null_logLik))
print(paste("Pseudo R-squared:", pseudo_r2))
```

```
[1] "Pseudo R-squared: 0.460475569043259"
```

The pseudo R-squared of 0.4605 indicates that the full model explains about 46.05% of the variation we see with the diabetes outcomes compared to our null model. The value shows a moderately strong fit.

Conclusion/Summary

For this project, we created a logistic regression model designed to predict diabetes via various health factors. The data used in this experiment came from the Diabetes Prediction Dataset from Kaggle, which contains certain key explanatory variables, including gender, BMI, age, hypertensive status, history of heart disease, history of smoking, A1c levels, and blood glucose levels. We began our analysis by first cleaning our data, via removing outliers and normalizing numerical variables to improve model accuracy. We then implemented our logistic regression to

predict diabetes, and we evaluated our model's accuracy. Our model achieved a high accuracy of 96.8%, much higher than the model null accuracy that we found of .0009. Evaluation via examining residual deviance, AIC, a likelihood ratio test, and a pseudo R^2 test confirm the statistical value of our chosen model. With regards to positive aspects of our project, we were able to correctly select predictors that do contribute to the high accuracy of our model. Diabetes can definitely be explained by our predictors, as shown via our verification methods. With regards to the negative aspects of our project, we definitely had a pretty low precision of only about 37.4%, and our model does have an issue with avoiding false positives. Also, our data and subsequent model lacked many variables that could add greatly to our model, including socioeconomic variables, mental health variables, lifestyle variables, and genetic variables (predisposition to diabetes, other health risks). Looking forward, our model could be improved by incorporation of those other diabetes-relevant variables. Our dataset was somewhat limited, and including more variables could definitely help us avoid type 1 error.

References

Kassambara, Thanos, Kassambara, & Sfd. (2018, March 11). Logistic Regression Essentials in R. STHDA. http://www.sthda.com/english/articles/36-classification-methods-essentials/151-logistic-regression-essentials-in-r/#:%7E:text=The%20R%20function%20glm(),want%20to%20fit%20logistic%2

Marx, J. (2002). Unraveling the Causes of Diabetes. Science 296, 5568. http://www.jstor.org/stable/3076573

Robinson, M., & Turner, C. (2019). Incidence and prevalence of type 2 diabetes in America: Is there culpability in the food industry? State Crime Journal, 8(2). https://doi.org/10.13169/statecrime.8.2.0175

Soga. SOGA •. (2016, August 30). https://www.geo.fu-berlin.de/en/v/soga/Basics-of-statistics/Logistic-Regression/Logistic-Regression-in-R—An-Example/index.html