Diabetes Classification Model

Jason Finkle

2025-07-14

library(readr)  
library(naniar)  
library(tidyverse)

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'purrr' was built under R version 4.3.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ purrr 1.0.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggplot2)  
library(gridExtra)

## Warning: package 'gridExtra' was built under R version 4.3.3

##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.3.3

## randomForest 4.7-1.2  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:gridExtra':  
##   
## combine  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library(caret)

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(pROC)

## Warning: package 'pROC' was built under R version 4.3.3

## Type 'citation("pROC")' for a citation.  
##   
## Attaching package: 'pROC'  
##   
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(e1071)

## Warning: package 'e1071' was built under R version 4.3.3

library(tidyr)  
##########################################################################  
  
#Initial Data Cleaning and Exploration  
df <- read\_csv("~/Desktop/diabetes\_project/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.

head(df)

## # A tibble: 6 × 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>

colnames(df)

## [1] "gender" "age" "hypertension"   
## [4] "heart\_disease" "smoking\_history" "bmi"   
## [7] "HbA1c\_level" "blood\_glucose\_level" "diabetes"

df %>%  
 select(age, bmi, HbA1c\_level, blood\_glucose\_level) %>%  
 summary()

## age bmi HbA1c\_level blood\_glucose\_level  
## Min. : 0.08 Min. :10.01 Min. :3.500 Min. : 80.0   
## 1st Qu.:24.00 1st Qu.:23.63 1st Qu.:4.800 1st Qu.:100.0   
## Median :43.00 Median :27.32 Median :5.800 Median :140.0   
## Mean :41.89 Mean :27.32 Mean :5.528 Mean :138.1   
## 3rd Qu.:60.00 3rd Qu.:29.58 3rd Qu.:6.200 3rd Qu.:159.0   
## Max. :80.00 Max. :95.69 Max. :9.000 Max. :300.0

# Count missing values per column  
colSums(is.na(df))

## gender age hypertension heart\_disease   
## 0 0 0 0   
## smoking\_history bmi HbA1c\_level blood\_glucose\_level   
## 0 0 0 0   
## diabetes   
## 0

# Gender distribution  
table(df$gender)

##   
## Female Male Other   
## 58552 41430 18

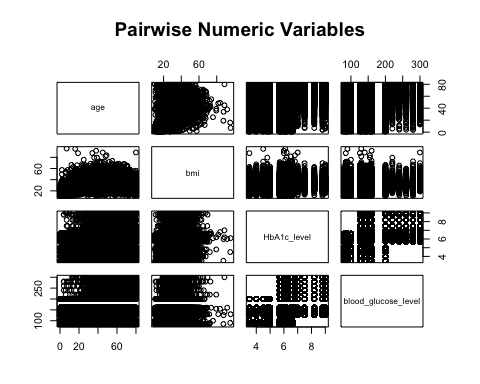
# Smoking history distribution  
table(df$smoking\_history)

##   
## current ever former never No Info not current   
## 9286 4004 9352 35095 35816 6447

# Diabetes prevalence  
table(df$diabetes)

##   
## 0 1   
## 91500 8500

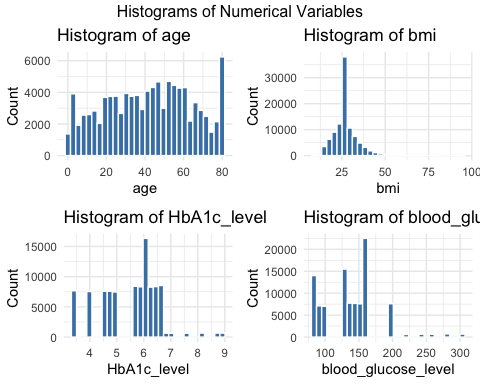
pairs(df %>% select(age, bmi, HbA1c\_level, blood\_glucose\_level), main = "Pairwise Numeric Variables")



# List of numerical variables  
num\_vars <- c("age", "bmi", "HbA1c\_level", "blood\_glucose\_level")  
  
  
  
# Create list of histograms  
hist\_list <- lapply(num\_vars, function(var) {  
 ggplot(df, aes\_string(x = var)) +  
 geom\_histogram(fill = "steelblue", color = "white", bins = 30) +  
 theme\_minimal() +  
 labs(title = paste("Histogram of", var), x = var, y = "Count")  
})

## Warning: `aes\_string()` was deprecated in ggplot2 3.0.0.  
## ℹ Please use tidy evaluation idioms with `aes()`.  
## ℹ See also `vignette("ggplot2-in-packages")` for more information.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

# Create list of boxplots  
box\_list <- lapply(num\_vars, function(var) {  
 ggplot(df, aes\_string(x = "diabetes", y = var, fill = "diabetes")) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(title = paste("Boxplot of", var, "by Diabetes Status"),  
 x = "Diabetes", y = var) +  
 theme(legend.position = "none")  
})  
  
  
# Combine histograms  
grid.arrange(grobs = hist\_list, ncol = 2, top = "Histograms of Numerical Variables")



# Combine boxplots  
grid.arrange(grobs = box\_list, ncol = 2, top = "Boxplots by Diabetes Status")

## Warning: Continuous x aesthetic  
## ℹ did you forget `aes(group = ...)`?

## Warning: The following aesthetics were dropped during statistical transformation: fill.  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?

## Warning: Continuous x aesthetic  
## ℹ did you forget `aes(group = ...)`?

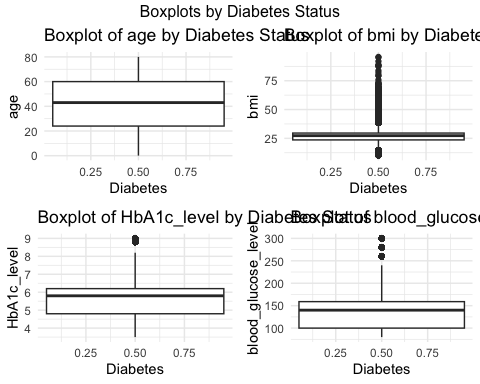
## Warning: The following aesthetics were dropped during statistical transformation: fill.  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?

## Warning: Continuous x aesthetic  
## ℹ did you forget `aes(group = ...)`?

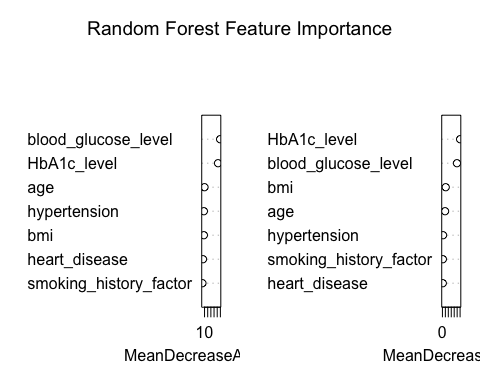
## Warning: The following aesthetics were dropped during statistical transformation: fill.  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?

## Warning: Continuous x aesthetic  
## ℹ did you forget `aes(group = ...)`?

## Warning: The following aesthetics were dropped during statistical transformation: fill.  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?



#Removing observations with no smoking info  
df\_clean <- df %>%  
 filter(!smoking\_history %in% c("ever", "No Info"))  
  
  
#combining former and not\_current  
df\_clean <- df\_clean %>%  
 mutate(smoking\_history = case\_when(  
 smoking\_history %in% c("not current", "former") ~ "Not\_Current",  
 TRUE ~ smoking\_history  
 ))  
  
#making the categorical variable smoking into an ordinal variable  
df\_clean <- df\_clean %>%  
 mutate(smoking\_history\_factor = case\_when(  
 smoking\_history == "current" ~ 2,  
 smoking\_history == "Not\_Current" ~ 1,  
 smoking\_history == "never" ~ 0,  
 TRUE ~ NA\_real\_ # optional: assigns NA to any unexpected category  
 ))  
  
  
#Removing non numeric categories  
df\_clean <- df\_clean %>%  
 select(where(is.numeric))  
  
# Ensure diabetes is a factor with consistent levels BEFORE splitting  
df\_clean <- df\_clean %>%  
 mutate(diabetes = factor(diabetes, levels = c(0,1)))  
  
############################################################################  
#Random Forest Model  
set.seed(123) # for reproducibility  
  
train\_index <- createDataPartition(df\_clean$diabetes, p = 0.7, list = FALSE)  
  
train\_data <- df\_clean[train\_index, ]  
test\_data <- df\_clean[-train\_index, ]  
  
  
  
rf\_model <- randomForest(  
 diabetes ~ .,   
 data = train\_data,   
 importance = TRUE,  
 ntree = 100  
)  
  
predictions <- predict(rf\_model, newdata = test\_data)  
  
  
varImpPlot(rf\_model, main = "Random Forest Feature Importance")



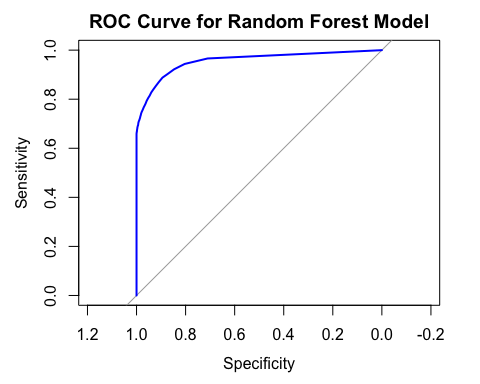
cm <- confusionMatrix(predictions, test\_data$diabetes)  
cm

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 16072 662  
## 1 9 1310  
##   
## Accuracy : 0.9628   
## 95% CI : (0.96, 0.9655)  
## No Information Rate : 0.8908   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7765   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9994   
## Specificity : 0.6643   
## Pos Pred Value : 0.9604   
## Neg Pred Value : 0.9932   
## Prevalence : 0.8908   
## Detection Rate : 0.8903   
## Detection Prevalence : 0.9269   
## Balanced Accuracy : 0.8319   
##   
## 'Positive' Class : 0   
##

# Get predicted probabilities (for class '1' or 'Yes')  
rf\_probs <- predict(rf\_model, newdata = test\_data, type = "prob")[, 2]  
  
  
roc\_obj <- roc(test\_data$diabetes, rf\_probs)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

auc\_value <- auc(roc\_obj)  
  
# Plot ROC Curve  
plot(roc\_obj, col = "blue", main = "ROC Curve for Random Forest Model")



accuracy\_rf <- cm$overall["Accuracy"]  
precision\_rf <- cm$byClass["Precision"]  
recall\_rf <- cm$byClass["Recall"]  
f1\_rf <- cm$byClass["F1"]  
  
  
###########################################################################  
  
#Logistic Regression  
  
log\_model <- glm(diabetes ~ ., data = train\_data, family = binomial)  
  
# Predict probabilities and classes  
log\_probs <- predict(log\_model, newdata = test\_data, type = "response")  
log\_pred <- ifelse(log\_probs > 0.5, 1, 0)  
log\_pred <- factor(log\_pred, levels = levels(test\_data$diabetes))  
  
# Confusion Matrix  
log\_cm <- confusionMatrix(log\_pred, test\_data$diabetes)  
log\_cm

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 15936 739  
## 1 145 1233  
##   
## Accuracy : 0.951   
## 95% CI : (0.9478, 0.9541)  
## No Information Rate : 0.8908   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7101   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9910   
## Specificity : 0.6253   
## Pos Pred Value : 0.9557   
## Neg Pred Value : 0.8948   
## Prevalence : 0.8908   
## Detection Rate : 0.8827   
## Detection Prevalence : 0.9237   
## Balanced Accuracy : 0.8081   
##   
## 'Positive' Class : 0   
##

# ROC and AUC  
log\_roc <- roc(test\_data$diabetes, log\_probs)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

log\_auc <- auc(log\_roc)  
  
accuracy\_log <- log\_cm$overall["Accuracy"]  
precision\_log<- log\_cm$byClass["Precision"]  
recall\_log <- log\_cm$byClass["Recall"]  
f1\_log <- log\_cm$byClass["F1"]  
  
#############################################################################  
  
#SVM model  
  
# Train SVM  
svm\_model <- svm(diabetes ~ ., data = train\_data, probability = TRUE)  
  
# Class Predictions  
svm\_pred <- predict(svm\_model, newdata = test\_data)  
  
# Probabilities  
svm\_probs <- attr(predict(svm\_model, test\_data, probability = TRUE), "probabilities")[,2]  
  
# Confusion Matrix  
svm\_cm <- confusionMatrix(svm\_pred, test\_data$diabetes)  
svm\_cm

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 16053 829  
## 1 28 1143  
##   
## Accuracy : 0.9525   
## 95% CI : (0.9493, 0.9556)  
## No Information Rate : 0.8908   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7032   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9983   
## Specificity : 0.5796   
## Pos Pred Value : 0.9509   
## Neg Pred Value : 0.9761   
## Prevalence : 0.8908   
## Detection Rate : 0.8892   
## Detection Prevalence : 0.9351   
## Balanced Accuracy : 0.7889   
##   
## 'Positive' Class : 0   
##

# ROC and AUC  
svm\_roc <- roc(test\_data$diabetes, svm\_probs)

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

svm\_auc <- auc(svm\_roc)  
  
accuracy\_log <- svm\_cm$overall["Accuracy"]  
precision\_log<- svm\_cm$byClass["Precision"]  
recall\_log <- svm\_cm$byClass["Recall"]  
f1\_log <- svm\_cm$byClass["F1"]  
  
  
  
############################################################################  
  
#Model Assessment and Comparison  
  
rf\_metrics <- data.frame(  
 Model = "Random Forest",  
 Accuracy = round(accuracy\_rf, 4),  
 Precision = round(precision\_rf, 4),  
 Recall = round(recall\_rf, 4),  
 F1\_Score = round(f1\_rf, 4),  
 AUC = round(auc\_value, 4)  
)  
  
log\_metrics <- data.frame(  
 Model = "Logistic Regression",  
 Accuracy = round(log\_cm$overall["Accuracy"], 4),  
 Precision = round(log\_cm$byClass["Precision"], 4),  
 Recall = round(log\_cm$byClass["Recall"], 4),  
 F1\_Score = round(log\_cm$byClass["F1"], 4),  
 AUC = round(log\_auc, 4)  
)  
  
svm\_metrics <- data.frame(  
 Model = "SVM",  
 Accuracy = round(svm\_cm$overall["Accuracy"], 4),  
 Precision = round(svm\_cm$byClass["Precision"], 4),  
 Recall = round(svm\_cm$byClass["Recall"], 4),  
 F1\_Score = round(svm\_cm$byClass["F1"], 4),  
 AUC = round(svm\_auc, 4)  
)  
  
# Combine with Random Forest metrics  
all\_metrics <- bind\_rows(rf\_metrics, log\_metrics, svm\_metrics)  
  
  
metrics\_long <- all\_metrics %>%  
 pivot\_longer(cols = c(Accuracy, Precision, Recall, F1\_Score, AUC),  
 names\_to = "Metric",  
 values\_to = "Value")  
  
  
#Metrics Plot  
ggplot(metrics\_long, aes(x = Model, y = Value, fill = Model)) +  
 geom\_col() +  
 facet\_wrap(~Metric, scales = "free\_y") +  
 theme\_minimal() +  
 labs(title = "Model Metrics Breakdown (Zoomed In)") +  
 coord\_cartesian(ylim = c(0.85, 1)) + # Adjust starting point here  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



##############################################################################  
#Risk Evaluation  
  
  
rf\_probs <- predict(rf\_model, newdata = test\_data, type = "prob")[, 2]  
svm\_probs <- attr(predict(svm\_model, newdata = test\_data, probability = TRUE), "probabilities")[, 2]  
test\_data <- test\_data %>%  
 mutate(  
 combined\_prob = (log\_probs + rf\_probs + svm\_probs) / 3  
 )  
  
test\_data <- test\_data %>%  
 mutate(  
 combined\_risk = case\_when(  
 combined\_prob <= 0.3 ~ "Low Risk",  
 combined\_prob <= 0.6 ~ "Moderate Risk",  
 TRUE ~ "High Risk"  
 )  
 )  
  
ggplot(test\_data, aes(x = combined\_prob, fill = combined\_risk)) +  
 geom\_density(alpha = 0.6) +  
 labs(  
 title = "Density Plot of Combined Diabetes Risk",  
 x = "Combined Probability",  
 y = "Density",  
 fill = "Risk Tier"  
 ) +  
 theme\_minimal()

