PREDICTING DIABETES

Ifenna

2025-03-23

# Predicting Diabetes

This project makes use of various Machine Learning (ML) algorithms to predict outcomes such as **“Diabetic”** or **“Not Diabetic”** based on the following variables:  
Age, Pregnancies, BMI, Glucose, BloodPressure, HbA1c, LDL, HDL, Triglycerides, WaistCircumference, HipCircumference, WHR, FamilyHistory, DietType, Hypertension, MedicationUse.

It is important to note that the target variable is the **Outcome** column, which has been converted to a factor, as stated earlier.

In this project, the following ML algorithms are used:  
- K-Nearest Neighbors (KNN)  
- Decision Tree  
- One Rule (1R)  
- JRIP

Diabetes<- read.csv("diabetes\_dataset.csv")  
Diabetes$Labels<- factor(Diabetes$Outcome, levels = c(0, 1),  
 labels = c("Not Diabetic", "Diabetic"))

## Building Training and Testing Data

library(rsample)

## Warning: package 'rsample' was built under R version 4.4.2

set.seed(1234)  
Knn\_split<- initial\_split(Diabetes[-18], .85, Diabetes$Outcome)  
Train\_data<- training(Knn\_split)  
Test\_data<- testing(Knn\_split)  
Test\_labels<- factor(Test\_data$Outcome, levels = c(0, 1),  
 labels = c("Not Diabetic", "Diabetic"))  
Train\_labels<- factor(Train\_data$Outcome, levels = c(0, 1),  
 labels = c("Not Diabetic", "Diabetic"))

## Predicting with KNN

normalize<- function(x){  
 return((x-min(x))/(max(x)-min(x)))  
}  
normalize(c(0,1,2,3,4))

## [1] 0.00 0.25 0.50 0.75 1.00

Test\_knn<- as.data.frame(lapply(Test\_data[-17], normalize))  
Train\_knn<- as.data.frame(lapply(Train\_data[-17], normalize))  
  
library(class)  
one\_knn<- knn(Train\_knn, Test\_knn, cl = Train\_labels, k = 5)  
one<-table(one\_knn, Test\_labels)  
  
Accuracy<- round((sum(diag(one))/(sum(one)))\*100, digits = 1)  
print(Accuracy)

## [1] 94.7

second\_knn<- knn(Train\_knn, Test\_knn, cl = Train\_labels, k = 15)  
second<-table(second\_knn, Test\_labels)  
Accuracy\_second<- round((sum(diag(second))/(sum(second)))\*100, digits = 1)  
print(Accuracy\_second)

## [1] 96.4

third\_knn<- knn(Train\_knn, Test\_knn, cl = Train\_labels, k = 22)  
third<-table(third\_knn, Test\_labels)  
  
Accuracy\_third<- round((sum(diag(third))/(sum(third)))\*100, digits = 1)  
print(Accuracy\_third)

## [1] 96.2

## Decision Tree

library(C50)

## Warning: package 'C50' was built under R version 4.4.2

set.seed(257)  
Test\_Tree<- Test\_data[-17]  
Train\_Tree<- Train\_data  
Train\_Tree<- Train\_Tree[-17]  
Train\_Tree$labels<- as.factor(Train\_labels)  
Test\_Tree$labels<- as.factor(Test\_labels)  
Model<- C5.0(Train\_Tree$labels~., data= Train\_Tree, method = "class")  
summary(Model)

##   
## Call:  
## C5.0.formula(formula = Train\_Tree$labels ~ ., data = Train\_Tree, method  
## = "class")  
##   
##   
## C5.0 [Release 2.07 GPL Edition] Sun Apr 20 19:54:09 2025  
## -------------------------------  
##   
## Class specified by attribute `outcome'  
##   
## Read 8105 cases (17 attributes) from undefined.data  
##   
## Decision tree:  
##   
## FamilyHistory > 0: Diabetic (2462)  
## FamilyHistory <= 0:  
## :...Glucose <= 140: Not Diabetic (5317)  
## Glucose > 140: Diabetic (326)  
##   
##   
## Evaluation on training data (8105 cases):  
##   
## Decision Tree   
## ----------------   
## Size Errors   
##   
## 3 0( 0.0%) <<  
##   
##   
## (a) (b) <-classified as  
## ---- ----  
## 5317 (a): class Not Diabetic  
## 2788 (b): class Diabetic  
##   
##   
## Attribute usage:  
##   
## 100.00% FamilyHistory  
## 69.62% Glucose  
##   
##   
## Time: 0.1 secs

Test\_Tree$predict<- predict(Model, Test\_Tree, type = "class")  
s<- table(Test\_Tree$predict, Test\_Tree$labels)  
T.Accuracy<- round((sum(diag(s)))/(sum(s))\*100, digits = 1)  
print(T.Accuracy)

## [1] 100

## Plotting Tree

library(partykit)

## Warning: package 'partykit' was built under R version 4.4.2

## Loading required package: grid

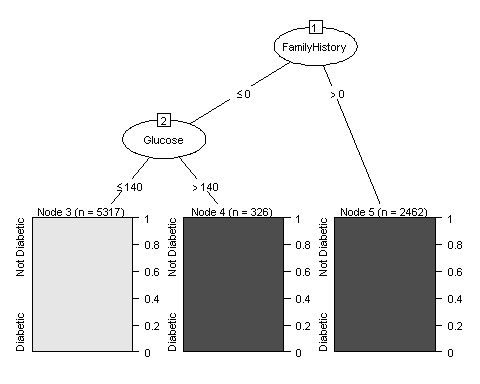
## Loading required package: libcoin

## Warning: package 'libcoin' was built under R version 4.4.2

## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 4.4.2

D.tree<- as.party(Model)  
  
plot(D.tree, gp = gpar(fontsize = 8))

 # Interpretation of the Decision Tree: This decision tree shows how key features—Family History and Glucose Level—are used to predict whether an individual is Diabetic or Not Diabetic.

Interpretation of the Nodes: Root Node (FamilyHistory)

Split Condition: The first and most important factor in the tree is FamilyHistory:

If FamilyHistory ≤ 0 (no family history of diabetes), the person is likely Not Diabetic (Node 3).

If FamilyHistory > 0 (a positive family history), the tree groups individuals as diabetic (Node 5)

for those whose family history is <= 0 the tree then split based on glucose

Second-Level Split: Glucose Level

If Glucose ≤ 140, the individual is classified as Not Diabetic (Node 3).

If Glucose > 140, the individual is classified as Diabetic (Node 4).

## USING ONE RULE AND JRIP

library(RWeka)

## Warning: package 'RWeka' was built under R version 4.4.2

Diabetes\_one<- OneR(Train\_Tree$labels~., data = Train\_Tree)  
summary(Diabetes\_one)

##   
## === Summary ===  
##   
## Correctly Classified Instances 7779 95.9778 %  
## Incorrectly Classified Instances 326 4.0222 %  
## Kappa statistic 0.9083  
## Mean absolute error 0.0402  
## Root mean squared error 0.2006  
## Relative absolute error 8.9119 %  
## Root relative squared error 42.2188 %  
## Total Number of Instances 8105   
##   
## === Confusion Matrix ===  
##   
## a b <-- classified as  
## 5317 0 | a = Not Diabetic  
## 326 2462 | b = Diabetic

Test\_1R<- Test\_Tree[-18]  
Predict\_1R<- predict(Diabetes\_one, Test\_1R)  
R\_One<-table(Predict\_1R, Test\_1R$labels)  
R\_Accuracy<- round((sum(diag(R\_One)))/(sum(R\_One))\*100, digits = 1)  
print(R\_Accuracy)

## [1] 95

### JRIP

Diabetes\_JRIP<- JRip(Train\_Tree$labels~., data = Train\_Tree)  
summary(Diabetes\_JRIP)

##   
## === Summary ===  
##   
## Correctly Classified Instances 8105 100 %  
## Incorrectly Classified Instances 0 0 %  
## Kappa statistic 1   
## Mean absolute error 0   
## Root mean squared error 0   
## Relative absolute error 0 %  
## Root relative squared error 0 %  
## Total Number of Instances 8105   
##   
## === Confusion Matrix ===  
##   
## a b <-- classified as  
## 5317 0 | a = Not Diabetic  
## 0 2788 | b = Diabetic

Jrip\_predict<- predict(Diabetes\_JRIP, Test\_1R)  
JRip\_matrix<-table(Jrip\_predict, Test\_1R$labels)  
JRip\_Accuracy<- round((sum(diag(JRip\_matrix)))/(sum(JRip\_matrix))\*100, digits = 1)  
print(JRip\_Accuracy)

## [1] 100

# Recommendation

All models performed well; however, the Decision Tree and JRIP models achieved the best performance, with a perfect 100% accuracy. This suggests that these models are highly reliable for predicting diabetes.

Additionally, the exploratory analysis highlighted a strong correlation (0.9091) between family history and the target variable, further supporting the model’s decision to prioritize this feature. Based on these findings, it is recommended to:

1. Focus on family history and glucose level as the key factors in determining whether a person is diabetic or not diabetic.
2. Consider leveraging the Decision Tree or JRIP model in real-world applications due to their superior performance.

Explore further feature engineering and dataset expansion to validate and enhance the robustness of the models.