Notebook 3 Ensemble Techniques

Code

Umar Ali-Salaam, Carolline Osei

2022-10-23

Source: https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset (https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset)

This is a dataset based off of 70,000 records of patient data (Heart Related). Columns (13): ID, Age, Height(cm), Weight(kg), Gender, Systolic Blood Pressure (AP_HIGH), Diastolic Blood Pressure (AP LOW), Cholesterol, Glucose, Smoking, Alcohol Intake, Physical Activity, Presence or Absence of cardiovascular disease.

The .csv file needed to be edited a bit in Microsoft Excel before using it in R. I just performed a split column delimiter function around semicolons, to divide the singular column that existed into 13. Each row had 13 variables in 1 column separated by semicolons, the function I ran split it up into 13 columns, making a 70,000 x 13 table.

https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings (https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings)

Visit the website above to better understand Systolic and Diastolic Blood Pressure

Cleaning Data

Hide

```
setwd("C:/Users/ualis/OneDrive/Documents")
```

```
Warning: The working directory was changed to C:/Users/ualis/OneDrive/Documents inside a notebook chunk. The working directory will be reset when the chunk is finished running. Use the knitr root.dir option in the setup chunk to change the working directory for notebook chunks.
```

Hide

```
# Importing Libraries
library(caret)
library(tidyverse)

# Importing data sets
dataset <- read.csv("cardio_train.csv")

# Running a few data exploration functions.
glimpse(dataset)
```

```
Rows: 70,000
Columns: 13
$ ID                <int> 0, 1, 2, 3, 4, 8, 9, 12, 13, 14, 15, 16, 18, 21, 23, 24, 25, 27, 28, 29,
30, 31, 32, 33, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 47, 49, 51, 52, 53, 54, 5~
$ AGE               <int> 18393, 20228, 18857, 17623, 17474, 21914, 22113, 22584, 17668, 19834, 22
530, 18815, 14791, 19809, 14532, 16782, 21296, 16747, 17482, 21755, 19778, 21413, 2304~
$ GENDER            <int> 2, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1, 1, 2,
1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 2,~
$ HEIGHT            <int> 168, 156, 165, 169, 156, 151, 157, 178, 158, 164, 169, 173, 165, 158, 18
1, 172, 170, 158, 154, 162, 163, 157, 158, 156, 170, 153, 156, 159, 166, 169, 155, 169~
$ WEIGHT            <dbl> 62, 85, 64, 82, 56, 67, 93, 95, 71, 68, 80, 60, 60, 78, 95, 112, 75, 52,
68, 56, 83, 69, 90, 45, 68, 65, 59, 78, 66, 74, 105, 71, 60, 73, 82, 55, 95, 70, 72, ~
$ AP_HIGH           <int> 110, 140, 130, 150, 100, 120, 130, 130, 110, 110, 120, 120, 120, 110, 13
0, 120, 130, 110, 100, 120, 120, 130, 145, 110, 150, 130, 130, 120, 120, 130, 120, 140~
$ AP_LOW            <int> 80, 90, 70, 100, 60, 80, 80, 90, 70, 60, 80, 80, 80, 70, 90, 80, 70, 70,
70, 70, 80, 80, 85, 60, 90, 100, 90, 80, 80, 70, 80, 90, 70, 85, 90, 80, 80, 90, 80, ~
$ CHOLESTEROL       <int> 1, 3, 3, 1, 1, 2, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1,
3, 2, 1, 1, 1, 1, 3, 3, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 2, 1, 2, 1, 1, 1, 1,~
$ GLUCOSE           <int> 1, 1, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 2, 1,
1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ SMOKE             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ ALCOHOL           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ PHYSICAL_ACTIVITY <int> 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,~
$ CARDIO_DISEASE    <int> 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1,~
```

Hide

```
summary(dataset)
```

```
       ID              AGE            GENDER         HEIGHT          WEIGHT          AP_HIGH
AP_LOW           CHOLESTEROL       GLUCOSE          SMOKE            ALCOHOL
 Min.   :    0   Min.   :10798   Min.   :1.00   Min.   : 55.0   Min.   : 10.00   Min.   : -150.0
Min.   :  -70.00   Min.   :1.000   Min.   :1.000   Min.   :0.00000   Min.   :0.00000
 1st Qu.:25007   1st Qu.:17664   1st Qu.:1.00   1st Qu.:159.0   1st Qu.: 65.00   1st Qu.:  120.0
1st Qu.:   80.00   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:0.00000
 Median :50002   Median :19703   Median :1.00   Median :165.0   Median : 72.00   Median :  120.0
Median :   80.00   Median :1.000   Median :1.000   Median :0.00000   Median :0.00000
 Mean   :49972   Mean   :19469   Mean   :1.35   Mean   :164.4   Mean   : 74.21   Mean   :  128.8
Mean   :   96.63   Mean   :1.367   Mean   :1.226   Mean   :0.08813   Mean   :0.05377
 3rd Qu.:74889   3rd Qu.:21327   3rd Qu.:2.00   3rd Qu.:170.0   3rd Qu.: 82.00   3rd Qu.:  140.0
3rd Qu.:   90.00   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:0.00000
 Max.   :99999   Max.   :23713   Max.   :2.00   Max.   :250.0   Max.   :200.00   Max.   :16020.0
Max.   :11000.00   Max.   :3.000   Max.   :3.000   Max.   :1.00000   Max.   :1.00000
 PHYSICAL_ACTIVITY CARDIO_DISEASE
 Min.   :0.0000    Min.   :0.0000
 1st Qu.:1.0000    1st Qu.:0.0000
 Median :1.0000    Median :0.0000
 Mean   :0.8037    Mean   :0.4997
 3rd Qu.:1.0000    3rd Qu.:1.0000
 Max.   :1.0000    Max.   :1.0000
```

Hide

```
# Converting the target variable into factor levels
dataset$CARDIO_DISEASE = as.factor(dataset$CARDIO_DISEASE)
```

Splitting into train and test

Hide

```
split = sample.split(dataset$CARDIO_DISEASE, SplitRatio = 0.8)
```

```
Error in sample.split(dataset$CARDIO_DISEASE, SplitRatio = 0.8) :
  could not find function "sample.split"
```

Creating Baseline Decission Tree

Hide

```
# specifying the CV technique which will be passed into the train() function later and number para
meter is the "k" in K-fold cross validation
train_control = trainControl(method = "cv", number = 5, search = "grid")

## Customsing the tuning grid (ridge regression has alpha = 0)
classification_Tree_Grid =  expand.grid(maxdepth = c(1,3,5,7,9))

set.seed(50)

# training a Regression model while tuning parameters (Method = "rpart")
model = train(CARDIO_DISEASE~., data = training_set, method = "rpart2", trControl = train_control,
tuneGrid = classification_Tree_Grid)

# summarising the results
print(model)
```

Making Predictions on Baseline Model

Hide

```


# Using baseline model to make predictions on test set
pred_y = predict(model, test_set)

# Confusion Matrix
confusionMatrix(dataset = pred_y, test_set$CARDIO_DISEASE)
```

XGboost

Hide

```
install.packages('xgboost')
library(xgboost)

train_label <- ifelse(training_set$CARDIO_DISEASE==1, 1, 0)
train_matrix <- dataset.matrix(training_set[, -31])
model <- xgboost(dataset = train_matrix, label = train_label, nrounds = 100, objective='binary:log
istic')
```

Hide

```
test_label <- ifelse(test_set$CARDIO_DISEASE==1, 1, 0)
test_matrix <- dataset.matrix(test_set[, -31])

probs <- predict(model, test_matrix)
pred <- ifelse(probs > 0.5, 1, 0)

acc_xg <- mean(pred==test_label)
cc_xg <- mcc(pred, test_label)

print(paste("Accuracy: ", acc_xg))
print(paste("Correlation Coefficient: ", cc_xg))
```

Random Forest

Hide

```
library(randomForest)
library(caret)
library(e1071)

# Define the control
trControl <- trainControl(method = "cv",
    number = 10,
    search = "grid")

set.seed(1234)

# Run the model
rf_default <- train(CARDIO_DISEASE~.,
    data = training_set,
    method = "rf",
    metric = "Accuracy",
    trControl = trControl)

# Print the results
print(rf_default)

# Testing 20 values
set.seed(1234)

tuneGrid <- expand.grid(.mtry = c(1: 10))
rf_mtry <- train(CARDIO_DISEASE~.,
    data = training_set,
    method = "rf",
    metric = "Accuracy",
    tuneGrid = tuneGrid,
    trControl = trControl,
    importance = TRUE,
    nodesize = 14,
    ntree = 300)
print(rf_mtry)
```

Support Vector Machine (SVM Classification)

Hide

```
install.packages('e1071')
library(e1071)

svm_c <- svm(CARDIO_DISEASE~., data=training_set, kernel="linear", cost=10, scale=TRUE)
summary(svm_c)
```

Evaluating and Plotting Results

In this line of code veiwers can use the following R code to make evaluation based on their

Hide

```
pred <- predict(svm_c, newdata = test_set)
table(pred, test_set$CARDIO_DISEASE)
mean(pred==test_set$CARDIO_DISEASE)
```

following the evaluation users can visualize the output by plotting the support vectors.

Hide

```
plot(svm_c, test_set, WEIGHT - CHOLESTEROL, slice - list(CARDIO_DISEASE = 1, CARDIO_DISEASE = 0))
```