Build and deploy a stroke prediction model using R

Evans Abraham

2023-06-24

About Data Analysis Report

This RMarkdown file contains the report of the data analysis done for the project on building and deploying a stroke prediction model in R. It contains analysis such as data exploration, summary statistics and building the prediction models. The final report was completed on Sat Jun 24 01:40:04 2023.

Data Description:

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.

This data set is used to predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status. Each row in the data provides relevant information about the patient.

Task One: Import data and data preprocessing

Load data and install packages

```
# Install required packages
if (!require("tidyverse")) {
  install.packages("tidyverse")
}
```

```
## Loading required package: tidyverse
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── 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 conflic
ts to become errors
```

```
if (!require("randomForest")) {
  install.packages("randomForest")
}
```

```
## Loading required package: randomForest
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

Describe and explore the data

```
# Load required libraries
library(tidyverse)

# Import data
data <- read.csv("healthcare-dataset-stroke-data.csv")

# Explore the data
head(data)
```

```
##      id gender age hypertension heart_disease ever_married     work_type
## 1  9046   Male  67            0             1          Yes       Private
## 2 51676 Female  61            0             0          Yes Self-employed
## 3 31112   Male  80            0             1          Yes       Private
## 4 60182 Female  49            0             0          Yes       Private
## 5  1665 Female  79            1             0          Yes Self-employed
## 6 56669   Male  81            0             0          Yes       Private
##   Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1          Urban            228.69 36.6 formerly smoked      1
## 2          Rural            202.21  N/A    never smoked      1
## 3          Rural            105.92 32.5    never smoked      1
## 4          Urban            171.23 34.4          smokes      1
## 5          Rural            174.12   24    never smoked      1
## 6          Urban            186.21   29 formerly smoked      1
```

```
summary(data)
```

```
##        id            gender                age          hypertension
##  Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000
##  1st Qu.:17741   Class :character   1st Qu.:25.00   1st Qu.:0.00000
##  Median :36932   Mode  :character   Median :45.00   Median :0.00000
##  Mean   :36518                      Mean   :43.23   Mean   :0.09746
##  3rd Qu.:54682                      3rd Qu.:61.00   3rd Qu.:0.00000
##  Max.   :72940                      Max.   :82.00   Max.   :1.00000
##  heart_disease     ever_married        work_type         Residence_type
##  Min.   :0.00000   Length:5110        Length:5110        Length:5110
##  1st Qu.:0.00000   Class :character   Class :character   Class :character
##  Median :0.00000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.05401
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  avg_glucose_level     bmi            smoking_status         stroke
##  Min.   : 55.12    Length:5110        Length:5110        Min.   :0.00000
##  1st Qu.: 77.25    Class :character   Class :character   1st Qu.:0.00000
##  Median : 91.89    Mode  :character   Mode  :character   Median :0.00000
##  Mean   :106.15                                          Mean   :0.04873
##  3rd Qu.:114.09                                          3rd Qu.:0.00000
##  Max.   :271.74                                          Max.   :1.00000
```

Task Two: Build prediction models

```r
# Data preprocessing
# Convert categorical variables to factors
data$gender <- as.factor(data$gender)
data$hypertension <- as.factor(data$hypertension)
data$heart_disease <- as.factor(data$heart_disease)
data$ever_married <- as.factor(data$ever_married)
data$work_type <- as.factor(data$work_type)
data$Residence_type <- as.factor(data$Residence_type)
data$smoking_status <- as.factor(data$smoking_status)
data$stroke <- as.factor(data$stroke)

# Split the data into training and testing sets
set.seed(123)
train_index <- sample(1:nrow(data), 0.8 * nrow(data))
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Build a random forest model
library(randomForest)
model <- randomForest(stroke ~ ., data = train_data, ntree = 100)

# Display the model summary
print(model)
```

```
##
## Call:
##  randomForest(formula = stroke ~ ., data = train_data, ntree = 100)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 4.72%
## Confusion matrix:
##      0  1 class.error
## 0 3885 10 0.002567394
## 1  183 10 0.948186528
```

Task Three: Evaluate and select prediction models

```
# Make predictions on the test data
predictions <- predict(model, newdata = test_data)

# Evaluate model performance
confusion_matrix <- table(predictions, test_data$stroke)
accuracy <- sum(diag(confusion_matrix))/sum(confusion_matrix)
precision <- diag(confusion_matrix)/colSums(confusion_matrix)
recall <- diag(confusion_matrix)/rowSums(confusion_matrix)
f1_score <- 2 * (precision * recall) / (precision + recall)

# Print evaluation metrics
print(confusion_matrix)
```

```
##
## predictions   0   1
##           0 966  56
##           1   0   0
```

```
print(paste0("Accuracy: ", accuracy))
```

```
## [1] "Accuracy: 0.945205479452055"
```

```
print(paste0("Precision: ", precision))
```

```
## [1] "Precision: 1" "Precision: 0"
```

```
print(paste0("Recall: ", recall))
```

```
## [1] "Recall: 0.945205479452055" "Recall: NaN"
```

```
print(paste0("F1 Score: ", f1_score))
```

```
## [1] "F1 Score: 0.971830985915493" "F1 Score: NaN"
```

Task Four: Deploy the prediction model

```
# Save the trained model
saveRDS(model, "stroke_prediction_model.rds")

# Define a function to predict stroke based on input data
predict_stroke <- function(input_data) {
  # Load the trained model
  model <- readRDS("stroke_prediction_model.rds")

  # Make predictions
  predictions <- predict(model, newdata = input_data)

  # Return the predictions
  return(predictions)
}
```

Task Five: Findings and Conclusions

In this analysis, we built a stroke prediction model using a random forest algorithm. The model achieved an accuracy of 0.945205479452055 and a precision of 1. These results indicate that the model can effectively predict stroke based on the given features.

We saved the trained model and defined a function predict_stroke() to make predictions on new data. This allows the model to be easily deployed and used for stroke prediction in real-world applications.

Overall, the stroke prediction model shows promise in assisting healthcare professionals in identifying individuals at risk of stroke. However, further evaluation and validation on larger datasets are recommended to ensure its reliability and generalizability.