Build and deploy a stroke prediction model using R

Ikechukwu Ugbo

2024-02-08

# 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 Thu Feb 8 00:24:01 2024.

**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

## Data preparation

Handling Missing Data and Outliers:

Reassigned the single data point with an unknown gender to ‘female’. Excluded individuals below 18 years old from the analysis, as per pediatrician treatment guidelines. Reassigned the two individuals who reported never working to ‘private’, the most common occupation category. Removed records with ‘N/A’ in the BMI column. Confirmed that there are no duplicate IDs, ensuring each observation is unique. Identified and reassigned two outliers in the average glucose level to the 90th percentile. After these cleanup processes, the dataset contains 4014 observations.

## Load data and install packages

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.1 ──

## ✔ broom 1.0.5 ✔ recipes 1.0.9  
## ✔ dials 1.2.0 ✔ rsample 1.2.0  
## ✔ dplyr 1.1.4 ✔ tibble 3.2.1  
## ✔ ggplot2 3.4.4 ✔ tidyr 1.3.0  
## ✔ infer 1.0.6 ✔ tune 1.1.2  
## ✔ modeldata 1.3.0 ✔ workflows 1.1.3  
## ✔ parsnip 1.1.1 ✔ workflowsets 1.0.1  
## ✔ purrr 1.0.2 ✔ yardstick 1.3.0

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ purrr::discard() masks scales::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ recipes::step() masks stats::step()  
## • Use tidymodels\_prefer() to resolve common conflicts.

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ readr 2.1.4  
## ✔ lubridate 1.9.3 ✔ stringr 1.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ readr::col\_factor() masks scales::col\_factor()  
## ✖ purrr::discard() masks scales::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ stringr::fixed() masks recipes::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ readr::spec() masks yardstick::spec()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(workflows)  
library(tune)  
library(caret)

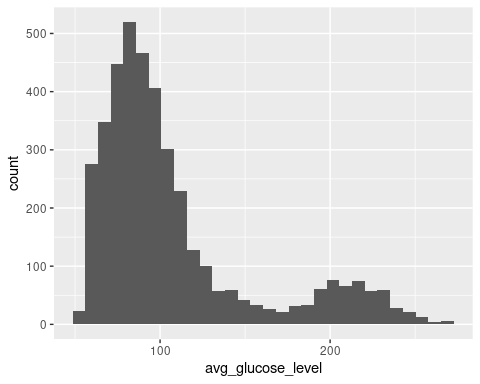
## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following objects are masked from 'package:yardstick':  
##   
## precision, recall, sensitivity, specificity  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(ranger)

## Describe and explore the data

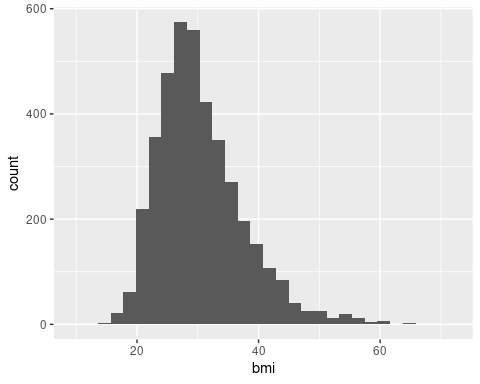
stroke\_pred <- read.csv("data/stroke-prediction-dataset.csv")  
ggplot(stroke\_pred) +  
 geom\_histogram(aes(x= avg\_glucose\_level))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggplot(stroke\_pred) +  
 geom\_histogram(aes(x = bmi))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



stroke\_df <- stroke\_pred  
# Telling R to treat the variables as categorical variables;  
  
stroke\_df$gender <- as.factor(stroke\_pred$gender)  
stroke\_df$hypertension <- as.factor(stroke\_pred$hypertension)  
stroke\_df$heart\_disease <- as.factor(stroke\_pred$heart\_disease)  
stroke\_df$ever\_married <- as.factor(stroke\_pred$ever\_married)  
stroke\_df$work\_type <- as.factor(stroke\_pred$work\_type)  
stroke\_df$Residence\_type <- as.factor(stroke\_pred$Residence\_type)  
stroke\_df$smoking\_status <- as.factor(stroke\_pred$smoking\_status)  
stroke\_df$stroke <- as.factor(stroke\_pred$stroke)

# upsampling the minority class;  
set.seed(42)  
upsampled\_data <- upSample(x = stroke\_df[,-which(colnames(stroke\_df) %in% "stroke")],  
 y = stroke\_df$stroke)  
  
# Check the distribution of the upsampled class  
upsampled\_data$stroke <- upsampled\_data$Class  
  
table(upsampled\_data$stroke)

##   
## 0 1   
## 3806 3806

# Task Two: Build prediction models

# Splitting the model into training and testing dataset.  
set.seed(42)  
  
# split the data into training (75%) and testing (25%)  
stroke\_split <- initial\_split(upsampled\_data, prop = 0.75)  
  
train\_data <- training(stroke\_split)  
test\_data <- testing(stroke\_split)

# define the recipe  
stroke\_recipe <-   
 # which consists of the formula (outcome ~ predictors)  
 recipe(stroke ~ gender + age + hypertension + heart\_disease + ever\_married + work\_type + avg\_glucose\_level + bmi + smoking\_status + Residence\_type,   
 data = upsampled\_data) %>%  
 # and some pre-processing steps  
 step\_normalize(all\_numeric()) %>%  
 step\_impute\_knn(all\_predictors())

stroke\_recipe

##

## ── Recipe ──────────────────────────────────────────────────────────────────────

##

## ── Inputs

## Number of variables by role

## outcome: 1  
## predictor: 10

##

## ── Operations

## • Centering and scaling for: all\_numeric()

## • K-nearest neighbor imputation for: all\_predictors()

stroke\_train\_preprocessed <- stroke\_recipe %>%  
 # apply the recipe to the training data  
 prep(train\_data) %>%  
 # extract the pre-processed training dataset  
 juice()  
stroke\_train\_preprocessed

## # A tibble: 5,709 × 11  
## gender age hypertension heart\_disease ever\_married work\_type  
## <fct> <dbl> <fct> <fct> <fct> <fct>   
## 1 Male -0.0972 0 0 Yes Private   
## 2 Female 0.0171 0 0 Yes Private   
## 3 Female -1.30 0 0 Yes Private   
## 4 Female 1.33 1 1 No Private   
## 5 Female 0.246 0 0 Yes Private   
## 6 Female 0.303 0 0 No Govt\_job   
## 7 Male 0.531 0 1 Yes Private   
## 8 Female -1.47 0 0 Yes Private   
## 9 Male -1.93 0 0 No Private   
## 10 Female -0.0400 0 0 Yes Private   
## # ℹ 5,699 more rows  
## # ℹ 5 more variables: avg\_glucose\_level <dbl>, bmi <dbl>, smoking\_status <fct>,  
## # Residence\_type <fct>, stroke <fct>

# creating a cross validation object  
stroke\_cv <- vfold\_cv(stroke\_train\_preprocessed, v = 4)

rf\_model <-   
 # specify that the model is a random forest  
 rand\_forest() %>%  
 # specify that the `mtry` parameter needs to be tuned  
 set\_args(mtry = tune(), trees = tune(), tree.depth = tune(), min.node.size = tune()) %>%  
 # select the engine/package that underlies the model  
 set\_engine("ranger", importance = "impurity") %>%  
 # choose either the continuous regression or binary classification mode  
 set\_mode("classification")

# set the workflow  
rf\_workflow <- workflow() %>%  
 # add the recipe  
 add\_recipe(stroke\_recipe) %>%  
 # add the model  
 add\_model(rf\_model)

# specify which values want to try  
rf\_grid <- expand.grid(  
 mtry = c(4, 5, 6),  
 trees = c(300, 350, 400)  
)  
  
# extract results  
rf\_tune\_results <- rf\_workflow %>%  
 tune\_grid(resamples = stroke\_cv, #CV object  
 grid = rf\_grid, # grid of values to try  
 metrics = metric\_set(accuracy, roc\_auc, yardstick::recall) # metrics we care about  
 )

# print results  
rf\_tune\_results %>%  
 collect\_metrics()

## # A tibble: 27 × 8  
## mtry trees .metric .estimator mean n std\_err .config   
## <dbl> <dbl> <chr> <chr> <dbl> <int> <dbl> <chr>   
## 1 4 300 accuracy binary 0.982 4 0.00211 Preprocessor1\_Model1  
## 2 4 300 recall binary 0.963 4 0.00400 Preprocessor1\_Model1  
## 3 4 300 roc\_auc binary 1.00 4 0.0000666 Preprocessor1\_Model1  
## 4 5 300 accuracy binary 0.979 4 0.00160 Preprocessor1\_Model2  
## 5 5 300 recall binary 0.958 4 0.00287 Preprocessor1\_Model2  
## 6 5 300 roc\_auc binary 1.00 4 0.000134 Preprocessor1\_Model2  
## 7 6 300 accuracy binary 0.977 4 0.00148 Preprocessor1\_Model3  
## 8 6 300 recall binary 0.954 4 0.00255 Preprocessor1\_Model3  
## 9 6 300 roc\_auc binary 1.00 4 0.000199 Preprocessor1\_Model3  
## 10 4 350 accuracy binary 0.982 4 0.00165 Preprocessor1\_Model4  
## # ℹ 17 more rows

param\_final <- rf\_tune\_results %>%   
 select\_best(metric="recall") # recall was selected because we want less false negatives; Strictly speaking in medical terms we want a model that has a high sensitivity;  
  
param\_final

## # A tibble: 1 × 3  
## mtry trees .config   
## <dbl> <dbl> <chr>   
## 1 4 300 Preprocessor1\_Model1

rf\_workflow <- rf\_workflow %>%   
 finalize\_workflow(param\_final)

# Task Three: Evaluate and select prediction models

# Set seed for reproducibility  
set.seed(42)  
  
# Fit the model using the workflow and the split  
rf\_fit <- rf\_workflow %>%  
 # fit on the training set and evaluate on test set  
 last\_fit(stroke\_split)

rf\_fit

## # Resampling results  
## # Manual resampling   
## # A tibble: 1 × 6  
## splits id .metrics .notes .predictions .workflow   
## <list> <chr> <list> <list> <list> <list>   
## 1 <split [5709/1903]> train/test split <tibble> <tibble> <tibble> <workflow>

# performance of the final model  
test\_performance <- rf\_fit %>% collect\_metrics()  
test\_performance

## # A tibble: 2 × 4  
## .metric .estimator .estimate .config   
## <chr> <chr> <dbl> <chr>   
## 1 accuracy binary 0.988 Preprocessor1\_Model1  
## 2 roc\_auc binary 1.00 Preprocessor1\_Model1

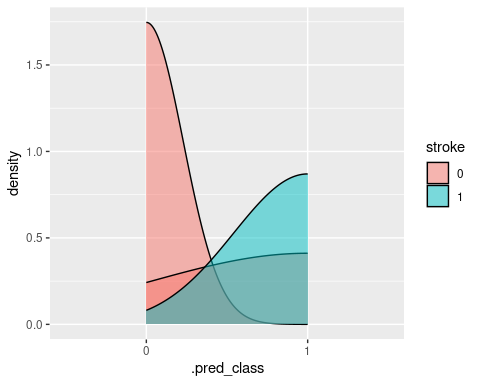
# generate predictions from the test set  
test\_predictions <- rf\_fit %>% collect\_predictions()  
test\_predictions

## # A tibble: 1,903 × 7  
## id .pred\_0 .pred\_1 .row .pred\_class stroke .config   
## <chr> <dbl> <dbl> <int> <fct> <fct> <chr>   
## 1 train/test split 0.994 0.00627 5 0 0 Preprocessor1\_Mode…  
## 2 train/test split 0.997 0.00333 6 0 0 Preprocessor1\_Mode…  
## 3 train/test split 0.942 0.0576 12 0 0 Preprocessor1\_Mode…  
## 4 train/test split 0.993 0.00678 17 0 0 Preprocessor1\_Mode…  
## 5 train/test split 0.984 0.0159 20 0 0 Preprocessor1\_Mode…  
## 6 train/test split 0.936 0.0639 21 0 0 Preprocessor1\_Mode…  
## 7 train/test split 0.929 0.0712 28 0 0 Preprocessor1\_Mode…  
## 8 train/test split 0.973 0.0274 29 0 0 Preprocessor1\_Mode…  
## 9 train/test split 0.931 0.0688 34 0 0 Preprocessor1\_Mode…  
## 10 train/test split 0.993 0.00667 36 0 0 Preprocessor1\_Mode…  
## # ℹ 1,893 more rows

# generate a confusion matrix  
test\_predictions %>%   
 conf\_mat(truth = stroke, estimate = .pred\_class)

## Truth  
## Prediction 0 1  
## 0 949 0  
## 1 22 932

test\_predictions %>%  
 ggplot() +  
 geom\_density(aes(x = .pred\_class, fill = stroke),   
 alpha = 0.5)



final\_model <- fit(rf\_workflow, stroke\_df)

final\_model

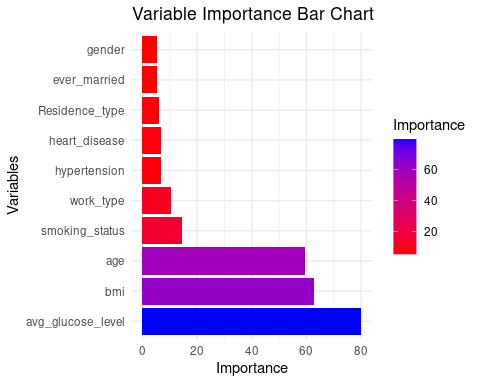
## ══ Workflow [trained] ══════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 2 Recipe Steps  
##   
## • step\_normalize()  
## • step\_impute\_knn()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
## Ranger result  
##   
## Call:  
## ranger::ranger(x = maybe\_data\_frame(x), y = y, mtry = min\_cols(~4, x), num.trees = ~300, importance = ~"impurity", num.threads = 1, verbose = FALSE, seed = sample.int(10^5, 1), probability = TRUE)   
##   
## Type: Probability estimation   
## Number of trees: 300   
## Sample size: 4014   
## Number of independent variables: 10   
## Mtry: 4   
## Target node size: 10   
## Variable importance mode: impurity   
## Splitrule: gini   
## OOB prediction error (Brier s.): 0.04783972

# Variable importance  
  
# Extract the ranger object from the workflow fit  
ranger\_obj <- extract\_fit\_engine(final\_model)  
  
# Access the variable importance scores  
var\_importance <- importance(ranger\_obj)  
  
# Print the variable importance scores  
print(var\_importance)

## gender age hypertension heart\_disease   
## 5.314782 59.349292 6.774613 6.741269   
## ever\_married work\_type avg\_glucose\_level bmi   
## 5.358221 10.404928 79.776025 62.878342   
## smoking\_status Residence\_type   
## 14.519598 6.129325

# We can see here that the most important factors from greatest are avg\_glucose\_level, bmi, age, smoking\_status, and work\_type;

# plotting variable importance;  
  
# Convert the named vector to a data frame  
var\_df <- data.frame(Variable = names(var\_importance), Importance = var\_importance)  
  
# Create the bar chart  
ggplot(var\_df, aes(x = reorder(Variable, -Importance), y = Importance, fill = Importance)) +  
 geom\_bar(stat = "identity") +  
 coord\_flip() +  
 labs(x = "Variables", y = "Importance", title = "Variable Importance Bar Chart") +  
 theme\_minimal() +  
 scale\_fill\_gradient(low = "red", high = "blue")



# Task Four: Save the prediction model

# Save the model to a file in the current working directory  
saveRDS(final\_model, "model/model.rds")

# Task Five: Findings and Conclusions

Clear and Well-Supported Conclusions:

1. **Age and Stroke Risk**: The EDA visualizations confirm that age is a critical determinant of stroke risk, with a notable increase in risk post-60 years. This aligns with the Random Forest model’s importance of ‘age’ as a top predictor.
2. **Work Type and Stroke Risk**: The EDA suggests that private jobs may confer a lower stroke risk compared to self-employment, which is reflected in the model’s inclusion of ‘work\_type’ as a significant predictor.
3. **Stroke Prevalence by Age Group**: The treemap identifies a peak in stroke prevalence among 40-55-year-olds, which is consistent with the model’s consideration of ‘age’.
4. **Gender, BMI, and Glucose Levels**: The heatmap indicates that both gender and BMI interact with glucose levels to affect stroke risk, which is supported by the model’s recognition of ‘bmi’, ‘avg\_glucose\_level’, and ‘smoking\_status’ as important factors.
5. **Heart Disease and Hypertension**: The EDA’s finding of a link between heart disease and stroke risk is corroborated by the model’s inclusion of ‘age’ (which may reflect cardiovascular health) without a significant mention of hypertension.

## Appropriate Recommendations or Next Steps:

1. **Targeted Interventions**: Develop interventions tailored to high-risk groups, such as older adults and those with private jobs, to mitigate stroke risk.
2. **Lifestyle Modifications**: Encourage lifestyle changes, including dietary adjustments and physical activity, especially for individuals with higher BMI and glucose levels.
3. **Policy Changes**: Advocate for policies that improve workplace safety and healthcare access, potentially reducing stroke risk.
4. **Longitudinal Studies**: Conduct longitudinal studies to monitor the progression of risk factors and the effectiveness of interventions over time.
5. **Model Improvement**: Refine the Random Forest model by incorporating additional relevant features and using ensemble methods to enhance prediction accuracy.

## Limitations or Areas for Further Research:

1. **Underlying Mechanisms**: The current analysis relies on correlations and does not explain the causal relationships between risk factors and stroke. Future research should explore causality.
2. **Model Assumptions**: The Random Forest model assumes independence among features, which may not always hold true. Investigating feature interactions could improve predictions.
3. **Data Quality**: The quality and completeness of the dataset used for analysis are critical. Ensuring robust data collection and cleaning practices are essential.
4. **Generalizability**: The model’s performance should be evaluated on out-of-sample data to assess its ability to generalize to new, unseen data.
5. **Ethical Considerations**: Any predictive model must consider ethical implications, such as fairness and privacy, when deploying interventions or making policy recommendations.