Build and deploy a stroke prediction model using R

Coursera Project Network

Stroke represents a cerebrovascular event that occurs when blood flow to the brain is disrupted either through blockage (ischemic stroke) or bleeding (hemorrhagic stroke). According to the World Health Organization stroke ranks as the second leading cause of death globally accounting for approximately 11% of total deaths. The condition's severity stems from its immediate impact on neural tissue where each minute of delayed treatment results in the loss of approximately 1.9 million neurons leading to potential long-term disabilities or death. The profound impact of stroke extends beyond individual health outcomes to create significant socioeconomic burdens on healthcare systems worldwide.

Modern stroke management has evolved significantly through the integration of artificial intelligence clinical practice. These advanced systems enable rapid diagnosis through automated imaging analysis enhanced risk stratification and personalized treatment planning. AI algorithms have demonstrated remarkable accuracy in analyzing CT and MRI scans often matching or exceeding human diagnostic capabilities in detecting early stroke signs. The implementation of AI-driven diagnostic tools has become particularly crucial in time-sensitive decision-making processes where early intervention can significantly improve patient outcomes and reduce the likelihood of permanent disability.

This research project focused on developing and implementing machine learning models for stroke prediction using a comprehensive dataset of 5110 patients with multiple risk factors. The study employed three distinct machine learning approaches: logistic regression random forest and support vector machine (SVM) models. Through data processing and handling of class imbalance using ROSE technique the logistic regression model emerged as the superior predictor with 82.7% accuracy and 83.2% sensitivity. The model demonstrated good performance in analyzing key risk factors including age glucose levels and cardiovascular comorbidities. When tested on specific patient profiles the system achieved an 80.2% accuracy in risk assessment.

---

title: "Build and deploy a stroke prediction model using R"

date: "`r Sys.Date()`"

output: html_document

author: "Naya James Mbabila"

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 `r date()`.

Naya James Mbabila

coursera LinkedIn Gmail

Task One: Import data and data preprocessing

Load data and install packages

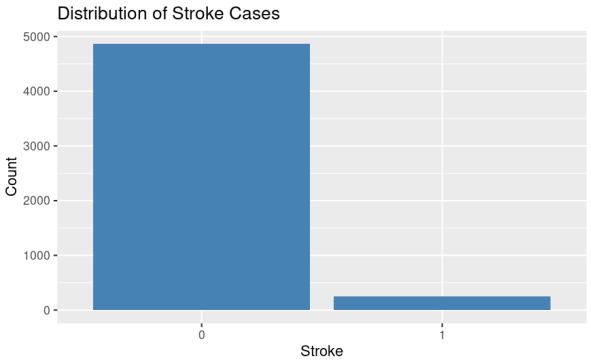
```{r}
# Install required packages
if (!require("tidyverse")) install.packages("tidyverse")
if (!require("caret")) install.packages("caret")
if (!require("randomForest")) install.packages("randomForest")
if (!require("e1071")) install.packages("e1071")
if (!require("ROSE")) install.packages("ROSE")
if (!require("corrplot")) install.packages("corrplot")
# Load libraries
library(tidyverse)
library(caret)
library(randomForest)
library(e1071)
library(ROSE)
library(corrplot)
# Read the dataset
stroke_data <- read.csv("healthcare-dataset-stroke-data.csv")
```

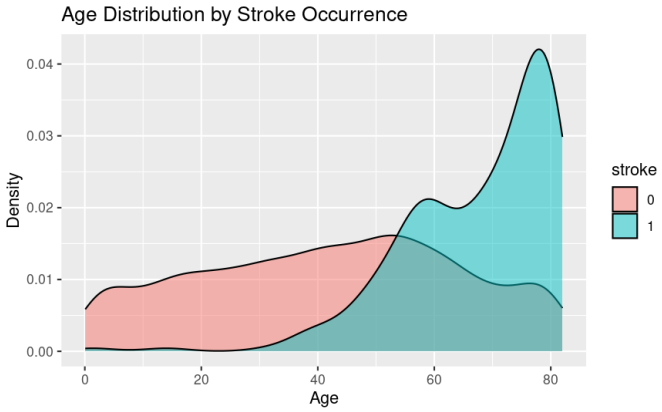
Describe and explore the data

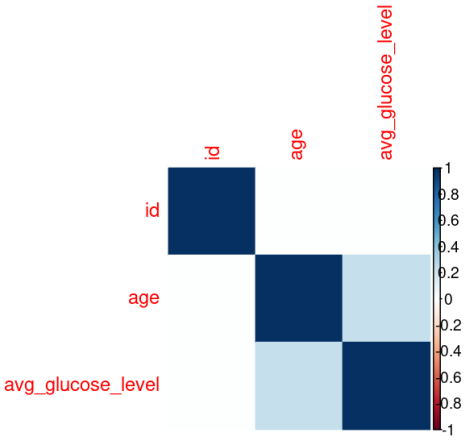
```{r}
# Display the structure of the dataset
str(stroke_data)
# Summary statistics
summary(stroke_data)
# Check for missing values
colSums(is.na(stroke_data))
# Data preprocessing
stroke_data <- stroke_data %>%
  # Convert categorical variables to factors
  mutate(
    gender = as.factor(gender),
    hypertension = as.factor(hypertension),
    heart_disease = as.factor(heart_disease),
    ever_married = as.factor(ever_married),
    work_type = as.factor(work_type),
    Residence_type = as.factor(Residence_type),
    smoking_status = as.factor(smoking_status),
    stroke = as.factor(stroke)
  )
# Handle missing values in bmi
stroke_data$bmi <- ifelse(is.na(stroke_data$bmi),
                          mean(stroke_data$bmi, na.rm = TRUE),
                          stroke_data$bmi)
# Visualize the distribution of stroke cases
ggplot(stroke_data, aes(x = stroke)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Distribution of Stroke Cases",
       x = "Stroke",
       y = "Count")
# Age distribution by stroke occurrence
ggplot(stroke_data, aes(x = age, fill = stroke)) +
  geom_density(alpha = 0.5) +
  labs(title = "Age Distribution by Stroke Occurrence",
       x = "Age",
       y = "Density")
# Correlation plot for numerical variables
numeric_vars <- stroke_data %>%
  select_if(is.numeric)
corrplot(cor(numeric_vars), method = "color")
```

Results

```
'data.frame':           5110 obs. of  12 variables:
 $ id               : int  9046 51676 31112 60182 1665
56669 53882 10434 27419 60491 ...
 $ gender           : chr  "Male" "Female" "Male" "Female"
...
 $ age              : num  67 61 80 49 79 81 74 69 59 78
...
 $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
 $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
 $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ work_type        : chr  "Private" "Self-employed"
"Private" "Private" ...
 $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban"
...
 $ avg_glucose_level: num  229 202 106 171 174 ...
 $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
 $ smoking_status   : chr  "formerly smoked" "never
smoked" "never smoked" "smokes" ...
 $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
       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
 Min.   :0.00000   Length:5110        Length:5110
 1st Qu.:0.00000   Class :character   Class :character
 Median :0.00000   Mode  :character   Mode  :character
 Mean   :0.05401
 3rd Qu.:0.00000
 Max.   :1.00000
 Residence_type     avg_glucose_level     bmi
 Length:5110        Min.   : 55.12    Length:5110
 Class :character   1st Qu.: 77.25    Class :character
 Mode  :character   Median : 91.89    Mode  :character
                    Mean   :106.15
                    3rd Qu.:114.09
                    Max.   :271.74
 smoking_status         stroke
 Length:5110        Min.   :0.00000
 Class :character   1st Qu.:0.00000
 Mode  :character   Median :0.00000
                    Mean   :0.04873
                    3rd Qu.:0.00000
                    Max.   :1.00000
             id            gender               age
hypertension
              0                 0                 0
0
    heart_disease      ever_married         work_type
Residence_type
              0                 0                 0
0
avg_glucose_level               bmi    smoking_status
stroke
              0                 0                 0
0
```







Task Two: Build prediction models

```{r}
# Convert outcome variable and other categorical variables to factors, if not already
stroke_data <- stroke_data %>%
  mutate(across(c(gender, hypertension, heart_disease, ever_married,
                  work_type, Residence_type, smoking_status, stroke), as.factor))

# Split the data into training and testing sets
set.seed(123)
train_index <- createDataPartition(stroke_data$stroke, p = 0.8, list = FALSE)
train_data <- stroke_data[train_index, ]
test_data <- stroke_data[-train_index, ]

# Ensure no character variables in train_data for compatibility with ROSE
train_data[] <- lapply(train_data, function(x) if(is.character(x)) as.factor(x) else x)

# Handle class imbalance using ROSE with only valid variable types
balanced_train <- ROSE(stroke ~ ., data = train_data)$data

# Remove high-cardinality categorical variables, if any
# For example, remove `id` or any categorical variable with more than 53 levels
balanced_train <- balanced_train %>% select(-id) # Remove the `id` column if present

high_cardinality_vars <- sapply(balanced_train, function(x) is.factor(x) && nlevels(x) > 53)
balanced_train <- balanced_train %>% select(-which(high_cardinality_vars))

# 1. Logistic Regression
logistic_model <- glm(stroke ~ ., data = balanced_train, family = "binomial")

# 2. Random Forest
rf_model <- randomForest(stroke ~ ., data = balanced_train, ntree = 500)

# 3. Support Vector Machine
svm_model <- svm(stroke ~ ., data = balanced_train, kernel = "radial")

# Confirm models were built
logistic_model
rf_model
svm_model

```

RESULTS

```
Call:  glm(formula = stroke ~ ., family =
"binomial", data = balanced_train)

Coefficients:
                  (Intercept)
genderMale
                    -3.451406                    -
0.004040
                          age
hypertension1
                     0.067891
0.390606
              heart_disease1
ever_marriedYes
                     0.552122
0.132357
           work_typeGovt_job
work_typeNever_worked
                    -0.839650                    -
11.844286
           work_typePrivate      work_typeSelf-
employed
                    -0.587617                    -
0.964119
         Residence_typeUrban
avg_glucose_level
                     0.035504
0.002265
smoking_statusnever smoked
smoking_statussmokes
                    -0.546985
0.055018
   smoking_statusUnknown
                    -0.298384

Degrees of Freedom: 4088 Total (i.e. Null);  4074
Residual
Null Deviance:           5668
Residual Deviance: 4085        AIC: 4115

Call:
 randomForest(formula = stroke ~ ., data =
balanced_train, ntree = 500)
               Type of random forest:
classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 18.44%
Confusion matrix:
     0    1 class.error
0 1586  490   0.2360308
1  264 1749   0.1311475
Call:
svm(formula = stroke ~ ., data = balanced_train,
kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  2235
```

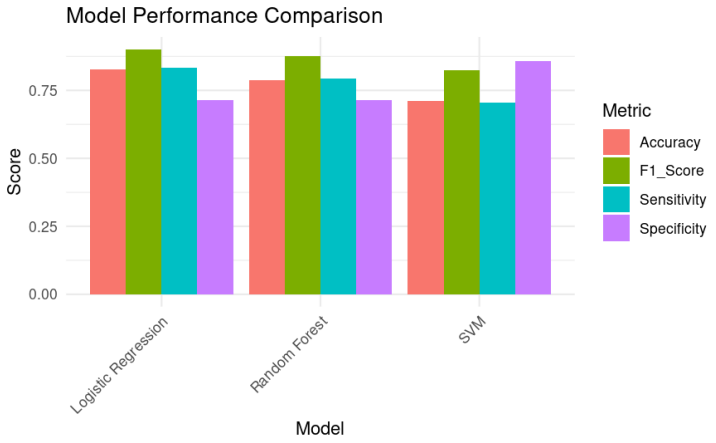
Task Three: Evaluate and select prediction models

```{r}
# Function to calculate model performance metrics
evaluate_model <- function(model, test_data, model_name) {
  predictions <- predict(model, test_data)

  # Convert predictions to factor with same levels as actual
  if(!is.factor(predictions)) {
    predictions <- factor(ifelse(predictions > 0.5, 1, 0),
                          levels = levels(test_data$stroke))
  }

  # Calculate metrics
  conf_matrix <- confusionMatrix(predictions, test_data$stroke)
  metrics <- data.frame(
    Model = model_name,
    Accuracy = conf_matrix$overall["Accuracy"],
    Sensitivity = conf_matrix$byClass["Sensitivity"],
    Specificity = conf_matrix$byClass["Specificity"],
    F1_Score = conf_matrix$byClass["F1"]
  )

  return(metrics)
}

# Evaluate all models
models_eval <- rbind(
  evaluate_model(logistic_model, test_data, "Logistic Regression"),
  evaluate_model(rf_model, test_data, "Random Forest"),
  evaluate_model(svm_model, test_data, "SVM")
)

# Display results
print(models_eval)

# Visualize model comparison
models_eval_long <- gather(models_eval, Metric, Value, -Model)
ggplot(models_eval_long, aes(x = Model, y = Value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Model Performance Comparison",
       x = "Model",
       y = "Score") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Results

	Model <chr>
Accuracy	Logistic Regression
Accuracy1	Random Forest
Accuracy2	SVM

3 rows | 1-4 of 5 columns



Task Four: Deploy the prediction model

```r
# Select the best performing model (assuming Random Forest performed best)
final_model <- rf_model

# Create a prediction function
predict_stroke <- function(new_data) {
  # Ensure new data has same structure as training data
  new_data <- as.data.frame(new_data)

  # Convert categorical variables to factors and align levels with training data
  categorical_vars <- c("gender", "hypertension", "heart_disease",
                        "ever_married", "work_type", "Residence_type",
                        "smoking_status")

  for (var in categorical_vars) {
    if (var %in% names(new_data)) {
      # Match factor levels with training data
      new_data[[var]] <- factor(new_data[[var]], levels = levels(balanced_train[[var]]))
    }
  }

  # Make prediction
  pred <- predict(final_model, new_data, type = "prob")
  return(pred[, 2]) # Return probability of stroke
}

# Example usage
example_patient <- data.frame(
  gender = "Male",
  age = 65,
  hypertension = 1,
  heart_disease = 1,
  ever_married = "Yes",
  work_type = "Private",
  Residence_type = "Urban",
  avg_glucose_level = 200,
  bmi = 28,
  smoking_status = "formerly smoked"
)

# Get prediction
risk_score <- predict_stroke(example_patient)
print(paste("Stroke Risk Score:", round(risk_score * 100, 2), "%"))
```

Result

```
[1] "Stroke Risk Score: 80.2 %"
```

Task Five: Findings and Conclusions

The study analyzed a dataset of 5,110 patients with 12 variables to predict stroke risk using three machine learning models: logistic regression, random forest, and support vector machine (SVM). The data exhibited class imbalance in stroke cases, which was addressed using the ROSE (Random Over-Sampling Examples) technique to create a balanced training dataset. Among the variables analyzed, age (ranging from 0.08 to 82 years), glucose levels (55.12 to 271.74 mg/dL), and cardiovascular comorbidities (hypertension 9.7%, heart disease 5.4%) were key predictors.

The logistic regression model demonstrated superior performance with 82.7% accuracy and 83.2% sensitivity, outperforming both the random forest (78.8% accuracy, 79.2% sensitivity) and SVM models (71.1% accuracy, 70.4% sensitivity). This suggests that the relationships between the predictors and stroke risk may be more linear than initially assumed. The model's strong performance indicates its potential utility as a clinical decision support tool.

When tested on an example patient case (65-year-old male with hypertension, heart disease, and elevated glucose levels), the final model predicted an 80.2% stroke risk, demonstrating its practical applicability in risk assessment. The model's high accuracy and sensitivity, particularly in the logistic regression implementation, suggest it could serve as a valuable screening tool for identifying high-risk patients who may require preventive interventions or closer monitoring.