2021 Stroke Prediction

Marriah Lewis and Glory Onyeugbo

6/18/2021

Table of Contents

Introduction

The initiation of sudden death of brain cells because of a loss or shortage of oxygen in the brain is known as a stroke. Ischemic and hemorrhagic strokes are the two forms of stroke that can affect an individual. Ischemic stroke occurs when an artery is blocked, resulting in a lack of oxygen in the brain. Hemorrhagic stroke, on the other hand, is caused by a break in the artery wall, which causes bleeding in the brain. The magnitude of a stroke is determined by the portion of the brain that is affected. The degree of brain damage also decides how severe a stroke would be for a person. Since brain cells have a high demand for oxygen, a lack of supply causes severe problems and the death of these vital cells.

Stroke is the second leading cause of death worldwide, accounting for about 11% of all deaths, according to the World Health Organization (WHO). A person's chances of experiencing a stroke are increased by several risk factors. Age and sex are the two main risk factors that result in a stroke case. According to various stroke studies, however, cases of stroke are usually caused by two or more causes. Age is a significant risk factor that increases as a person gets older. Stroke affects most people over the age of 65, according to research. The risk of stroke is higher in older people, as is the risk of post-stroke dementia. When compared to those in perfect health, older adults with elevated blood pressure, overweight, and diabetics are at an even greater risk. It should be known, however, that stroke is not just a risk for the elderly; young people in their teens or early twenties may also suffer from a stroke.

Gender is the other contributing factor. In comparison to men, women seem to have a higher risk of stroke. Stroke affects a greater percentage of women because of their improved longevity and the likelihood that stroke case rates rise significantly in the oldest age groups. A sudden severe headache, fatigue, numbness, vision issues, confusion, difficulty walking or talking, dizziness, and slurred speech are all symptoms of a stroke. In 2012, the President of World Stroke Organization Bo Norrving said, “The global burden of stroke has reached epidemic proportions and the situation will not improve until strong actions are taken”[1]. The stroke community must discuss the underlying molecular, epidemiological, and clinical causes and symptoms of stroke in men and women as a result of this growing epidemic. The aim of this study is identifying priority areas in stroke patients that will assist in determining the other risk factors that contribute to a patient having a stroke.

About the Data

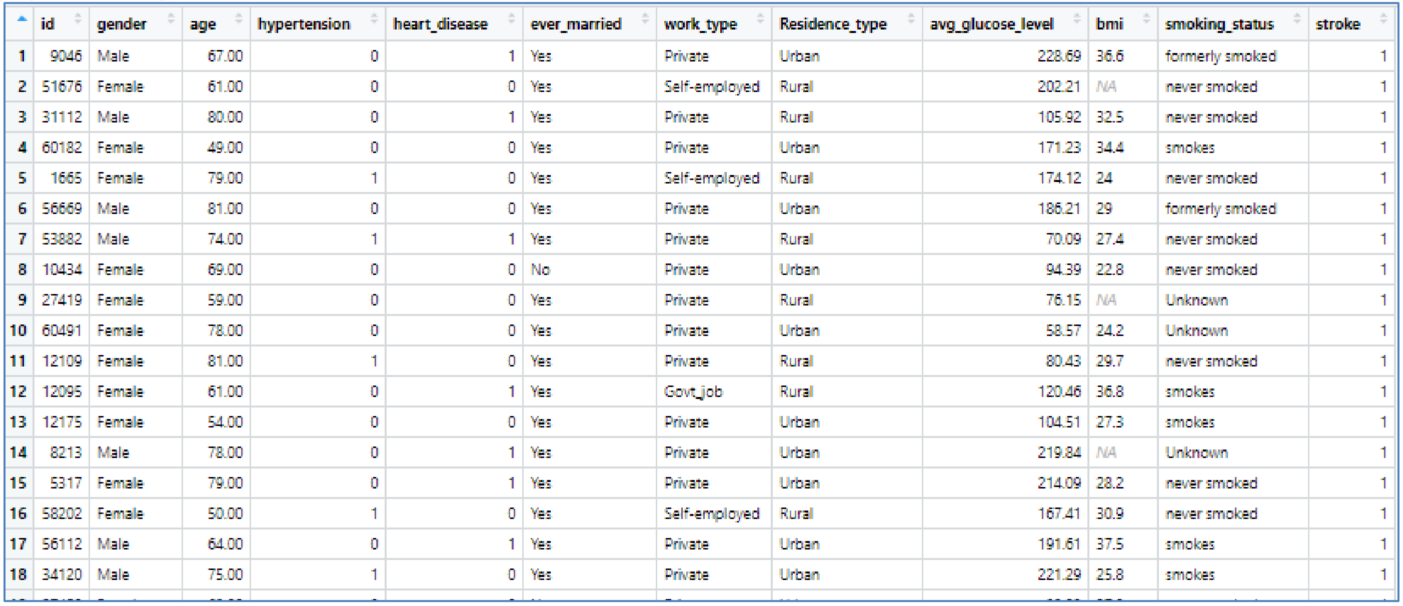
The data is a stroke prediction dataset from Kaggle[1]. The data was compiled from various clinic around the United States. The data set contains 5110 observations with 12 attributes but the project focus on a subset of the 8 most relevant attributes. The 12 attributes included in the dataset are: (1) unique_id, (2) gender (0=female, 1=male), (3) age (continuous), (4) hypertension (0=no, 1=yes),

[1] https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

(5) heart_disease (0=no, 1=yes), (6) ever_married (0=no, 1=yes), (7) work_type (private, self-employed, and others), (8) residence_type (urban or rural), (9) avg_glucose_level (continuous), (10) bmi (continuous), (11) smoking_status (former smoker, never smoked , smokes or unknown), (12) stroke (0=no, 1=yes) as shown in Figure 1.

Variable	Data Type	Description
unique_id	numeric	The patient's unique identifier
gender	character	The patient's gender: Male, Female, or Other
age	numeric	The patient's age
hypertension	int	0 if the patient does not have hypertension;1 if the patient has hypertension
heart_disease	int	0 if the patient does not have heart disease; 1 if the patient has heart disease
ever_married	character	Was the patient ever married? "Yes" or "No"
work_type	character	"children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
residence_type	character	Does the patient live in a "Rural" residence or "Urban" residence?
bmi	character	Average glucose in blood
avg_glucose_level	numeric	Body Mass Index
smoking_status	character	"formerly smoked", "never smoked", "smokes", or "unknown"
stroke	int	0 if the patient has not had a stroke; 1 if the patient has had a stroke.

Figure 1: Original dataset (stroke_pred)



	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
1	9046	Male	67.00	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
2	51676	Female	61.00	0	0	Yes	Self-employed	Rural	202.21	NA	never smoked	1
3	31112	Male	80.00	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
4	60182	Female	49.00	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
5	1665	Female	79.00	1	0	Yes	Self-employed	Rural	174.12	24	never smoked	1
6	56669	Male	81.00	0	0	Yes	Private	Urban	186.21	29	formerly smoked	1
7	53882	Male	74.00	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1
8	10434	Female	69.00	0	0	No	Private	Urban	94.39	22.8	never smoked	1
9	27419	Female	59.00	0	0	Yes	Private	Rural	76.15	NA	Unknown	1
10	60491	Female	78.00	0	0	Yes	Private	Urban	58.57	24.2	Unknown	1
11	12109	Female	81.00	1	0	Yes	Private	Rural	80.43	29.7	never smoked	1
12	12095	Female	61.00	0	1	Yes	Govt_job	Rural	120.46	36.8	smokes	1
13	12175	Female	54.00	0	0	Yes	Private	Urban	104.51	27.3	smokes	1
14	8213	Male	78.00	0	1	Yes	Private	Urban	219.84	NA	Unknown	1
15	5317	Female	79.00	0	1	Yes	Private	Urban	214.09	28.2	never smoked	1
16	58202	Female	50.00	1	0	Yes	Self-employed	Rural	167.41	30.9	never smoked	1
17	56112	Male	64.00	0	1	Yes	Private	Urban	191.61	37.5	smokes	1
18	34120	Male	75.00	1	0	Yes	Private	Urban	221.29	25.8	smokes	1

Load Libraries

Load the necessary libraries that contain the algorithms that will be used to analyze the stroke data set.

```
library(plyr)
library(dplyr)

library(grid)
library(RColorBrewer)
library(ggplot2)
library(reshape)

library(forcats)
library(arules)

library(arulesViz)
library(rpart)
library(rpart.plot)
library(unbalanced)

library(e1071)

library(randomForest)
```

Reading the Stroke Data

Used the read.csv function to call in the stroke data downloaded from Kaggle.

```
# Set the working directory.
setwd("C:\\Users\\17708\\OneDrive\\Desktop\\IST 707 Project")
# Load the dataset.
stroke.df <- read.csv("Stroke Data.csv")
# Print the first 5 instances of the data.
head(stroke.df[1:5])
```

```
##      id gender age hypertension heart_disease
## 1  9046   Male  67            0             1
## 2 51676 Female  61            0             0
## 3 31112   Male  80            0             1
## 4 60182 Female  49            0             0
## 5  1665 Female  79            1             0
## 6 56669   Male  81            0             0
```

According to the data structure of the stroke dataframe, “id”, “hypertension”, “heart_disease”, and “stroke” are considered int data types when they should be factors because they are nominal variables. The “gender”, “ever_married”, “work_type”, “residence_type”, and “smoking_status” are also nominal variables but they are deemed characters. The “bmi” column should be numeric but there appears to be N/As in the data that are causing them to be considered as character strings. The data structure presents many cleaning and organizing opportunities in the data.

Figure 2: Data Structure

```
str(stroke.df)
```

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 2
7419 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 ...
```

Data Cleaning

Once the dataset was loaded some pre-processing steps were used to clean the dataset. First, a new data frame was created to prevent compromising the integrity of the original stroke data. Next, the removal of unnecessary columns such as ever_married, work_type, residence, and unique_id since these columns were not relevant to the stroke prediction.

Figure 3: Column Removal

```
# Assign the data frame to a new data set as to not comprimise the integrity
of the original
stroke_pred <- stroke.df

# remove the id, ever_married, work_type, and Residence_type columns since th
ey have no importance to the ultimate goal of the analysis.
stroke_pred <- stroke_pred[,-c(1,6:8)]

colnames(stroke_pred)

## [1] "gender"            "age"               "hypertension"
## [4] "heart_disease"     "avg_glucose_level" "bmi"
## [7] "smoking_status"    "stroke"
```

Remove Missing Values and Duplicates

After removing the unnecessary variables, the data set was searched for duplicates and missing checked for duplicates and any missing values. Initially, there were no missing values but after viewing the dataset the bmi, contained NAs in certain rows that were not recognized by the "is.na" function. The reason for this is because bmi was considered a character column when it should have been numeric. To combat this mistake, the column was converted to numeric via the "as.numeric" function. Once again, the data was searched for missing values and return a total of 201 rows that had absent data.

Figure 4: Convert BMI to Numeric

```
stroke_pred$bmi <- as.numeric(stroke_pred$bmi)

## Warning: NAs introduced by coercion

(sum(is.na(stroke_pred)))

## [1] 201
```

There are solutions to the missing data problem; remove all rows with missing values or substitute those values with the average body mass of all the patients. There are 201 Nas in the data and if they were all removed, that can result in a loss of possibly some valuable information regarding the cause of stroke. Instead, the missing values were replaced with the average BMI of all the patients.

Figure 5: Replace Missing Value with the Column Mean

```
### Estimate and replace missing values
stroke_pred$bmi[is.na(stroke_pred$bmi)] <- mean(stroke_pred$bmi, na.rm = TRUE
)

sum(is.na(stroke_pred))

## [1] 0
```

After removing missing values, the data was searched for duplicate records but returned none.

Figure 6: Check for Duplicates

```
#check for duplicates
nrow(stroke_pred[duplicated(stroke_pred),]) #there are no duplicates

## [1] 0

#Just to make sure there are no duplicates
stroke_prediction <- stroke_pred[!duplicated(stroke_pred),]
```

Data Conversions

There were several variables in need of conversions, such as gender, hypertension, heart_disease, smoking_status, and stroke, all of which were converted to factors.

Figure 7: Convert Nominal Values to Factors

```
# Convert nominal values to factors (hypertension, heart_disease, stroke)
stroke_prediction$gender          <- factor(stroke_prediction$gender)

stroke_prediction$hypertension    <- factor(stroke_prediction$hypertension)

stroke_prediction$heart_disease   <- factor(stroke_prediction$heart_disease)

stroke_prediction$smoking_status  <- factor(stroke_prediction$smoking_status)

stroke_prediction$stroke          <- factor(stroke_prediction$stroke)
```

Discretize Age

Lastly, the final three numeric columns, age, avg_glucose_level, and bmi, were discretized to provide a more clean and organized structure. The age of each patient was placed into one of nine buckets, “child”, “teens”, “twenties”, “thirties”, “forties”, “fifties”, “sixties”, “seventies”, and “eighties”.

Figure 8: Discretize Age Column

```
# Discetize Age
(youngest_stroke <- min(stroke_prediction$age))

## [1] 0.08

(oldest_stroke  <- max(stroke_prediction$age))

## [1] 82

stroke_prediction$age <- cut(stroke_prediction$age, breaks = c(0,10,20,30,40,
50,60,70,80,90),
                    labels=c("child","teens","twenties","thirties","forties",
"fifties","sixties", "seventies", "eighties"))
```

Discretize Average Glucose Level

The avg_glucose_level column was split by 50s, starting with 50 and ending with 300.

Figure 9: Discretize Average Glucose Column

```
# Discretize Avg Glucose
(lowest_glucose   <- min(stroke_prediction$avg_glucose_level))

## [1] 55.12

(highest_glucose  <- max(stroke_prediction$avg_glucose_level))

## [1] 271.74

stroke_prediction$avg_glucose_level <- cut(stroke_prediction$avg_glucose_leve
l, breaks = c(50,100,150,200,250,300),
                    labels=c("50 - 100", "101 - 150", "151 - 200", "201 - 250
", "251 - 300"))
```

Discretize Body Mass Index

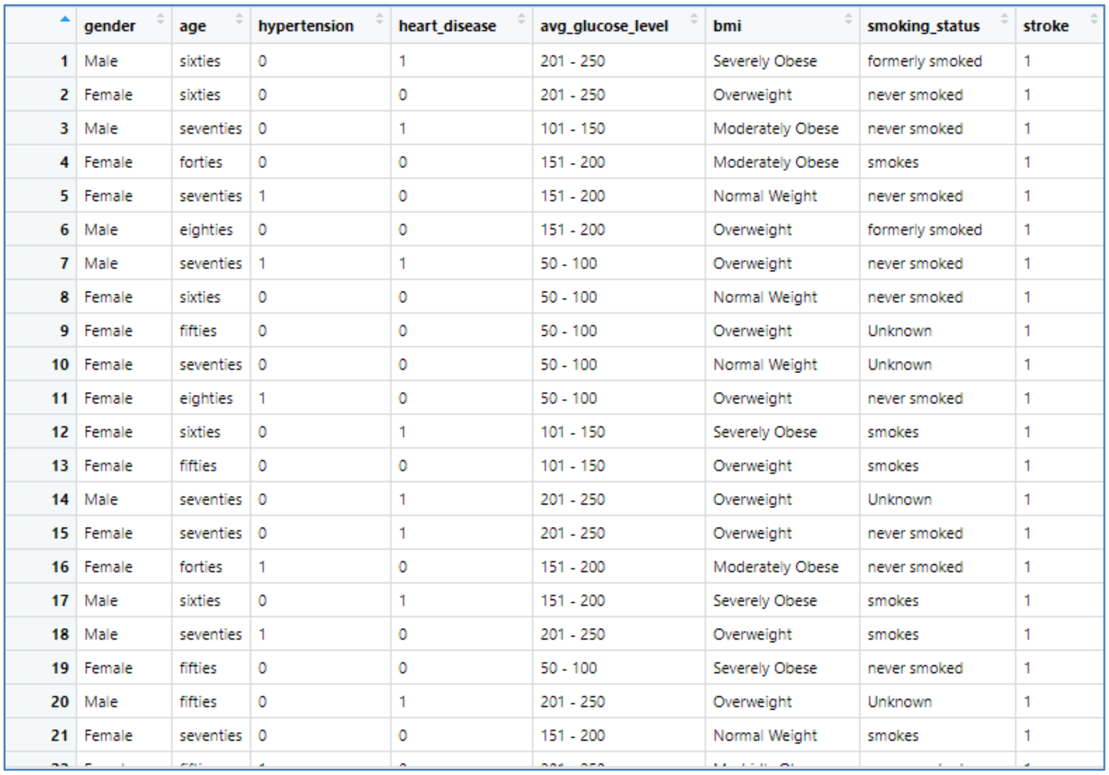
The BMI in the data ranged from 10.3 to 97.6. From this, the bmi of a patient was put into buckets: “Underweight” if bmi < 18.5, “Normal Weight” if bmi is 18.5 to 25, “Overweight” if bmi is 25 to 30, “Moderately Obese” if bmi is 30 to 35, “Severely Obese” if bmi is 35 to 40, “Very Severely Obese” is bmi is 40 to 45, “Morbidly Obese” if bmi is 45 to 50, “Super Obese” if bmi is 50 to 60, and “Hyper Obese” if bmi is > 60.

Figure 10: Discretize Body Mass Index

```
# Discretize Body Mass Index
(low_bmi   <- min(stroke_prediction$bmi))

## [1] 10.3

(high_bmi  <- max(stroke_prediction$bmi))

## [1] 97.6

stroke_prediction$bmi <- cut(stroke_prediction$bmi, breaks = c(0,18.5,25,30,3
5,40,45,50,60,Inf),
                    labels = c("Underweight", "Normal Weight", "Overweight",
"Moderately Obese", "Severely Obese", "Very Severely Obese", "Morbidly Obese"
, "Super Obese", "Hyper Obese"))
```

Completely Cleaned Stroke Data

Figure 11: Clean Data Set (stroke_prediction)



	gender	age	hypertension	heart_disease	avg_glucose_level	bmi	smoking_status	stroke
1	Male	sixties	0	1	201 - 250	Severely Obese	formerly smoked	1
2	Female	sixties	0	0	201 - 250	Overweight	never smoked	1
3	Male	seventies	0	1	101 - 150	Moderately Obese	never smoked	1
4	Female	forties	0	0	151 - 200	Moderately Obese	smokes	1
5	Female	seventies	1	0	151 - 200	Normal Weight	never smoked	1
6	Male	eighties	0	0	151 - 200	Overweight	formerly smoked	1
7	Male	seventies	1	1	50 - 100	Overweight	never smoked	1
8	Female	sixties	0	0	50 - 100	Normal Weight	never smoked	1
9	Female	fifties	0	0	50 - 100	Overweight	Unknown	1
10	Female	seventies	0	0	50 - 100	Normal Weight	Unknown	1
11	Female	eighties	1	0	50 - 100	Overweight	never smoked	1
12	Female	sixties	0	1	101 - 150	Severely Obese	smokes	1
13	Female	fifties	0	0	101 - 150	Overweight	smokes	1
14	Male	seventies	0	1	201 - 250	Overweight	Unknown	1
15	Female	seventies	0	1	201 - 250	Overweight	never smoked	1
16	Female	forties	1	0	151 - 200	Moderately Obese	never smoked	1
17	Male	sixties	0	1	151 - 200	Severely Obese	smokes	1
18	Male	seventies	1	0	201 - 250	Overweight	smokes	1
19	Female	fifties	0	0	50 - 100	Severely Obese	never smoked	1
20	Male	fifties	0	1	201 - 250	Overweight	Unknown	1
21	Female	seventies	0	0	151 - 200	Normal Weight	smokes	1
22	Female	fifties	1	0	201 - 250	Morbidly Obese	never smoked	1

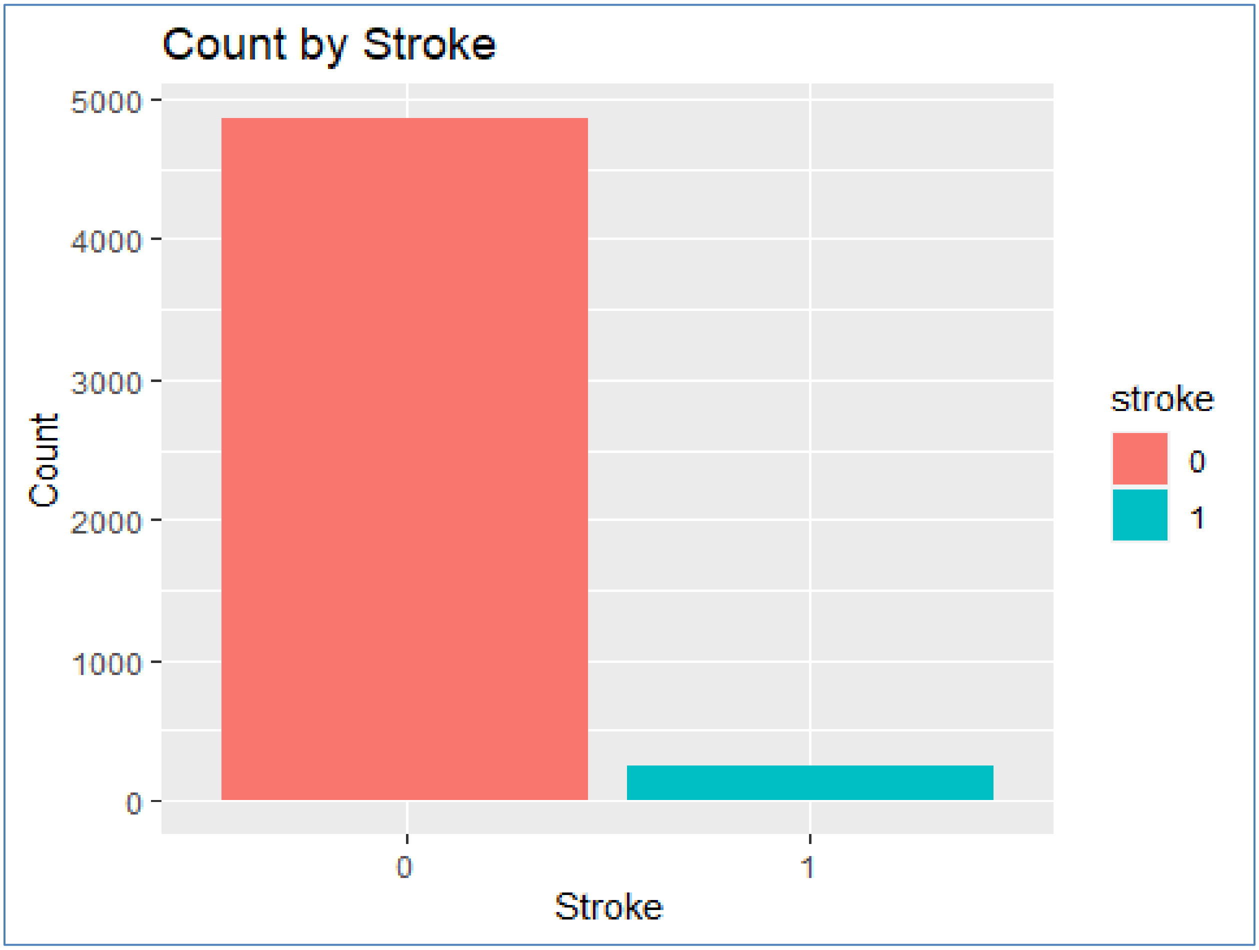
Figure 12: Summary of the Clean Data

```
summary(stroke_prediction)
```

```
##     gender            age       hypertension heart_disease avg_glucose_lev
el
##  Female:2994   fifties  : 823   0:4612       0:4834        50 - 100 :3131
##  Male  :2115   forties  : 739   1: 498       1: 276        101 - 150:1249
##  Other :   1   thirties : 674                              151 - 200: 296
##                sixties  : 594                              201 - 250: 409
##                seventies: 594                              251 - 300:  25
##                twenties : 545
##                (Other)  :1141
##                    bmi               smoking_status stroke
##  Overweight         :1610   formerly smoked: 885   0:4861
##  Normal Weight      :1258   never smoked   :1892   1: 249
##  Moderately Obese   : 985   smokes         : 789
##  Severely Obese     : 500   Unknown        :1544
##  Underweight        : 349
##  Very Severely Obese: 253
##  (Other)            : 155
```

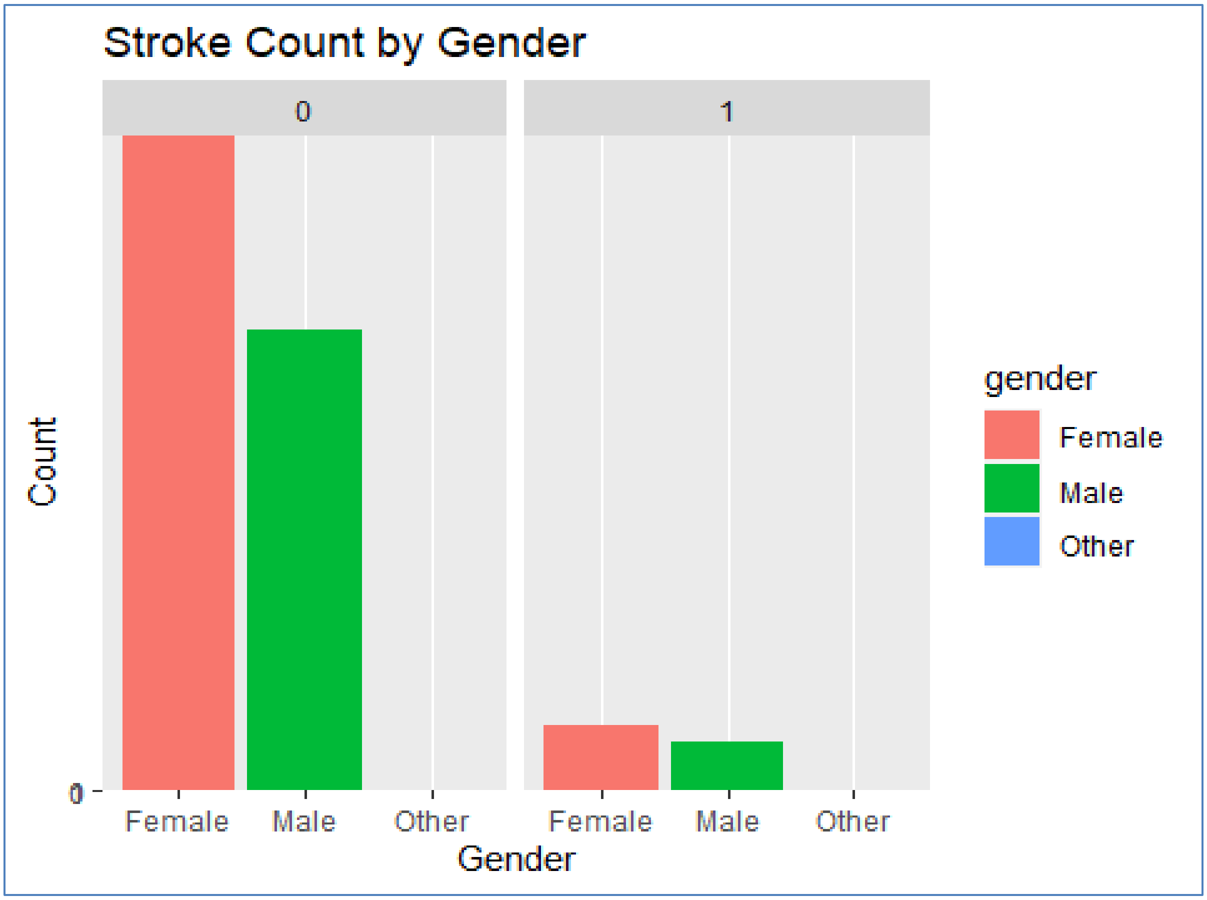
Exploratory Data Analysis

Figure 13: Patient Stroke Count (Bar Graph)



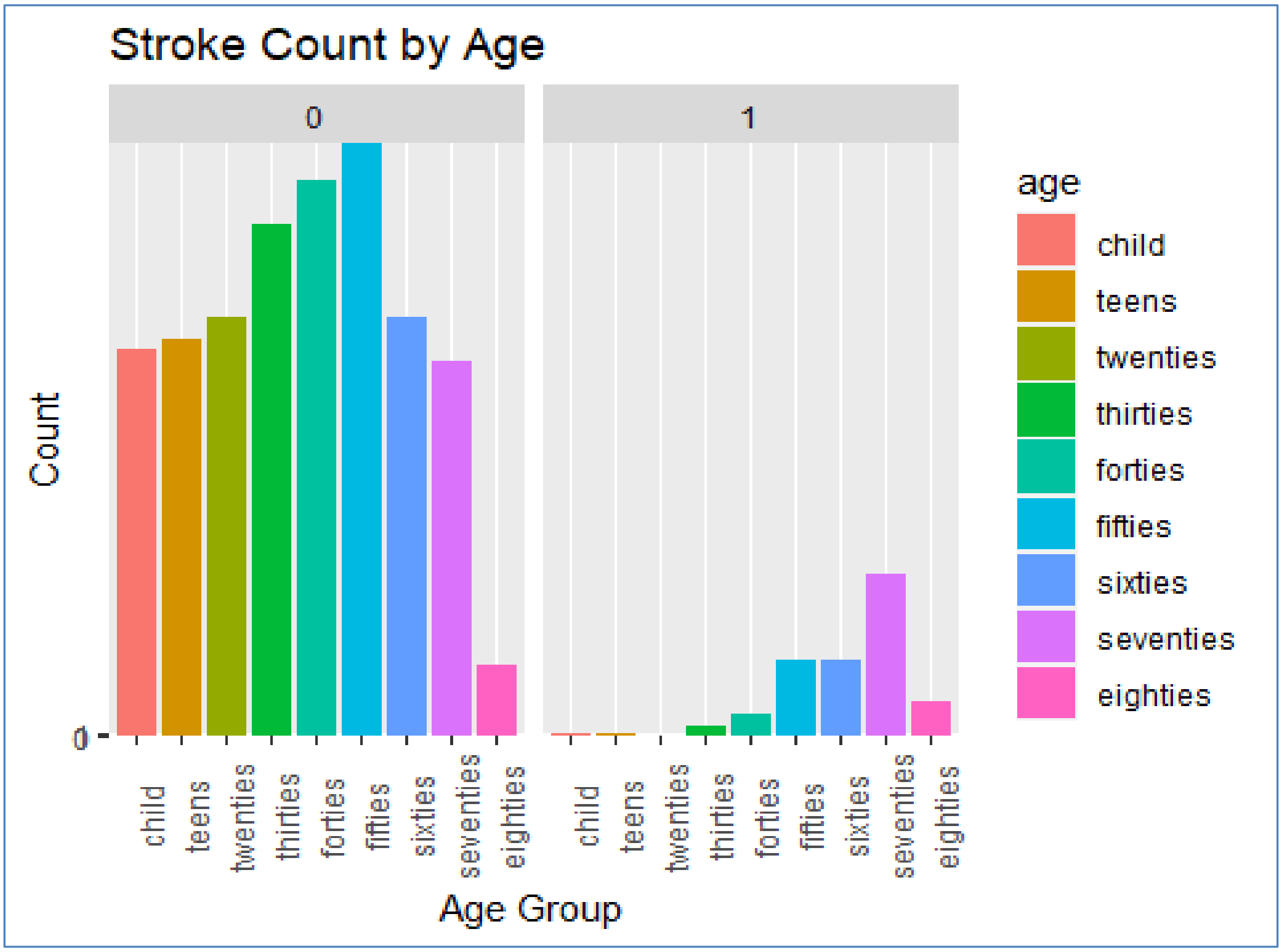
The data set contains 5,110 patient records, but only 249 patients have suffered a stroke while the other 4,861 have not. This shows a class imbalance in the data with stroke patients being in the minority. This may lead to poor modelling results if the stroke class imbalance is not properly addressed.

Figure 14: Patient Stroke Count by Gender (Bar Graph)



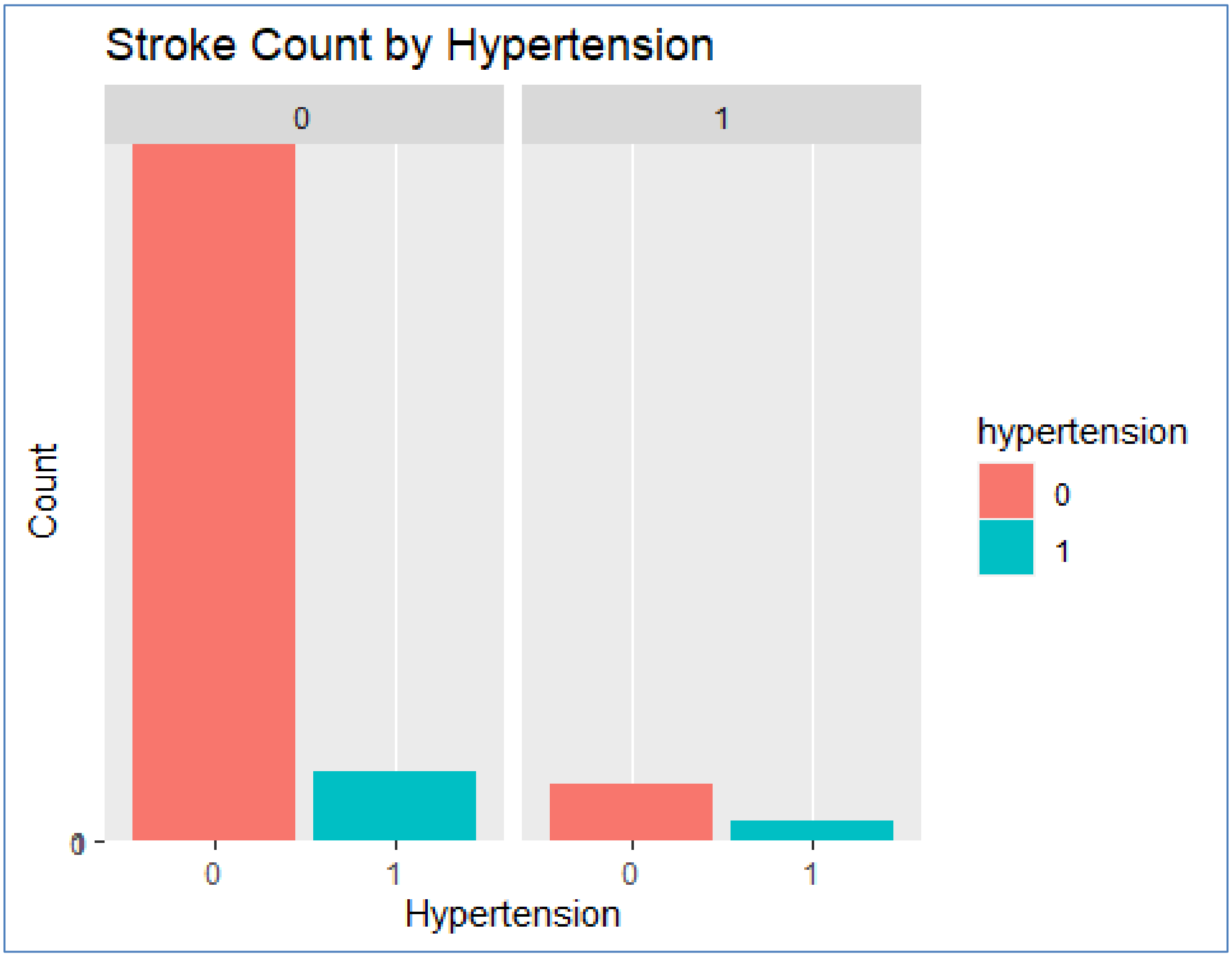
Most patients in the data classifies as females with some identifying as males and only one patient identifying as other. There are 2,994 female patients but only 141 had a stroke before. There are 2115 male patients and 108 have suffered a stroke. The one patient classified as other has never had stroke. According to the graph, it seems as though women and girls are more likely to encounter a stroke than men, but with many patients being female, it can cause a pull towards females. This can be explored further in this analysis.

Figure 15: Patient Stroke Count by Age (Bar Graph)



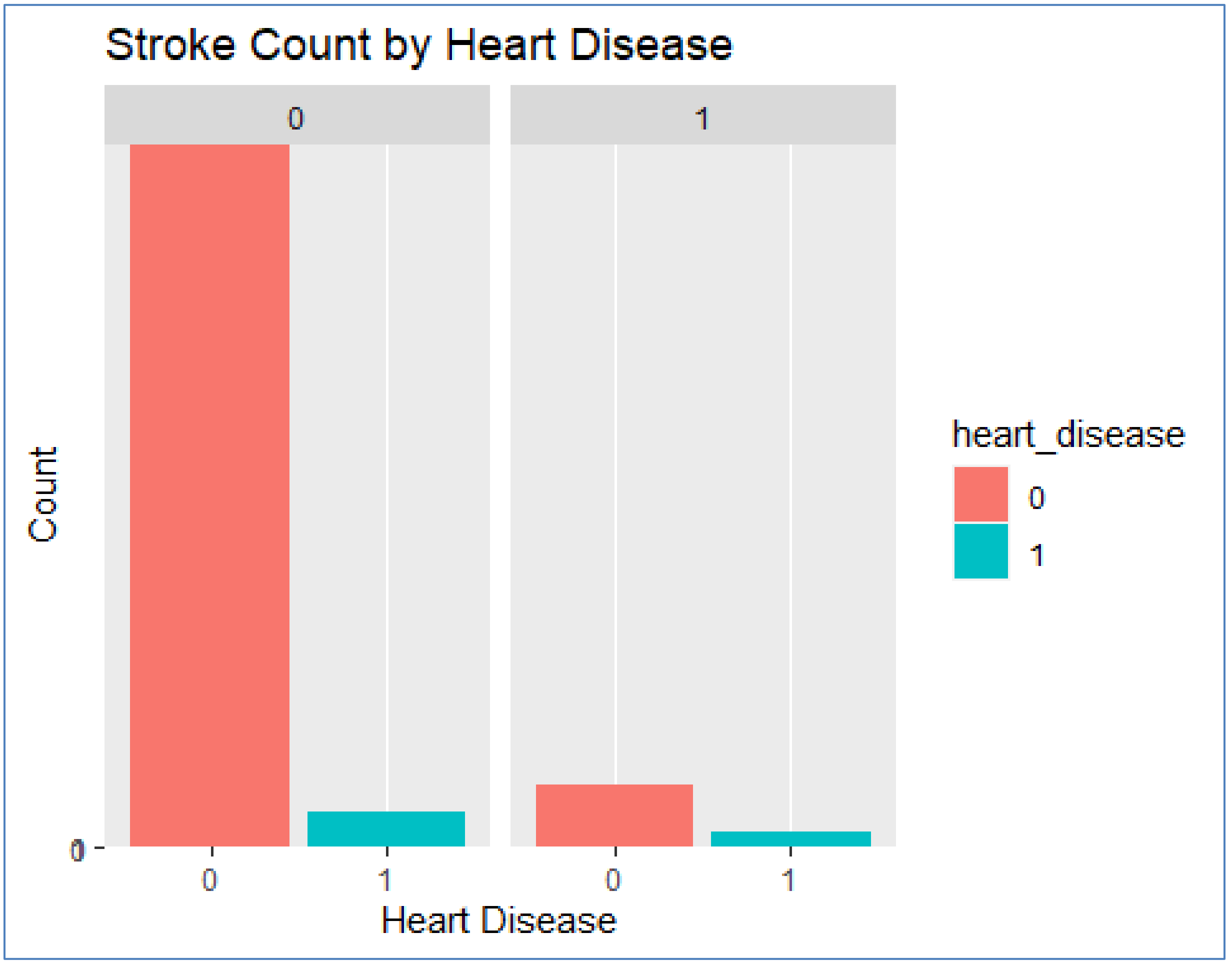
Patients in their fifties appear more in the data than any other age group whereas patients 80 years of older appear the least, but the others are not too far behind. The graph above shows that strokes tend to occur more with the older patients than younger ones. There are a few instances of someone in a younger age group suffering a stroke, for instance one child and one teen, but this seems more like a rarity and could be due to prior health issues rather than their age. None of the patients in their twenties have had a stroke before, but the stroke count starts to pick up with the thirties and reaches its peak in the seventies. Although there are more 50-something patients, they do not seem to struggle with strokes nearly as much as patients in their seventies.

Figure 16: Patient Stroke Count by Hypertension (Bar Graph)



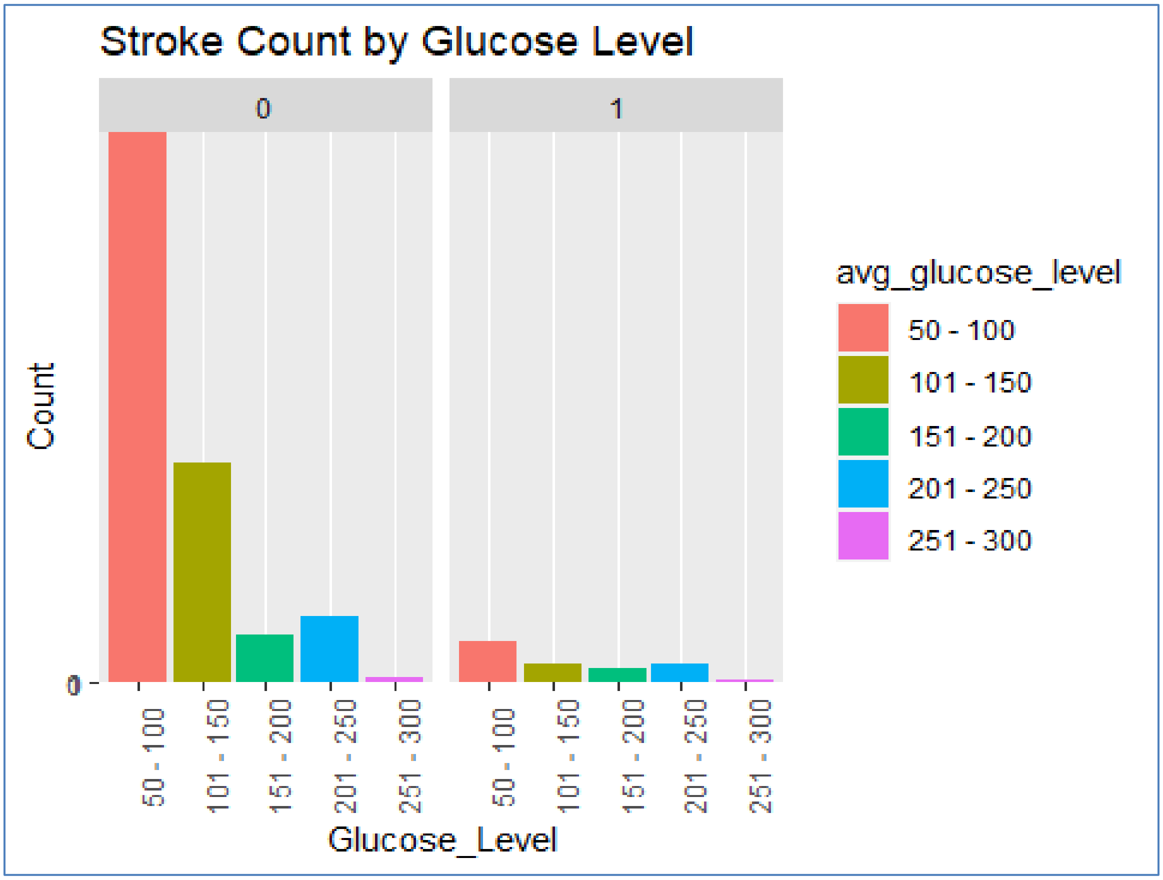
Much like the stroke class, there is a massive imbalance between patients who have hypertension and those who do not, with the positive (patients with hypertension) being the minority. However, the bar graph depicts a case where patients who are HBP negative have a higher chance of suffering a stroke than patients who are not. The imbalance could be causing this pull. There are 4,612 patients who are not living with hypertension and 498 who are, so it is practical to encounter more people who have suffered a stroke from such a large class.

Figure 17: Patient Stroke Count by Heart Disease (Bar Graph)



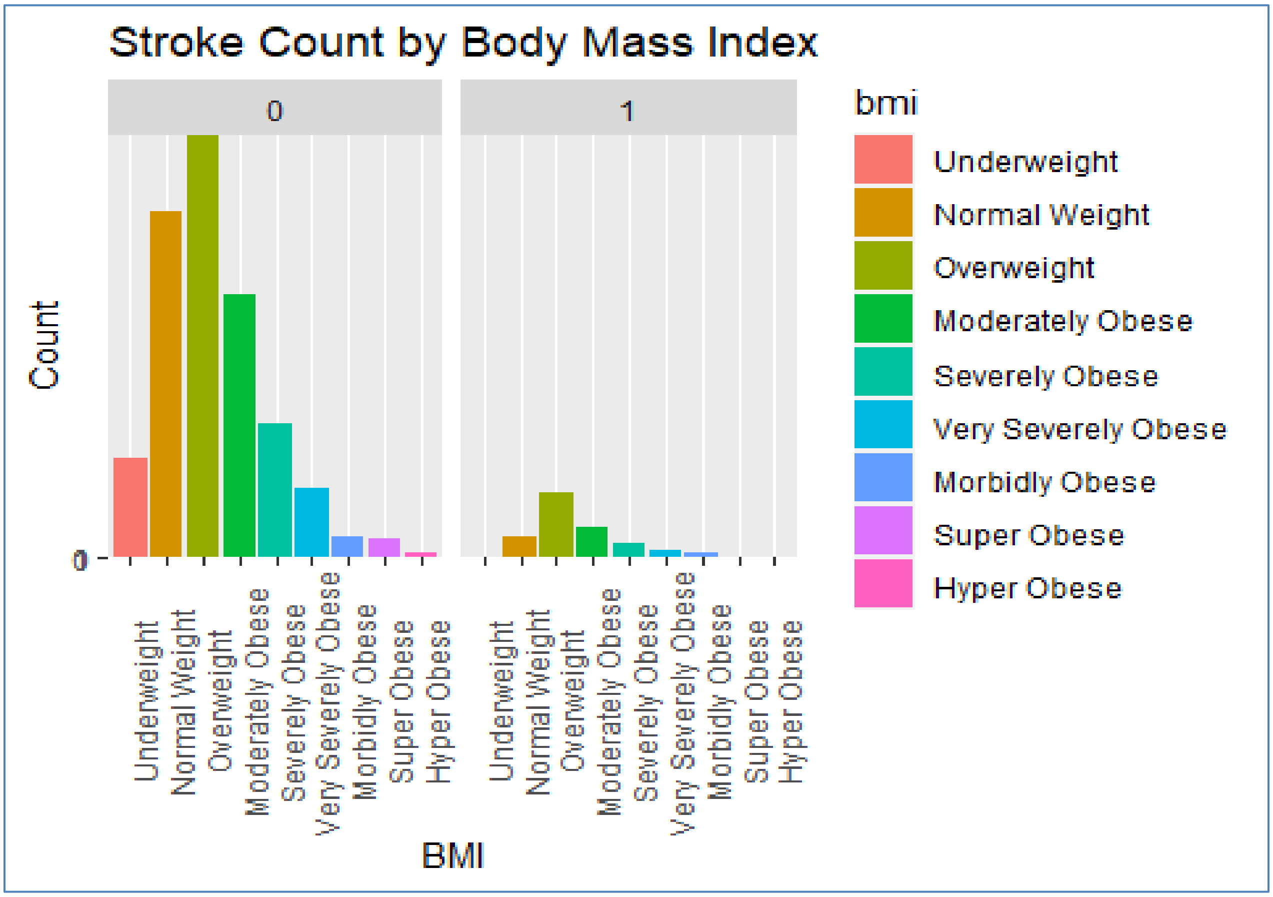
The results from heart disease are much like hypertension. There are less patients with heart disease than patients without. According to the graph, patients with heart disease are less likely to suffer a stroke but they also appear very little in the data so that could be a reason. On the other hand, this could possibly mean that heart disease is not a significant factor for stroke prediction as one might think. Maybe a person can still suffer a stroke even without prior health complications. This can be explored further with the models.

Figure 18: Patient Stroke Count by Glucose Level (Bar Graph)



Most patients have a glucose level between 50 and 100 whereas less patients have a glucose level above 250. However, it seems as though a person can still suffer a stroke, regardless of their glucose level. The graph does show that patients with lower levels of glucose have suffered stroke the most, once again, they also make up the major.

Figure 19: Patient Stroke Count by BMI (Bar Graph)

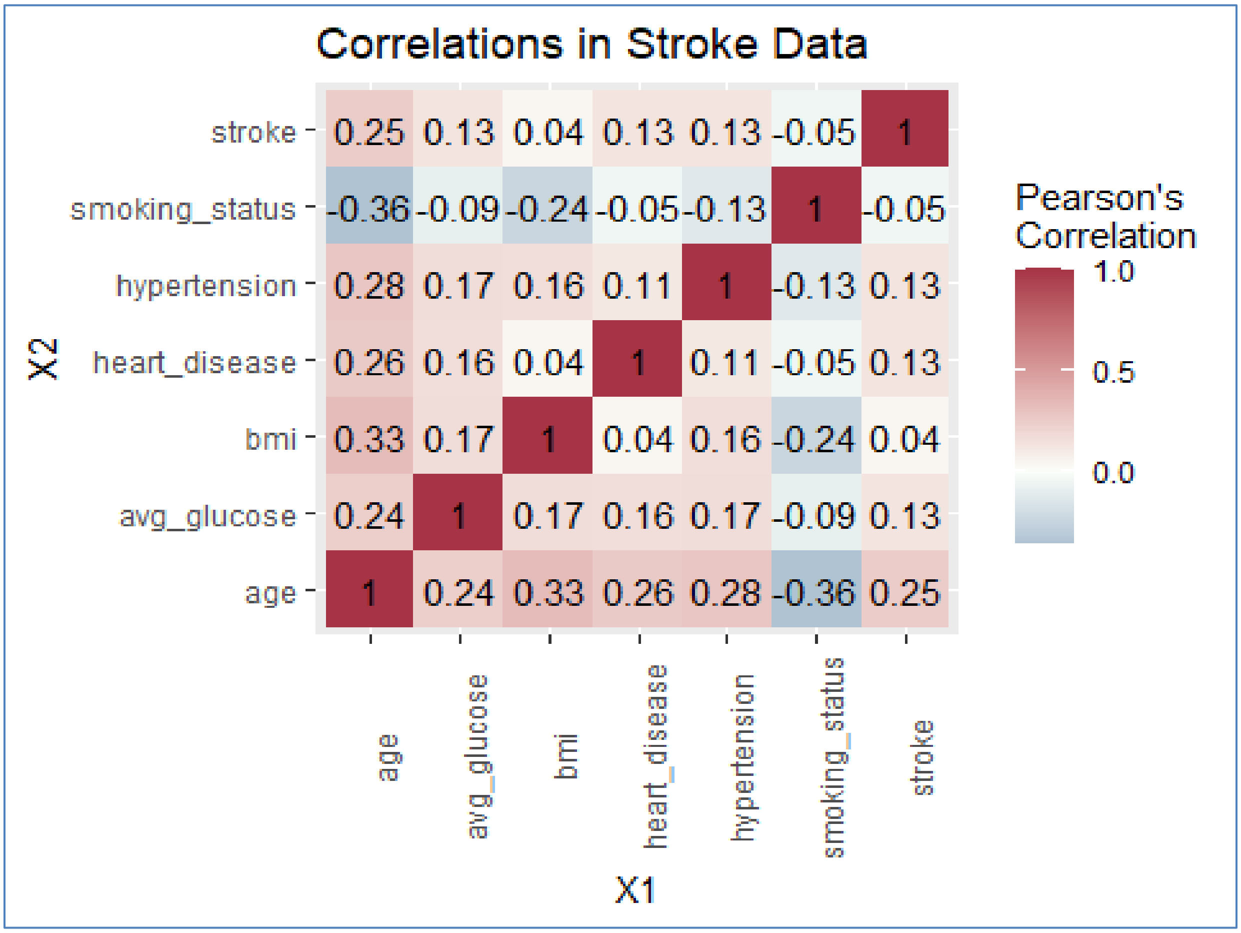


Most of the patients in that data are in the overweight class while a very small few are hyper obese. Patients who are hyper underweight, super obese, and hyper obese have not suffered a stroke. Strokes tend to hit patients who are overweight than any other weight class.

Correlation Matrix

The heat map below displays the correlation between the eight variables in the data. A positive correlation means that two variables have a positive connection as one variable increases, so does the other variable. A correlation of 1 represents a strong positive correlation. On the other hand, a negative correlation means as one variable increases, the other variable decreases. A correlation of -1 represents a strong negative correlation. A correlation of 0 shows no correlation between the two variables. The color of the tiles on the map, along with the numbers, shows the correlation strength between the variables. The redder the tile is, the strong the positive correlation is between the variable. The bluer the tile is, the stronger the negative correlation between the variables. If the tile's color is closer to white, then the correlation between the variables is bordering on 0.

Figure 20: Correlation Matrix



While ignoring the dark red diagonals, there does not seem to be any strong correlation (positive or negative) between variables. The strong correlation is -0.36, which is between age and smoking status. The strongest positive correlation (0.33) is between age and bmi, which is understandable because as a person gets older, they tend to weigh more. The map shows that age does have strongest correlation with all the variables, including stroke. Smoking status has a negative correlation with every variable on the map. BMI has the weakest correlation with heart_disease (0.04) and stroke (0.04).

Models and Analysis

Association Rule Mining (*apriori*)

Association rule mining is an unsupervised learning technique that evaluates transactions for associations between variables. Through association rule mining, patterns emerge in the data that predicts the occurrence of an item based on the occurrences of other items.

From this, actionable intel can be generated and used towards the future. The association rule mining function used is "apriori" from the arules library.

Association Rule Mining uses three key measurements to determine how strongly a rule is observed in the data -- support, confidence, and lift. Support shows the number of times the items appear together in the dataset and is calculated using the formula below.

$$\boldsymbol{Support}\ (\boldsymbol{lhs}, \boldsymbol{rhs}) = \frac{Count\ of\ lhs\ and\ rhs\ together}{Total\ Number\ of\ Entries\ in\ the\ data}$$

Confidence explores the probability that Y (rhs) appears in the data given that X (lhs) is in the data.

$$\boldsymbol{Confident}\ (\boldsymbol{lhs}, \boldsymbol{rhs}) = \frac{Count\ of\ lhs\ and\ rhs\ together}{Count\ of\ lhs\ in\ the\ data\ set}$$

Lastly, the lift measures the dependence or correlation between X (lhs) and Y(rhs).

$$= \frac{\begin{array}{c}\boldsymbol{Support}\ (\boldsymbol{lhs}, \boldsymbol{rhs}) \\ \dfrac{Count\ of\ lhs\ and\ rhs\ together}{Total\ Number\ of\ Entries\ in\ the\ data}\end{array}}{\left(\dfrac{Count\ of\ rhs}{Total\ Number\ of\ Entries\ in\ the\ data}\right) * \left(\dfrac{Count\ of\ lhs}{Total\ Number\ of\ Entries\ in\ the\ data}\right)}$$

A high lift is something to pay attention to. A lift of one indicates independence.

Once the rules are generated and stored in a vector, "sort" function sorts the rules by support, confidence, or lift in either descending or ascending order. The "inspect" function displays the list of the rules that were created.

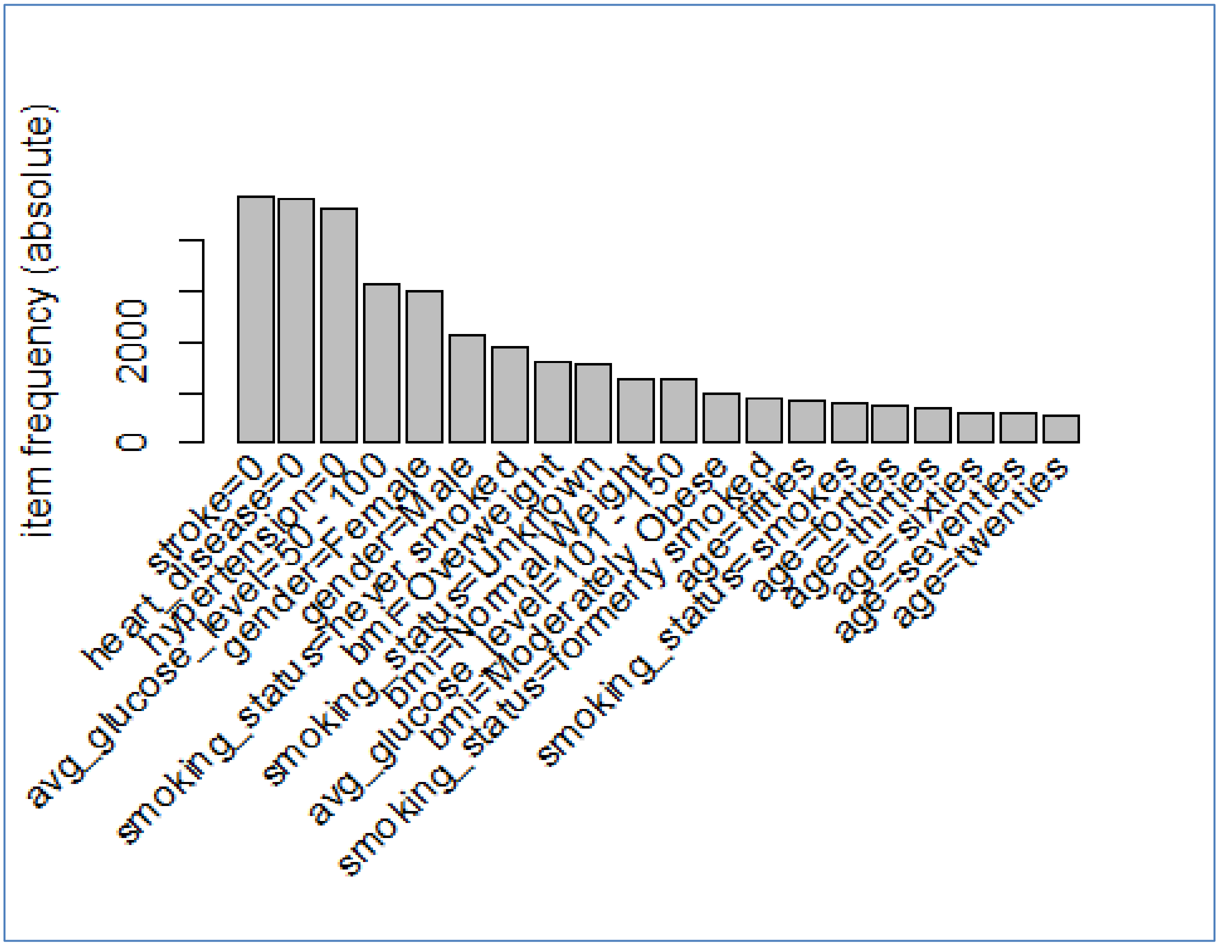
Data Preparation

The "apriori" function requires discretized variables and a transaction data set. Since all variables in the stroke prediction data frame are factors, no conversions were needed. However, the data is in a data frame format when is should be transactions. The "as" function converts the data frame to transactional data.

First, the most frequent items are identified to better understand what items are most likely to appear in mined rules and to prepare for making adjustments based on item frequency.

```r
# Include only factor variable in this analysis.
stroke_arules <- stroke_prediction

stroke_transactions <- as(stroke_arules, "transactions")

itemFrequencyPlot(stroke_transactions, topN = 20, type = "absolute")
```

Figure 21: Item Frequency Plot



The bar graph displays the top 20 “items” in the data. “stroke = 0” stands as the most popular item and “heart_disease = 0” comes in a close second. Thus, stroke=0 may possibly appear with many of the generated rules.

Initial Attempt

With the initial attempt at mining rules, support is set to 0.002 and confidence is 0.5.

```
rules_stroke <- apriori(stroke_transactions, parameter = list(supp = 0.002, c
onf = 0.5))

rules_stroke <- sort(rules_stroke, decreasing = TRUE, by = "lift")
```

Inspect the Rules

The top 5 rules had high lift values, very low support, and some high and some low confidence. Using the first rule as an example,   the metrics can be interpreted as such:

- A support of 0.002 means that of all the transaction, only 0.02% represent the left-hand side and right-hand side combinations.

- A confidence of 0.5454 indicates that of all transaction where the patient is a male in their seventies with a glucose level between 50 and 100 and smokes were 54.54% likely to have heart disease.

- A lift of 10.1 indicated that a patient with heart disease is 10 times more likely to be a male in their seventies with an average glucose level between 50 and 100 who smokes.

Figure 22: Inspect the Rules

```
inspect(rules_stroke[1:5])

##     lhs                              rhs               support confidence
coverage  lift  count
## [1] {gender=Male,
##      age=seventies,
##      avg_glucose_level=50 - 100,
##      smoking_status=smokes}       => {heart_disease=1} 0.002348337  0.5454

                                                    0.004305284 10.098814    12
## [2] {gender=Female,
##      avg_glucose_level=101 - 150,
##      bmi=Underweight,
##      smoking_status=Unknown}      => {age=child}       0.007045010  0.9000
0.007827789  9.071006    36


## [3] {gender=Female,
##      hypertension=0,
##      avg_glucose_level=101 - 150,
##      bmi=Underweight,
##      smoking_status=Unknown}      => {age=child}       0.007045010  0.9000
0.007827789  9.071006    36


## [4] {gender=Female,
##      heart_disease=0,
##      avg_glucose_level=101 - 150,
##      bmi=Underweight,
##      smoking_status=Unknown}      => {age=child}       0.007045010  0.9000
0.007827789  9.071006    36


## [5] {gender=Female,
##      avg_glucose_level=101 - 150,
##      bmi=Underweight,
##      smoking_status=Unknown,
##      stroke=0}                    => {age=child}       0.007045010  0.9000
000 0.007827789  9.071006    36
```

Second Attempt

Because the support at 0.002 is very low, adjustments were made to the apriori algorithm to mine for stronger rules. The strongest rules were found after setting support to 0.2 and confidence at 0.8.

```
rules_stroke_2 <- apriori(stroke_transactions, parameter = list(supp = 0.2, c
onf = 0.8))

rules_stroke_2 <- sort(rules_stroke_2, decreasing = TRUE, by = "lift")
```

Inspect the Rules

This configuration started to produce rules with high lift and relatively strong support. The apriori algorithm generated rules that included patients that do not have heart disease, which, as shown by the item frequency plot, is one of the most important "items" in the data set. According to the rules, patients who do not suffer heart complications do not usually have hypertension.

Figure 23: Inspect the Rules

```
inspect(rules_stroke_2[1:5])

##     lhs                          rhs                support confidence  cov
erage     lift count
## [1] {heart_disease=0,
##      smoking_status=Unknown,
##      stroke=0}                => {hypertension=0} 0.2767123  0.9704873 0.28
51272 1.075280  1414


## [2] {heart_disease=0,
##      smoking_status=Unknown} => {hypertension=0} 0.2835616  0.9685829 0.29
27593 1.073170  1449


## [3] {smoking_status=Unknown,
##      stroke=0}                => {hypertension=0} 0.2835616  0.9679359 0.29
29550 1.072453  1449


## [4] {heart_disease=0,
##      bmi=Normal Weight,
##      stroke=0}                => {hypertension=0} 0.2248532  0.9663583 0.23
26810 1.070705  1149
```

Generate Stroke Rules

The is to explore associations related to the “stroke” attribute to determine what type of patients are more likely to have a stroke. To do this, the right-hand side of the rule to must be set to “stroke”. However, given the class imbalance in the stroke data set, most of the generated rules are geared towards non-stroke patients. Thus, two different rules were mined, one for patients who never had a stroke and the other towards patients who did.

The first set of rules placed “stroke=0” on the right-hand side and support and confidence were set to high values, 0.3 and 0.9, respectively.

```
non_stroke_rules <- apriori(stroke_transactions, parameter = list(supp = 0.3,
conf = 0.9, maxlen = 3), appearance = list(rhs = "stroke=0"))

non_stroke_rules <- sort(non_stroke_rules, decreasing = TRUE, by = "lift")
```

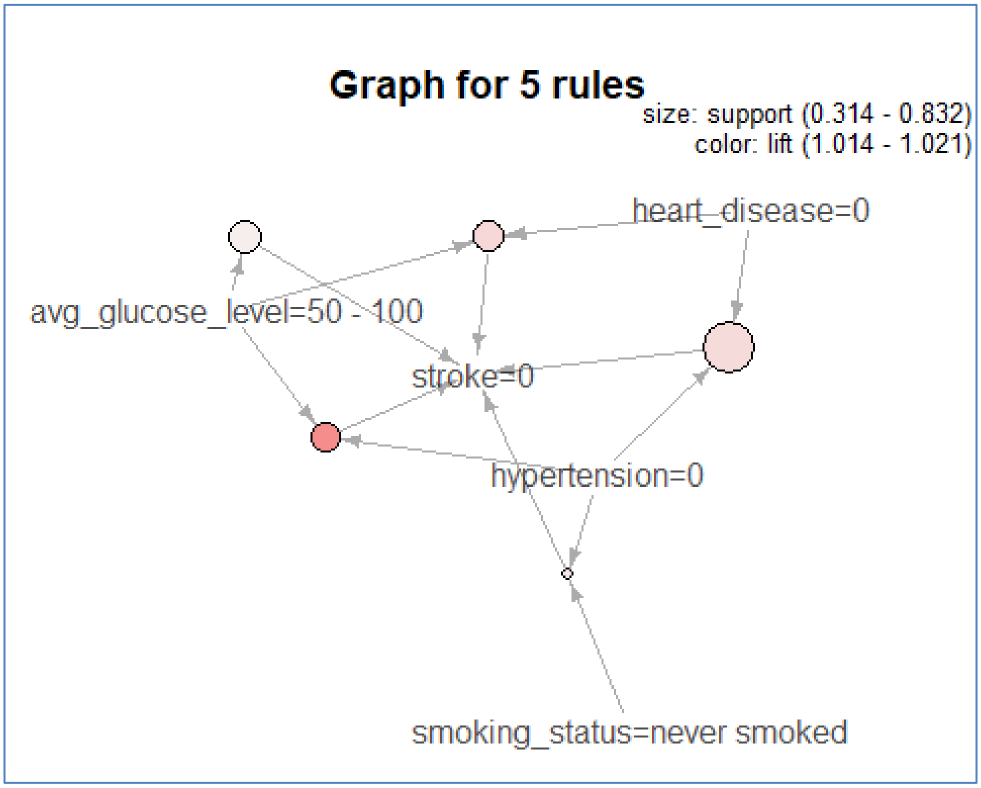
Figure 24: Inspect the Rules

```
inspect(non_stroke_rules)

##      lhs                              rhs          support confidence  cov
erage      lift count
## [1]  {hypertension=0,
##       avg_glucose_level=50 - 100}  => {stroke=0} 0.5510763  0.9713694 0.56
73190 1.0211269  2816
## [2]  {heart_disease=0,
##       avg_glucose_level=50 - 100}  => {stroke=0} 0.5700587  0.9664897 0.58
98239 1.0159972  2913
## [3]  {hypertension=0,
##       heart_disease=0}             => {stroke=0} 0.8318982  0.9661364 0.86
10568 1.0156258  4251
## [4]  {hypertension=0,
##       smoking_status=never smoked} => {stroke=0} 0.3135029  0.9650602 0.32
48532 1.0144945  1602
## [5]  {avg_glucose_level=50 - 100}  => {stroke=0} 0.5908023  0.9642287 0.61
27202 1.0136204  3019
## [6]  {gender=Female,
##       hypertension=0}              => {stroke=0} 0.5119374  0.9624724 0.53
18982 1.0117741  2616
## [7]  {gender=Female,
##       avg_glucose_level=50 - 100}  => {stroke=0} 0.3544031  0.9617631 0.36
84932 1.0110285  1811
## [8]  {hypertension=0}              => {stroke=0} 0.8667319  0.9603209 0.90
25440 1.0095124  4429
## [9]  {gender=Male,
##       heart_disease=0}             => {stroke=0} 0.3663405  0.9590164 0.38
19961 1.0081411  1872
## [10] {heart_disease=0}             => {stroke=0} 0.9064579  0.9582127 0.94
59883 1.0072962  4632
```

```
## [11] {gender=Female,
##       heart_disease=0}             => {stroke=0} 0.5399217  0.9576536 0.56
37965 1.0067085  2759
## [12] {gender=Male,
##       hypertension=0}              => {stroke=0} 0.3545988  0.9572108 0.37
04501 1.0062430  1812
## [13] {heart_disease=0,
##       smoking_status=never smoked} => {stroke=0} 0.3373777  0.9567148 0.35
26419 1.0057215  1724
## [14] {gender=Female}               => {stroke=0} 0.5583170  0.9529058 0.58
59100 1.0017175  2853
## [15] {smoking_status=never smoked} => {stroke=0} 0.3526419  0.9524313 0.37
02544 1.0012187  1802
## [16] {}                            => {stroke=0} 0.9512720  0.9512720 1.00
00000 1.0000000  4861
## [17] {gender=Male}                 => {stroke=0} 0.3927593  0.9489362 0.41
38943 0.9975445  2007
```

Most of the rule generated show that a patient that does not have heart disease and they have a low average glucose level (avg_glucose_level = 50 – 100) have not suffered a stroke before.

Figure 25: Association Rules Plot for Non-Stoke Patients


Graph for 5 rules
size: support (0.314 - 0.832)
color: lift (1.014 - 1.021)
heart_disease=0
avg_glucose_level=50 - 100
stroke=0
hypertension=0
smoking_status=never smoked

Rules for Stoke = 1

There are only so many patients who suffered from a stroke in the data set. Therefore, for the first attempt the confidence and support were set to very low.

```
stroke_rules <- apriori(stroke_transactions, parameter = list(supp = 0.001, c
onf = 0.01, maxlen = 3), appearance = list(rhs = "stroke=1"))

stroke_rules <- sort(stroke_rules, decreasing = TRUE, by = "lift")
```

Figure 26: Inspect the rules

```
inspect(stroke_rules[1:5])

##     lhs                               rhs          support  confidence
coverage     lift count
## [1] {heart_disease=1,
##      bmi=Severely Obese}           => {stroke=1} 0.001761252  0.3333333 0.0
05283757 6.840696     9
## [2] {age=seventies,
##      avg_glucose_level=151 - 200} => {stroke=1} 0.002935421  0.2727273 0.0
10763209 5.596933    15
## [3] {age=eighties,
##      bmi=Overweight}               => {stroke=1} 0.002348337  0.2666667 0.0
08806262 5.472557    12
## [4] {heart_disease=1,
##      avg_glucose_level=201 - 250} => {stroke=1} 0.003522505  0.2647059 0.0
13307241 5.432318    18
## [5] {age=eighties,
##      heart_disease=1}              => {stroke=1} 0.001369863  0.2592593 0.0
05283757 5.320541     7
```

Many of the generated rules show that older patients (70s and 80s) have suffered a stroke. However, the rules have very low support and confidence but a high lift.

New rules are generated for stroke patients. A new transaction data set was created that only contained stroke patients. The apriori algorithm is applied to the new data set where support is set to 1.0 and confidence is set to 0.9

```
transactions <- as(stroke_arules[stroke_arules$stroke == "1",], "transactions
")

stroke_rules_2 <- apriori(transactions, parameter = list(supp = 0.1, conf = 0
.9, minlen = 4), appearance = list(rhs = "stroke=1"))

stroke_rules_2 <- sort(stroke_rules_2, decreasing = TRUE, by = "lift")
```

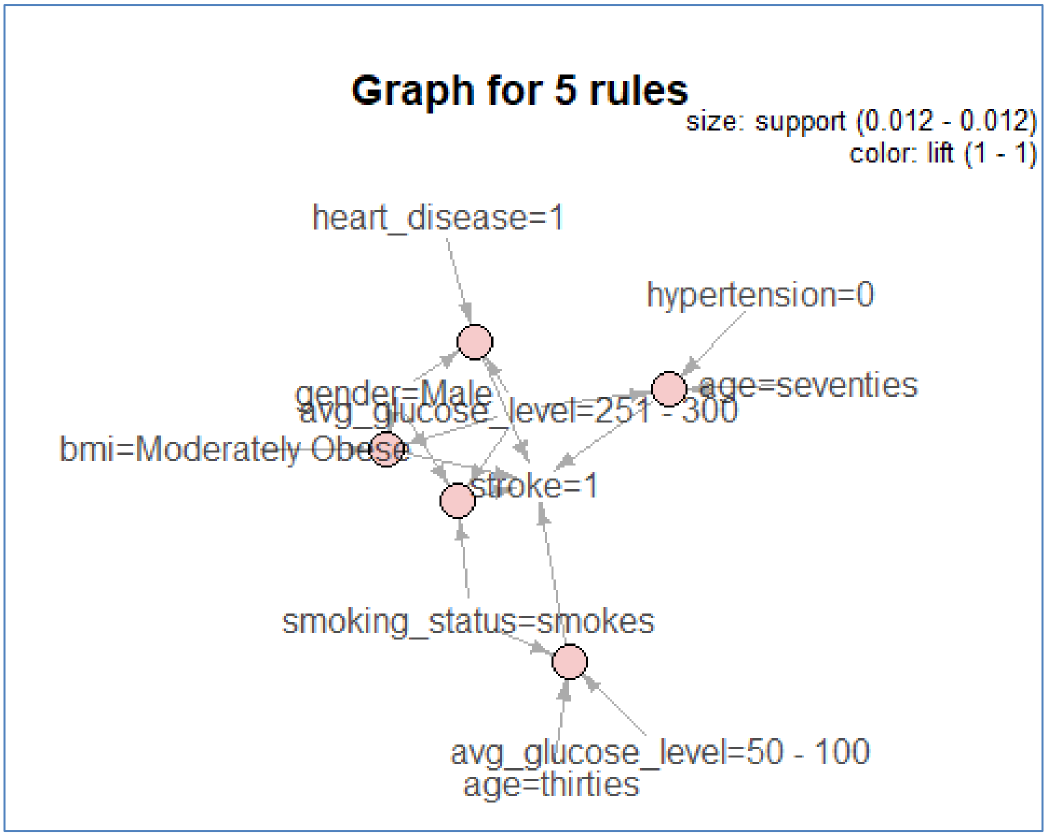
Inspect the Rules

The rules generated include “stroke=1” on the right-hand side, thus the items on the left-hand side are usually associated with patients who have suffered a stroke. Males, high glucose levels (251-300 and smoking are characteristics usually seen in stroke patients, according to the apriori algorithm. However, the count of patients that is the basis of these rules are quite low. In a data set of over 5000 patients (201 who suffered a stroke), only 3 are accounted for to generate rules, which is a low number.

Figure 27: Inspect the Rules

```
inspect(stroke_rules_2[1:10])

##      lhs                                rhs           support confidence   c
overage lift count
## [1]  {gender=Male,
##       avg_glucose_level=251 - 300,
##       smoking_status=smokes}       => {stroke=1} 0.01204819          1 0.0
1204819    1     3
## [2]  {gender=Male,
##       heart_disease=1,
##       avg_glucose_level=251 - 300} => {stroke=1} 0.01204819          1 0.0
1204819    1     3
## [3]  {gender=Male,
##       avg_glucose_level=251 - 300,
##       bmi=Moderately Obese}        => {stroke=1} 0.01204819          1 0.0
1204819    1     3
## [4]  {age=seventies,
##       hypertension=0,
##       avg_glucose_level=251 - 300} => {stroke=1} 0.01204819          1 0.0
1204819    1     3
## [5]  {age=thirties,
##       avg_glucose_level=50 - 100,
##       smoking_status=smokes}       => {stroke=1} 0.01204819          1 0.0
1204819    1     3
## [6]  {gender=Female,
##       age=thirties,
##       smoking_status=smokes}       => {stroke=1} 0.01204819          1 0.0
1204819    1     3
## [7]  {age=thirties,
##       heart_disease=0,
##       smoking_status=smokes}       => {stroke=1} 0.01204819          1 0.0
1204819    1     3
## [8]  {age=thirties,
##       avg_glucose_level=50 - 100,
##       bmi=Overweight}              => {stroke=1} 0.01204819          1 0.0
1204819    1     3
## [9]  {gender=Female,
##       age=thirties,
##         avg_glucose_level=50 - 100}  => {stroke=1} 0.02008032          1 0.0
2008032    1     5
## [10] {age=thirties,
##         hypertension=0,
##         avg_glucose_level=50 - 100}  => {stroke=1} 0.01606426          1 0.0
1606426    1     4
```

Figure 28: Association Rules Plot for Stroke Patients



The plot above displays the top 10 rules for stroke patients generated by the apriori algorithm.

Most Common Stroke Rules

The apriori algorithm was used to explore associations related to the “stroke” attribute to determine what type of patients are more likely to suffer a stroke in their lifetime. To do this, the right-hand side of the rule was set to stroke.

1. *{gender = Male, avg_glucose_level = 251-300, bmi = Moderately Obese} => {stroke = 1}*
   - Support: 0.012
   - Confidence: 1

- Lift: 1

2. *{gender = Female, age = thirties, avg_glucose_level = 50 - 100} => {stroke = 1}*
    - Support: 0.02
    - Confidence: 1
    - Lift: 1
    - Younger patients are not usually faced with the possibility of a stroke, especially if they are seemingly in good health.
3. *{age = seventies, hypertension = 0, avg_glucose_level = 251-300} => {stroke = 1}*
    - Support: 0.012
    - Confidence: 1
    - Lift: 1
4. *{heart_disease = 1, bmi = severely obese} => {stroke = 1}*
    - Support: 0.0018
    - Confidence: 0.3333
    - Lift: 6.84
    - It is not too surprising that heart disease and obesity can be contributors to the occurrence of a stroke. However, the confidence shows that these two factors are not always associated with strokes. Then again, there only 201 stroke patients in the data, so, there was not much to work with.
5. *{age = eighties, bmi = Overweight} => {stroke = 1}*
    - Support: 0.00235
    - Confidence: 0.2667
    - Lift: 5.4726

Decision Trees (*rpart*)

Decision trees visually and explicitly represent decisions and decisions making in a tree-like model. As a structure, a decision tree includes a root node, branches, and leaf nodes. Rpart is a supervised learning algorithm that can be used to predict both regression and classification problems.

Decision Trees create a classification model to predict the class or value of the target variable by learning simple decision rules inferred from the training data. The target variable must be a factor. Rpart requires a formula, the data frame that is being used and the method. The method for classification tasks is "class". Rpart creates a decision tree model to predict the classification of entries in the data frame.

Preprocessing

A new data frame was created from the original stroke prediction data set.

```
stroke.dt <- stroke_prediction
```

Next the data was split into training and testing sets. These sets will be used towards training and testing all the prediction models going forward.

```
set.seed(25)

train_split  <- sample(nrow(stroke.dt), nrow(stroke.dt) * 2/3)
train_stroke <- stroke.dt[train_split,]
test_stroke  <- stroke.dt[-train_split,]
```

As mentioned before, there is a large class imbalance in the stroke data. The goal is to predict the likelihood of stroke from different factors. However, given that most patients in the data have not suffered a stroke, it will be challenging to train a model to predict the occurrence of a stroke given so little instances of it occurring, especially after the data has been split into different sets.

To remedy this, the ubSMOTE() function from the unbalanced package was used on the training data. The ubSMOTE () function handles unbalanced classification problems by implementing SMOTE (synthetic minority over-sampling technique)[2]. SMOTE is an oversampling technique that generates synthetic samples from the minority class. It is used to obtain a synthetically class-balanced or nearly class-balanced training set, which is then used to train the classifier. The ubSMOTE() function input variables of the unbalanced data set, the response variable of the unbalanced data set (in this case, stroke), the number of new instances generated for each rare instance (perc. over), the number of nearest neighbors (k), and the number of majority class instances that are randomly selected for each smoted observation (perc.under). The response variable must be a factor.

The code below applies the ubSMOTE() function on the training set with perc.over = 200, k = 5, and perc.under = 200.

```
smoteD       <- ubSMOTE(train_stroke, train_stroke$stroke, perc.over = 200, k
= 5, perc.under = 200)

train_stroke <- cbind(smoteD$X, smoteD$Y)
```

[2] https://cran.r-project.org/web/packages/unbalanced/unbalanced.pdf

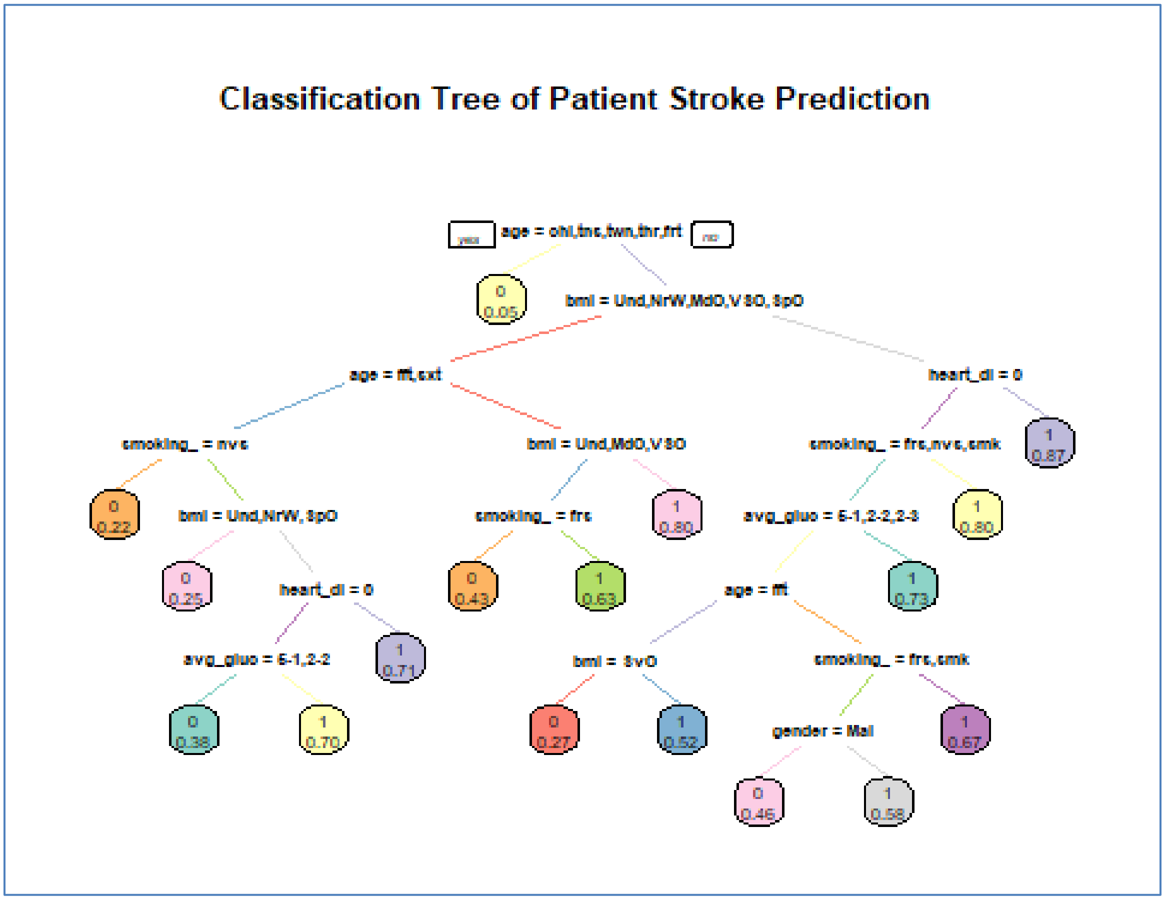
```
# Remove the extra column
train_stroke <- train_stroke[,-9]
```

Decision Tree Model 1

The first decision tree model utilized all the predictor variables.

```
stroke.tree <- rpart(stroke ~ ., data = train_stroke
                     , method = 'class'
                     , parms = list(split = 'gender')
                     , minsplit = 50, cp = 0 )
```

Figure 29: Classification Tree for the First Decision Tree Model



The minsplit was 50 and cp was set to 0.

Test the Accuracy of the Model

```
##               Predicted Class
## Actual Class    0    1
##             0 1205  416
##             1   21   62

err.tree <- sum(test_stroke$stroke != pred.tree)/nrow(test_stroke)
accuracy.tree <- round((1 - err.tree) * 100, 2)
accuracy.tree

## [1] 74.35
```

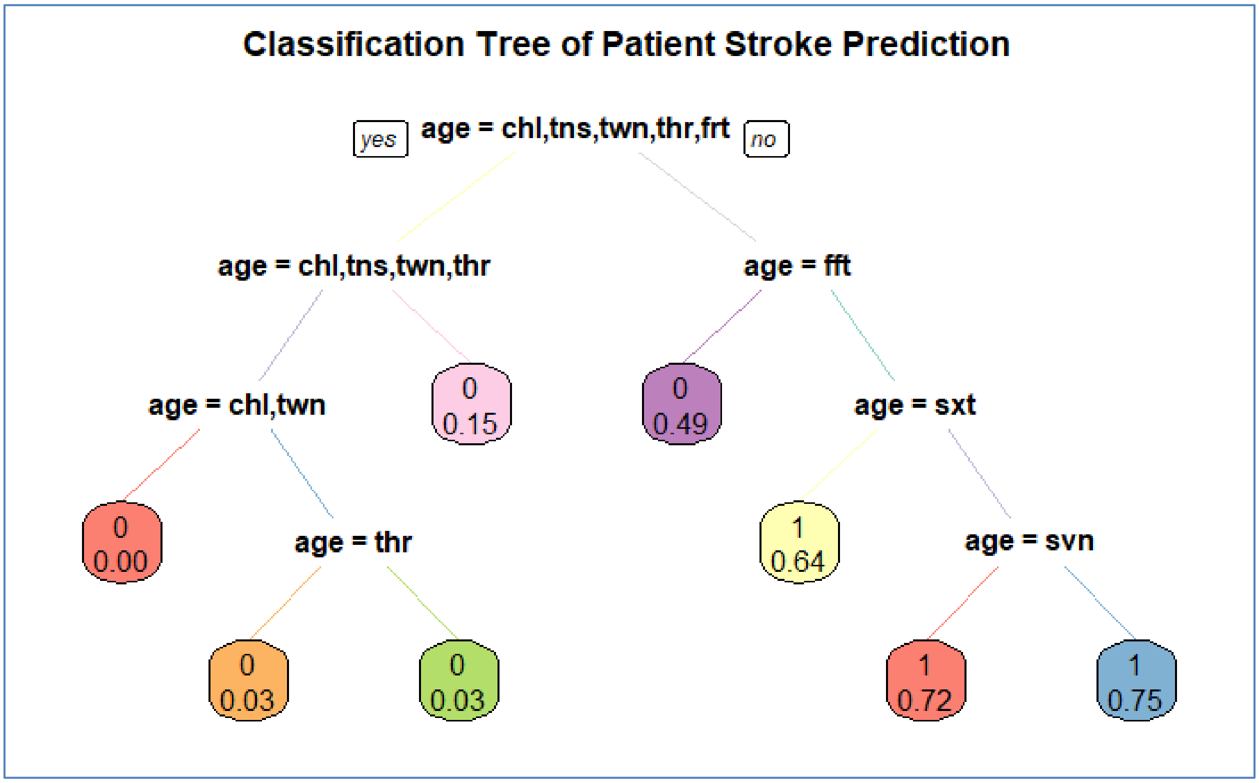
The decision tree had an overall accuracy of 74.35% when predicting stroke.

Decision Tree Model 2

The second decision tree model only uses age as the predictor variable as it was the variable that had the highest correlation with stroke, as shown by the correlation map from before.

```
Pc <- proc.time()

stroke_2.tree <- rpart(stroke ~ age, data = train_stroke
                       , method = ‘class’
                       , control = rpart.control(minbucket = 5, minsplit = 40
, cp = -1)
                       )
```

Figure 30: Classification Tree for the Second Decision Tree Model



The minsplit for this tree is 40 and the minbucket is 5. The cp was set to -1.

```
##             Predicted Class
## Actual Class    0    1
##            0 1222  399
##            1   23   60

err_2.tree <- sum(test_stroke$stroke != pred.tree)/nrow(test_stroke)
accuracy <- round((1 - err_2.tree) * 100, 2)
accuracy

## [1] 75.23
```

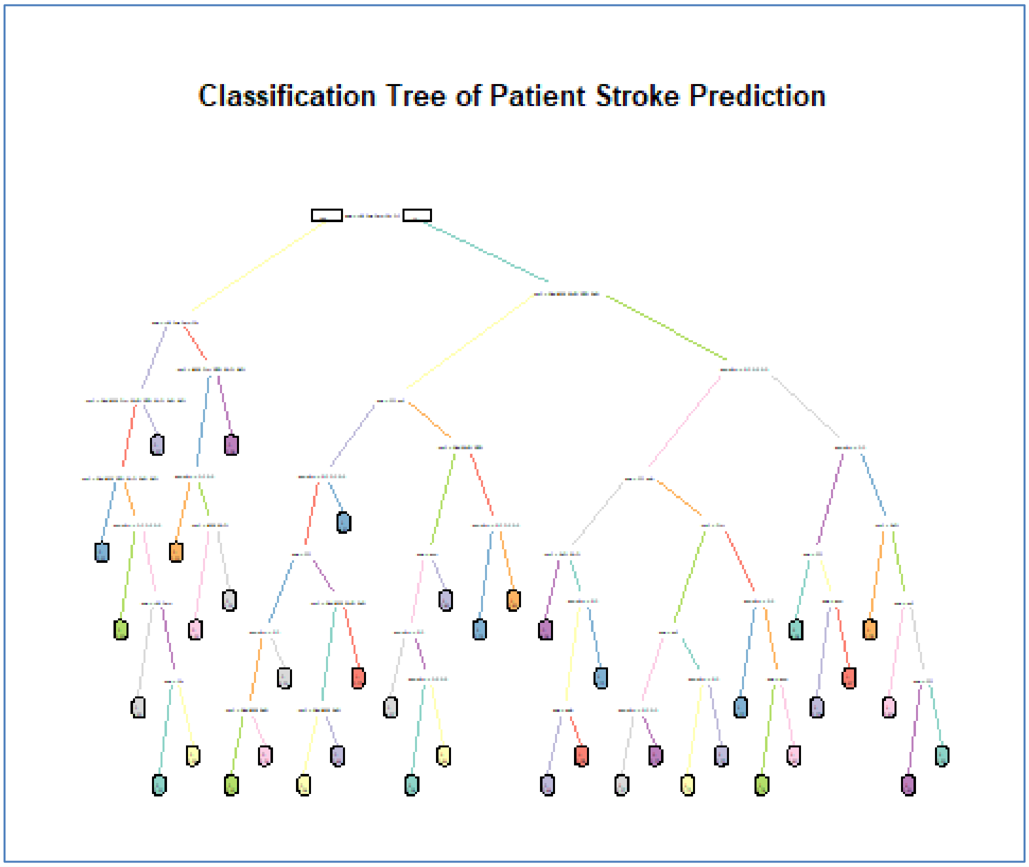
This has a 75.23% accuracy for predicting stroke, which is a slight increase from the first model that used all 7 predictors.

Decision Tree Model 3

The third decision tree model uses age, hypertension, and bmi to predict stroke. The minbucket, minsplit, and cp were kept the same as the second model.

```
stroke_3.tree <- rpart(stroke ~ age + hypertension + bmi
                       , data = train_stroke
                       , method = 'class'
                       , control = rpart.control(minbucket = 5, minsplit = 40
, cp = -1)
                       )
```

Figure 31: Classification Plot for the Third Decision Tree Model



```
##              Predicted Class
## Actual Class    0    1
##             0 1213  408
##             1   25   58

err_3.tree <- sum(test_stroke$stroke != pred.tree)/nrow(test_stroke)
accuracy <- round((1 - err_3.tree) * 100, 2)
accuracy

## [1] 74.18
```

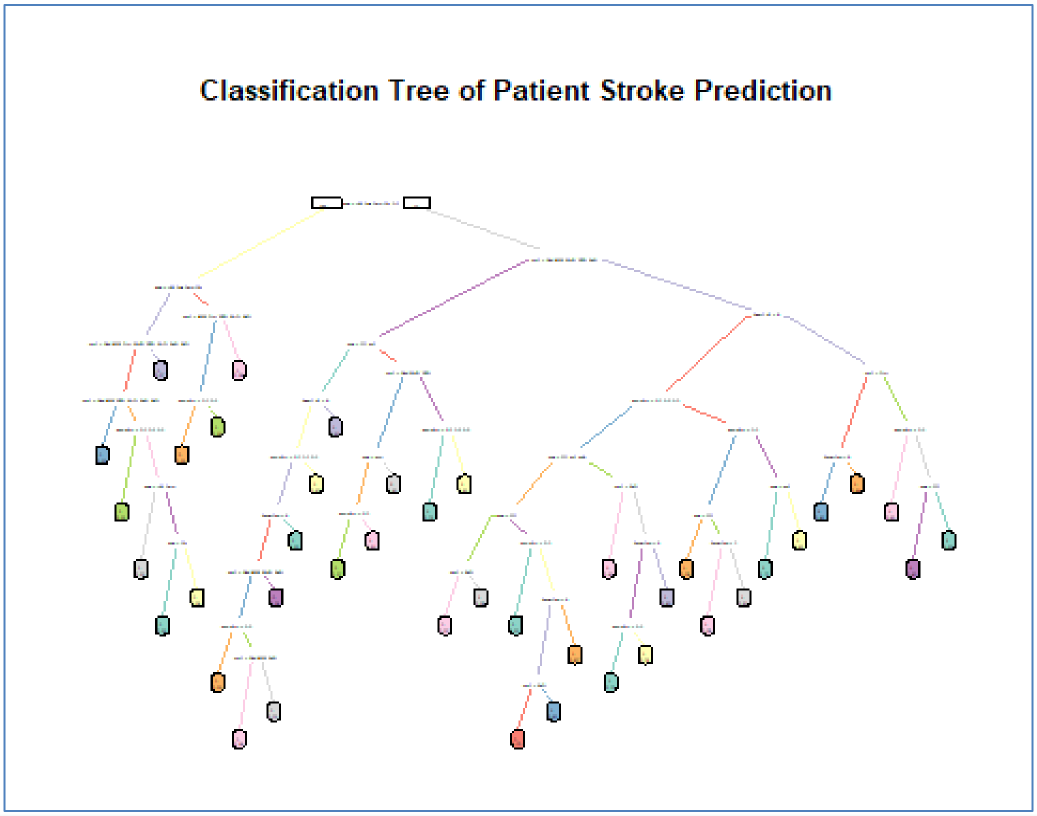
This model had a 74.18% accuracy after adding additional variables. This is a decrease from the last model.

Decision Tree Model 4

The fourth and final decision tree model added average glucose and heart disease to the previous r part algorithm. The min bucket and cp were kept the same as the previous two model, but min split is set to 50.

```
stroke_4.tree <- rpart(stroke ~ age + avg_glucose_level + bmi + heart_disease
+ hypertension
                       , data = train_stroke
                       , method = 'class'
                       , control = rpart.control(minbucket = 5, minsplit = 50
, cp = -1)
                       )
```

Figure 32: Classification Tree for the Fourth Decision Tree Model



```
##               Predicted Class
## Actual Class    0    1
##             0 1216  405
##             1   22   61

err_4.tree <- sum(test_stroke$stroke != pred.tree)/nrow(test_stroke)
accuracy <- round((1 - err_4.tree) * 100, 2)
accuracy

## [1] 74.94
```

This model has a 74.94% accuracy for predicting stroke which is a slight increase from model 3 but still lower than the second model that only used age as its predictor.

Naive Bayes Classifier (*naiveBayes*)

Naïve Bayes is a supervised machine learning classification technique that derives from the Bayes Theorem. The algorithm calculates the probability of the event happening. The Bayes Theorem (shown below) describes the probability of an event occurring based on prior knowledge of events related to that event.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$ [3]

In terms of machine learning:

- P(A|B) - **Posterior Probability**
  - The conditional probability of the target variable given the inputs of the training data.
- P(A) - **Prior Probability**
  - The probability of the target variable.
- P(B) - **Evidence**
  - The probability of the training data.
- P(B|A) - **Likelihood**
  - The conditional probability of the training data given the target variable.

There are multiple libraries that can be used for the naïve bayes technique. The library used in the analysis is “e1071”. The models have the same requirements; however, the plotting options are different. The target variable must be a factor. Naïve bayes requires a formula and the data frame that is being used.

Naïve Bayes Model 1

The Naïve Bayes algorithm is applied to the training set and uses all variables to build the first model.

```r
stroke.nb <- naiveBayes(stroke ~ ., data = train_stroke)
```

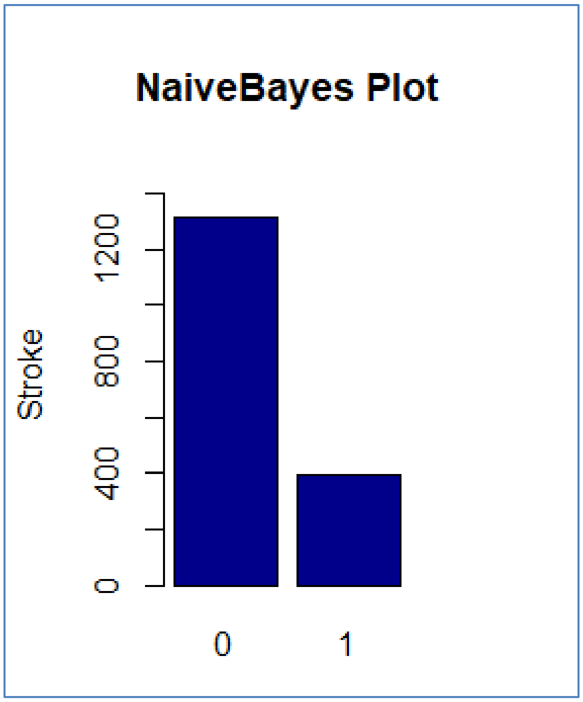
The prediction function is applied on the test data to predict stroke.

```r
# prediction function predicts the test target value from the model generated from train set.
pred.nb <- predict(stroke.nb, newdata = test_stroke, type = "class")
```

[3] https://www.kdnuggets.com/2019/04/naive-bayes-baseline-model-machine-learning-classification-performance.html

```
table(`Actual Class` = test_stroke$stroke, `Predicted Class` = pred.nb)

##               Predicted Class
## Actual Class    0    1
##             0 1282  339
##             1   28   55
```

Figure 33: Naïve Bayes Prediction Plot (*model 1*)



```
err.nb <- sum(test_stroke$stroke != pred.nb)/nrow(test_stroke)
accuracy <- round((1 - err.nb) * 100, 2)
accuracy

## [1] 78.46
```

The first model has a 78.46% accuracy for prediction the occurrence of stroke.

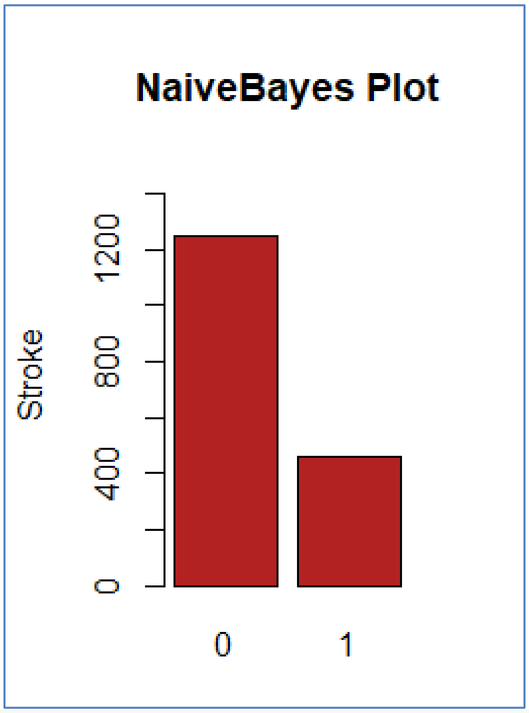
Naïve Bayes Model 2

The age variable is used to build the second Naïve Bayes model.

```
Stroke_2.nb <- naiveBayes(stroke ~ age
                          , data = train_stroke
                          , laplace = 1)
```

```
##             Predicted Class
## Actual Class    0    1
##            0 1222  399
##            1   23   60
```

Figure 34: Naïve Bayes Prediction Plot (*model 4*)



```
err_2.nb <- sum(test_stroke$stroke != pred_2.nb)/nrow(test_stroke)
accuracy <- round((1 - err_2.nb) * 100, 2)
accuracy

## [1] 75.23
```

This model has a 75.23% accuracy for predicting stroke, which is a decrease in accuracy from the first model but not too low for a model that only used one predictor.

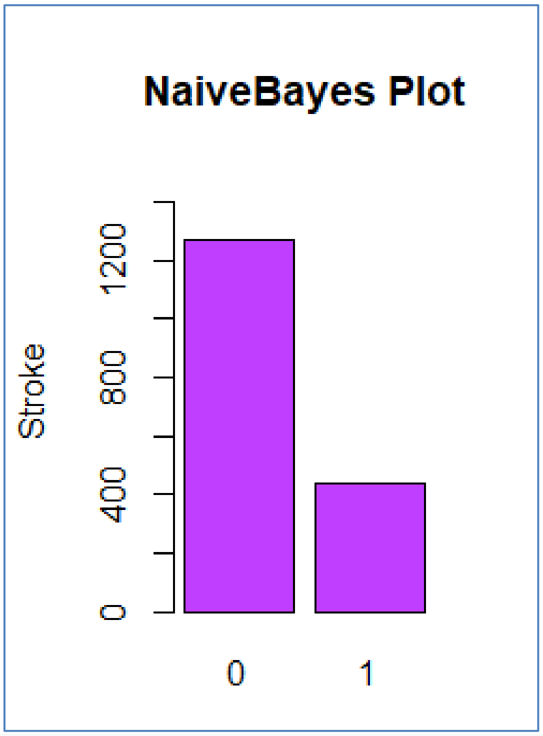
Naïve Bayes Model 3

The third Naïve Bayes Model uses age, average glucose level, smoking status, and heart disease to predict the occurrence of stroke in patients.

```
stroke_3.nb <- naiveBayes(stroke ~ age + avg_glucose_level + smoking_status + heart_disease
                          , data = train_stroke
                          , laplace = 1)
```

```
##              Predicted Class
## Actual Class    0    1
##             0 1246  375
##             1   23   60
```

Figure 35: Naïve Bayes Prediction Plot (*model 3*)



```
err_3.nb <- sum(test_stroke$stroke != pred_3.nb)/nrow(test_stroke)
accuracy <- round((1 - err_3.nb) * 100, 2)
accuracy

## [1] 76.64
```

This model has a 76.64% accuracy for predicting stroke which is a slight increase from the second model but lower than the first model, which uses all variables.

Random Forest (*randomForest*)

Random forest is a supervised ensemble learning algorithm that is used for both classification and regression problems[4]. Random forest trains multiple decision trees and then combines their results into one final model. The random forest algorithm requires a formula and the training data frame.

[4] https://www.kdnuggets.com/2020/01/random-forest-powerful-ensemble-learning-algorithm.html

In the analysis, the number of predictor variable used changes with the creation of a new model to determine which variables are significant in predicting strokes.

Random Forest: Model 1

The random forest algorithm is applied to the stroke training data set and uses all the variable to build the first random forest model.

```
rf.model <- randomForest(stroke ~ .
                         , data = train_stroke
                         , proximity = TRUE)
```

Prediction

```
pred.forest <- predict(rf.model, newdata = test_stroke, type = "class")
table('Actual Class' = test_stroke$stroke, 'Predicted Class' = pred.forest)

##             Predicted Class
## Actual Class    0    1
##            0 1282  339
##            1   33   50

# Calculate the error rate.
err.rf   <- sum(test_stroke$stroke != pred.forest) / nrow(test_stroke)
accuracy <- round((1 - err.rf) * 100, 2)
accuracy

## [1] 78.17
```

The first random forest model has a 78.17% accuracy in predicting the occurrence of a stroke in patients.

Random Forest: Model 2

The second model uses the age of patients as its only predictor of stroke.

```
rf.model2 <- randomForest(stroke ~ age
                         , data = train_stroke
                         , proximity = TRUE)
```

Prediction

```
##             Predicted Class
## Actual Class    0    1
##            0 1222  399
##            1   23   60
```

Test the Accuracy

```
# Calculate the error rate.
err_2.rf   <- sum(test_stroke$stroke != pred_2.forest) / nrow(test_stroke)
accuracy   <- round((1 - err_2.rf) * 100, 2)
accuracy

## [1] 75.23
```

This model gas a 75.23% accuracy for predicting stroke, which is a decrease in accuracy from the previous model, but not much. It is a possibility the accuracy can increase if a few model variables are passed to the algorithm.

Random Forest: Model 3

The third algorithm builds on the second random forest model by adding predictors to age. Six of the seven predictor variables were used to build this model. Smoking status was excluded because it had a weak, negative correlation with stroke.

```
rf.model3 <- randomForest(stroke ~ age + gender + bmi + hypetension +
avg_glucose_level + heart_disease
                         , data = train_stroke
                         , proximity = TRUE)
```

Prediction

```
pred_3.forest <- predict(rf.model3, newdata = test_stroke, type = "class")
table('Actual Class' = test_stroke$stroke, 'Predicted Class' = pred_3.forest)

##              Predicted Class
## Actual Class    0    1
##            0 1258  363
##            1   25   58
```

Test the Accuracy

```
# Calculate the error rate.
err_3.rf   <- sum(test_stroke$stroke != pred_3.forest) / nrow(test_stroke)
accuracy   <- round((1 - err_3.rf) * 100, 2)
accuracy

## [1] 77.23
```

The third random forest model has a 77.23% accuracy for predicting the occurrence of stroke. This model is more accurate than the second model but slightly leas accurate than the first.

Random Forest: Model 4

The fourth and final random forest model draws a sample of 20 patients from the training data and specific the number of trees used is equal to 10.

```
rf.model4 <- randomForest(train_stroke
                          , train_stroke$stroke
                          , sampsize = 20 , ntree = 10)
```

Prediction

```
pred_4.forest <- predict(rf.model4, newdata = test_stroke, type = "class")
table('Actual Class' = test_stroke$stroke, 'Predicted Class' = pred_4.forest)

##               Predicted Class
## Actual Class    0    1
##              0 1555   66
##              1    1   82
```

Test the accuracy

```
# Calculate the error rate.
err_4.rf   <- sum(test_stroke$stroke != pred_4.forest) / nrow(test_stroke)
accuracy <- round((1 - err_4.rf) * 100, 2)
accuracy

## [1] 96.07
```

The fourth random tree model has a 96.07% accuracy for predicting the occurrence of stroke within patients. This is the highest accuracy produced from a model.

Algorithm Performance Comparison

Decision Trees

Decision Tree Model 1 (all variables, minsplit = 50, cp = 0): **74.35% Accuracy**

Decision Tree Model 2 (age, minbucket = 5, minsplit = 40, cp = -1): **75.23% Accuracy**

Decision Tree Model 3 (age + hypertension + bmi, minbucket = 5, minsplit = 40, cp = -1): **74.18% Accuracy**

Decision Tree Model 4(age + avg_glucose_level + bmi + heart_disease + hypertension, minbucket = 5, minsplit = 50, cp = -1): **74.94% Accuracy**

Naïve Bayes

Naïve Bayes Model 1 (all variables): **78.46% Accuracy**

Naïve Bayes Model 2 (age, laplace = 1): **75.23% Accuracy**

Naïve Bayes Model 3 (age, avg_glucose_level, smoking_status, heart_disease): **76.64% Accuracy**

Random Forest

Random Forest Model 1 (all variables): **78.17% Accuracy**

Random Forest Model 2 (age): **75.23% Accuracy**

Random Forest Model 3 (age + gender + bmi + hypertension + avg_glucose_level + heart_disease): **77.23% Accuracy**

Random Forest Model 4 (sampsize = 20, ntree = 10): **96.07%**

All three algorithms gave high degrees of accuracy, with the lowest accuracy (74.18%) coming from the Decision Tree algorithm. Random Forest presented the highest accuracy (96.07%) when it took a sample of 20 patients from the training data and ntree was set to 10. In all algorithms used the age variable as the only predictor to build a model and each time the accuracy was the same (75.23%). Other times, variables were included or removed to find a model that will give the highest accuracy for predicting the occurrence of stroke within patients. The decision tree algorithm gave its highest accuracy when age was the only predictor, but Naïve Bayes and Random Forest presented its lowest accuracy. When more variables were passed through the algorithm, the accuracy of the models gradually started to increase (or decrease) by small percentages but never broke beyond 80%, except for when extra work was done on the training data. This demonstrates that all the variables used in the analysis are good predictors of stroke but not enough. The data should include more factors such as stress level, family health history, or high blood pressure to help build stronger prediction models.

Conclusion

Stroke prevention and its related risk factors has been a major public health concern worldwide. Several risk factors can enhance a person's chances of having a stroke. The two most important risk factors for stroke are age and gender. Recent clinical trials have shown that identifying high-risk patients and treating them for primary prevention can significantly reduce the chance of a stroke, providing a basis for identifying high risk people and implementing lifestyle invention techniques. Using machine learning techniques may provide individualized predictions for patients at risk for stroke. Machine learning is a class of computer algorithms that can automatically learn from data without explicit programming. Some initial studies have shown that machine learning can be used to predict stroke lesions.

This study found that age has a high correlation to determine the likelihood that a patient will have stroke or had a stroke. Using Association rule mining, this technique determined that a target patient with a moderate bmi, who is a male in his thirties or seventies with an average glucose level between 50-100 or 251-300, smoker, and hypertension are more likely to have a stroke or had a stroke. After further assessment, the occurrence of a patient having a stroke is also probable if a patient has a heart condition specifically heart disease. Other variables such as where a patient lives, if a patient is married or not, and where a patient works had no correlation to a patient having a stroke. Patients who are younger with no history of hypertension and heart disease were least likely to have a stroke no matter the gender. According to the dataset, females are prone to have a stroke more than males. Further variables needed to determine exactly why females are more likely to have strokes than males.

The prediction of long-term outcomes in stroke patients may be useful in treatment decisions. Several prognostic systems have been developed for this purpose. Application of machine learning in the medical field has given promising results, thanks to recent breakthroughs in the discipline. In many cases, the complicated and unpredictable aspect of human physiology has been better characterized by machine learning techniques versus traditional. Using machine learning can help predict long-term stroke outcomes and may help prevent a patient from having a stroke based on certain variables such as age, gender, medical history, and lifestyle choices. However, in this study more research is needed to confirm the findings in larger, more diverse datasets and to integrate critical clinical factors into the model to improve its effectiveness. Other models can help contribute to improving stroke prediction as well such as deep neural networks.