PREDICTIVE MODELING FOR STROKE RISK FACTORS

Group 1 - Niloofar Mirzadzare, Monica Chandramurthy, Ashique Jaman, Ali Haghighat

Introduction

Stroke, a medical emergency characterized by a sudden loss of brain function due to either a rupture or blockage of blood vessels in the brain, is a significant global health concern [1], It exhibits a range of symptoms, including facial drooping, confusion, vision loss, and severe headaches. As per the World Stroke Organization's data, approximately 12.2 million stroke cases are recorded annually worldwide, with nearly 50% resulting in mortality [2]. In Canada, stroke is recognized as the third leading cause of death and a principal source of long-term disabilities [1]. The risk factors for stroke are multifactorial and include hypertension, cardiovascular complications, age, alcohol and tobacco use, obesity, and genetic predisposition [3]. A comprehensive understanding of these physical and social risk factors can facilitate the identification of modifiable factors and potentially reduce stroke risk in susceptible individuals.

Objective

Using the "Stroke Prediction Dataset" available on Kaggle, our primary goal for this project is to delve deeper into the risk factors associated with stroke. We aim to identify the factors that contribute most significantly to the likelihood of a person experiencing a stroke. This includes exploring how demographic and lifestyle factors, such as age, sex, medical history, physical condition, work status, and place of residence, influence the risk of stroke. Additionally, the data collected from patients who have previously experienced a stroke at least once in their lifetime will serve as a basis for developing a predictive model. Our objective is to test the reliability of this model in predicting stroke risk and gauge its accuracy. We hope this study will ultimately help determine the likelihood of similar individuals with comparable conditions experiencing a stroke in the future.

Dataset Overview

The "Stroke Prediction Dataset" is a comprehensive collection of health and lifestyle data from patients who have previously experienced a stroke. This dataset includes 5110 observations and 12 variables such as sex, age, hypertension and heart disease status, marital and work status, residence type, average glucose level, body mass index (BMI), smoking habits, and stroke history. The dataset is publicly available on Kaggle for educational purposes and can be accessed here [4].

Variables in the dataset:

- Categorical Variables (Binary): sex, hypertension, heart_disease, ever_married, stroke
- Categorical Variables (Multiple Categories): work_type, Residence_type, smoking_status

- Quantitative Variables: id, age, avg_glucose_level, bmi

In our study, ‘stroke’ is the response variable, while all other variables listed above serve as explanatory variables as we aim to predict the probability of a stroke based on these parameters.

Overall Methodology

Our project will apply various statistical and machine learning concepts to fulfill its objectives. The process begins with data cleaning and transformation to ensure a workable dataset. We then employ different sampling methods to compare population and sample statistics, providing us with insights into the representativeness of our sample. Our analytical approach will leverage generalized linear regression and Linear Discriminant Analysis (LDA) to estimate the probability of stroke occurrence. We will assess the robustness and accuracy of our model using K-fold cross-validation. We will also explore the decision tree method, comparing its outcomes with the results from other analytical techniques. Other statistical tools like contingency tables and generalized linear models with varying link functions will be used throughout this project to further our understanding of stroke predictors.

Install.packages

```
#install.packages('')
```

Dateset and Importing neccessary libraries

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affec
ted by this.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(survey)
```

```
## Loading required package: grid
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survey'
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(mlbench)
library(sampling)
```

```
##
## Attaching package: 'sampling'
```

```
## The following objects are masked from 'package:survival':
##
##     cluster, strata
```

```
library(klaR)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following objects are masked from 'package:mosaic':
##
##     deltaMethod, logit
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(QuantPsyc)
```

```
## Loading required package: boot
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
## The following object is masked from 'package:survival':
##
##     aml
```

```
## The following object is masked from 'package:mosaic':
##
##     logit
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
## Loading required package: purrr
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:car':
##
##     some
## The following object is masked from 'package:mosaic':
##
##     cross
```

```
##
## Attaching package: 'QuantPsyc'
```

```
## The following object is masked from 'package:Matrix':
##
##     norm
```

```
## The following object is masked from 'package:base':
##
##     norm
```

```
library(energy)
library(vcd)
```

```
##
## Attaching package: 'vcd'
```

```
## The following object is masked from 'package:mosaic':
##
##     mplot
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:mosaic':
##
##     cov, var
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(tree)
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
## The following object is masked from 'package:sampling':
##
##     cluster
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
## The following object is masked from 'package:mosaic':
##
##     dotPlot
```

```
library(boot)
library(questionr)
```

```
##
## Attaching package: 'questionr'
```

```
## The following object is masked from 'package:mosaic':
##
##     prop
```

```
library(fmsb)
```

```
## Registered S3 methods overwritten by 'fmsb':
##   method    from
##   print.roc pROC
##   plot.roc  pROC
```

```
## 
## Attaching package: 'fmsb'
```

```
## The following object is masked from 'package:pROC':
## 
##     roc
```

```
## The following object is masked from 'package:vcd':
## 
##     oddsratio
```

Data Preprocessing

In this phase, the stroke dataset is imported and initially examined to understand its structure. The dataset consists of 5110 rows (entries) and twelve columns (features), with columns representing a mix of quantitative and qualitative attributes, including a unique identifier ‘id’, and the response variable ‘stroke’.

```
stroke_data_raw=read.csv("healthcare-dataset-stroke-data.csv")
```

Dimension and Summary of the Dataset

```
#Checking the dimension of the dataset and column names
cat("This dataset contains",dim(stroke_data_raw)[1],"rows",dim(stroke_data_raw)[2],"c
olumns\n\n")
```

```
## This dataset contains 5110 rows 12 columns
```

```
cat("The Columns names are:\n\n")
```

```
## The Columns names are:
```

```
names(stroke_data_raw)
```

```
##  [1] "id"                "gender"            "age"              
##  [4] "hypertension"      "heart_disease"     "ever_married"     
##  [7] "work_type"         "Residence_type"    "avg_glucose_level"
## [10] "bmi"               "smoking_status"    "stroke"
```

The dataset consists of 5110 rows (entries) and twelve columns (features), with columns representing a mix of quantitative and qualitative attributes, including a unique identifier ‘id’, and the response variable ‘stroke’

Checking for Missing Values

```
#Finding NA values in the dataset

NA_count <- colSums(is.na(stroke_data_raw))

# Displaying the result
print(NA_count)
```

```
##                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
```

The dataset was then checked for missing values. No missing (NA) values were found, but a closer inspection of the ‘bmi’ column revealed that 201 entries were marked as ‘N/A’, representing about 4% of the dataset’s entries.

```
for (col in names(stroke_data_raw)){
  cat("The number of N/A values in column",col,"is", sum((stroke_data_raw[,col]=='N/A
')),"\n")
}
```

```
## The number of N/A values in column id is 0
## The number of N/A values in column gender is 0
## The number of N/A values in column age is 0
## The number of N/A values in column hypertension is 0
## The number of N/A values in column heart_disease is 0
## The number of N/A values in column ever_married is 0
## The number of N/A values in column work_type is 0
## The number of N/A values in column Residence_type is 0
## The number of N/A values in column avg_glucose_level is 0
## The number of N/A values in column bmi is 201
## The number of N/A values in column smoking_status is 0
## The number of N/A values in column stroke is 0
```

Notably, these entries included 40 cases of stroke, a significant count given the total number of stroke cases (209). To retain these cases for analysis, ‘N/A’ entries in the ‘bmi’ column were replaced with the mean ‘bmi’ value across the dataset, assuming a normal distribution for this attribute.

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

```
## Warning: NAs introduced by coercion
```

To preserve these valuable stroke instances, we opted to impute missing ‘bmi’ values. The ‘bmi’ column, reflecting individuals’ body mass index – a measure based on their height and weight – was assumed to follow a normal distribution naturally. This assumption motivated the decision to replace missing ‘bmi’ values with the mean ‘bmi’ across all existing data.

Data wrangling for gender column:

In regards to gender, there were three unique labels: ‘Female’, ‘Male’, and ‘Other’. However, the ‘Other’ category contained only one record. With such limited data, it wouldn’t be beneficial to predict stroke occurrence for ‘Other’ gender, hence this single instance was removed from the dataset, leaving only ‘Female’ and ‘Male’ gender categories for subsequent analysis.

```
mean_value <- mean(stroke_data_raw$bmi, na.rm = TRUE)
stroke_data_raw$bmi[is.na(stroke_data_raw$bmi)] <- mean_value
```

Exploratory Data Analysis

After the preprocessing stage, an exploration of the categorical variables was conducted. Each categorical column was analyzed to identify the unique categories and their respective counts.

```
table(stroke_data_raw$gender)
```

```
##
## Female   Male  Other
##   2994   2115      1
```

```
stroke_data_raw <- subset(stroke_data_raw, gender != 'Other')
table(stroke_data_raw$gender)
```

```
##
## Female   Male
##   2994   2115
```

```
categorical_columns=c("gender","hypertension","heart_disease",
                      "ever_married","work_type","Residence_type","smoking_status","s
troke")
for (col in categorical_columns){
  cat("\n\nSummary of categories in",col,"is:\n")
  print(table(stroke_data_raw[,col]))
}
```

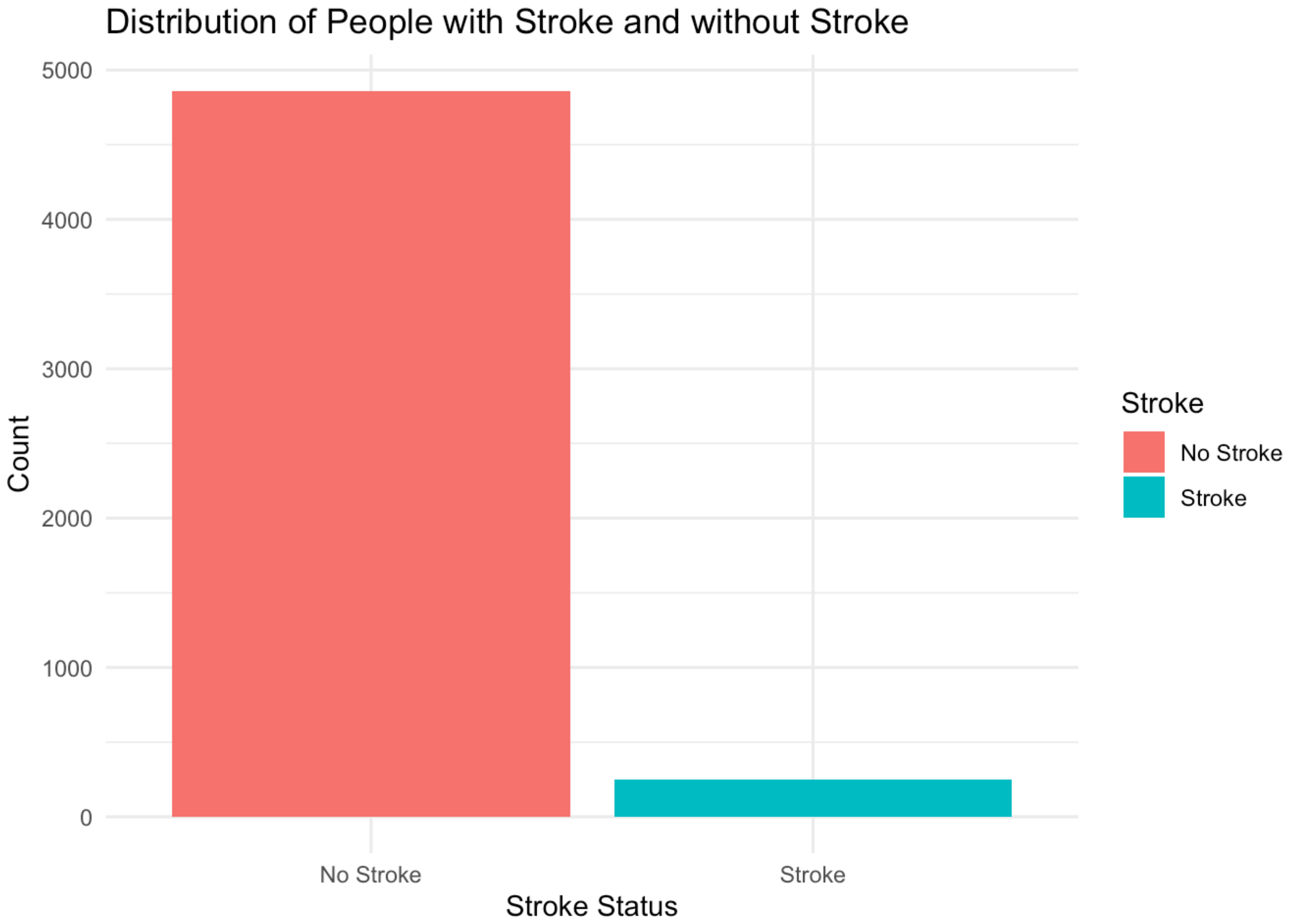
```
##
##
## Summary of categories in gender is:
##
## Female   Male
##   2994   2115
##
##
## Summary of categories in hypertension is:
##
##    0    1
## 4611  498
##
##
## Summary of categories in heart_disease is:
##
##    0    1
## 4833  276
##
##
## Summary of categories in ever_married is:
##
##   No  Yes
## 1756 3353
##
```

```
##
## Summary of categories in work_type is:
##
##      children      Govt_job  Never_worked       Private Self-employed
##           687           657            22          2924           819
##
##
## Summary of categories in Residence_type is:
##
## Rural Urban
##  2513  2596
##
##
## Summary of categories in smoking_status is:
##
## formerly smoked    never smoked          smokes         Unknown
##             884            1892             789            1544
##
##
## Summary of categories in stroke is:
##
##    0    1
## 4860  249
```

Based on teh above results, binary variables like ‘hypertension’, ‘heart_disease’, and ‘stroke’ used 0 and 1 to represent absence and presence of the condition. For the remaining categorical columns, the frequency of each category was summarized, providing insight into the distribution of these variables in the dataset.

Visualizing and Interpreting Variable Distributions in Relation to Stroke Incidence

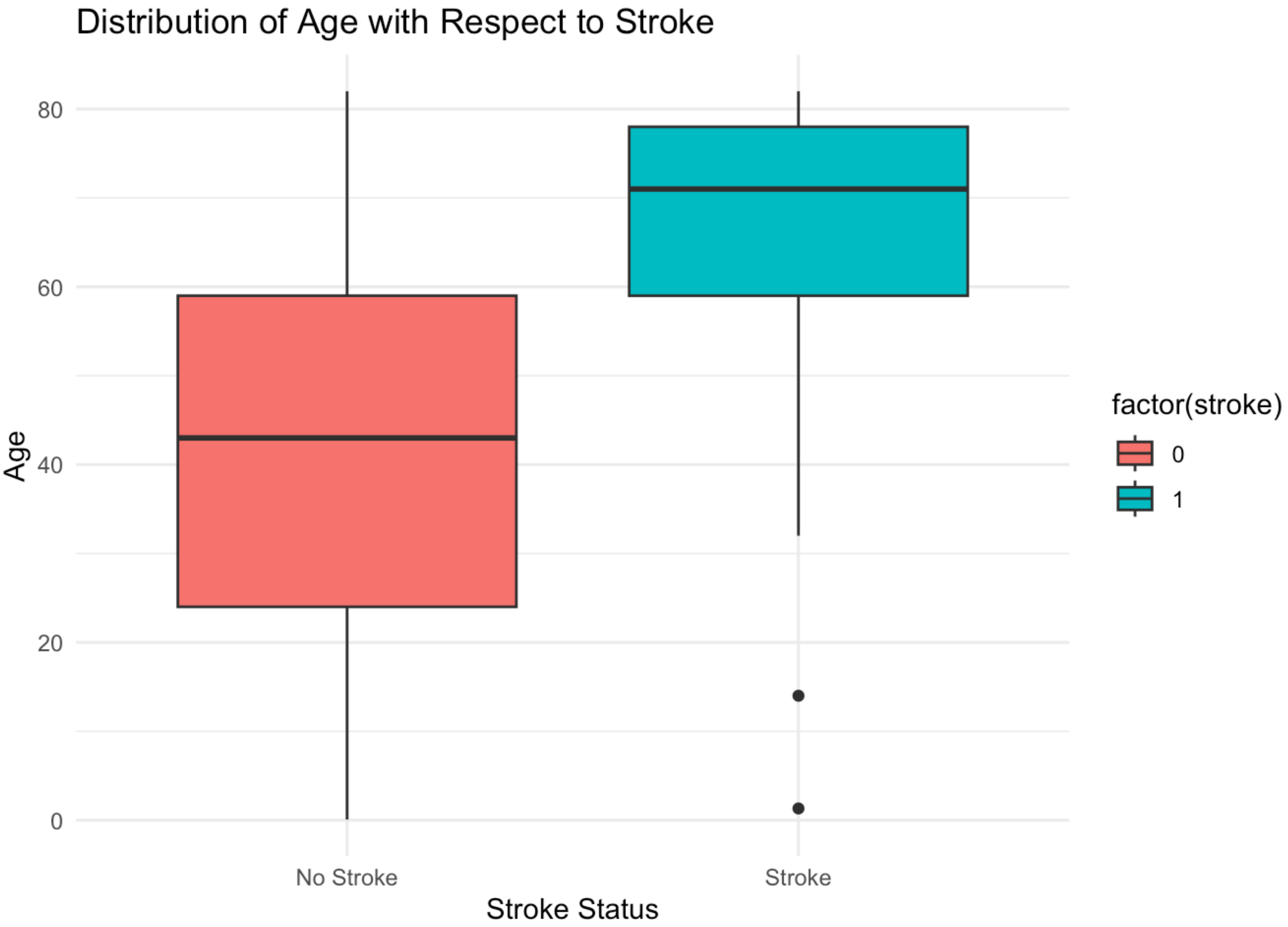
```
ggplot(stroke_data_raw, aes(x = factor(stroke), fill = factor(stroke))) +
  geom_bar() +
  labs(title = "Distribution of People with Stroke and without Stroke",
       x = "Stroke Status",
       y = "Count") +
  scale_x_discrete(labels = c("No Stroke", "Stroke")) +
  scale_fill_discrete(name = "Stroke", labels = c("No Stroke", "Stroke")) +
  theme_minimal()
```



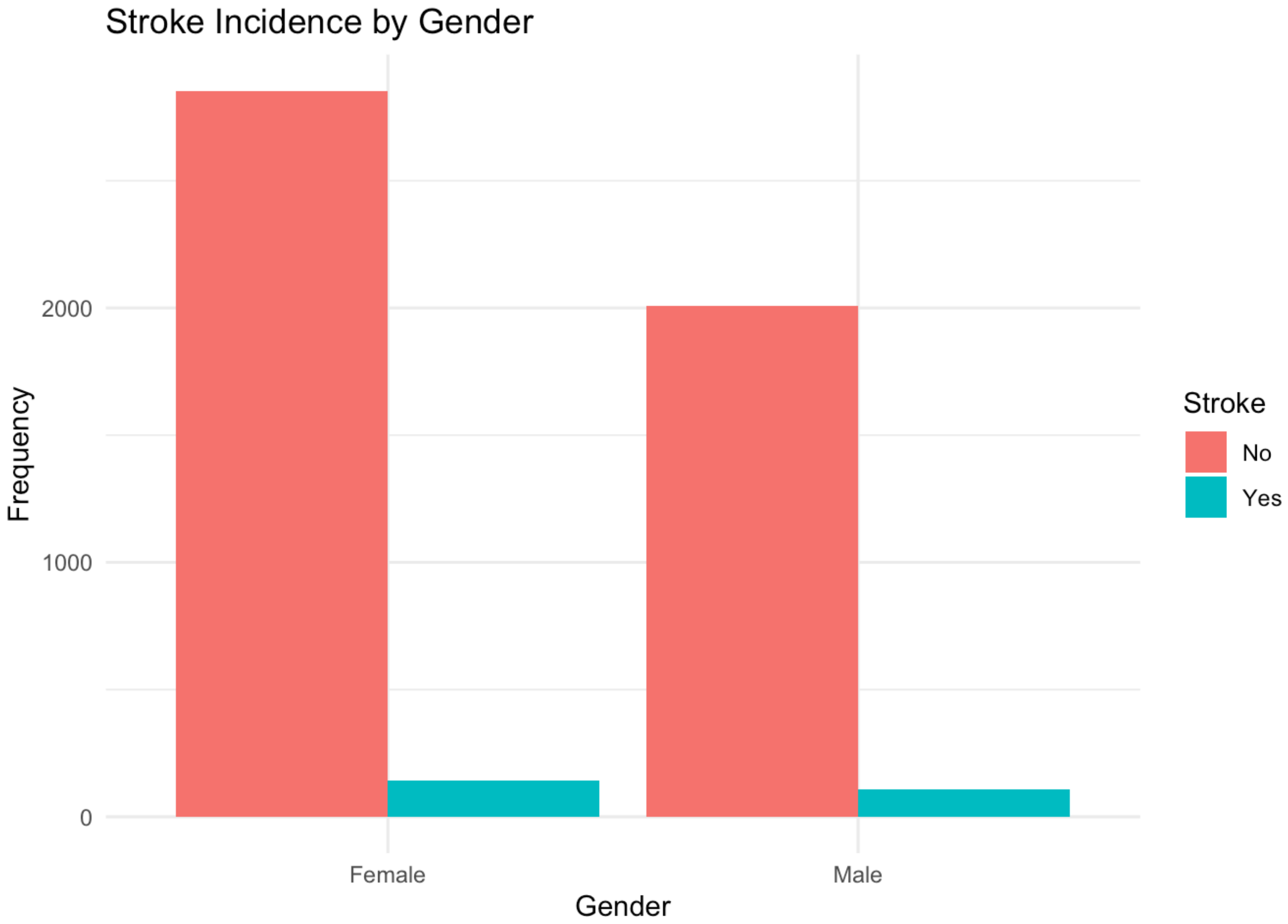
The distribution of patients with and without stroke: Out of a total of 5,109 patients, only 249 have experienced a stroke, making up approximately 4.9% of the dataset. In contrast, 4,860 patients, representing about 95.1% of the data, have never had a stroke.

The distribution of patients with and without the stroke with respect to hypertension, heart_disease, avg_glucose_level and smoking_status conditions

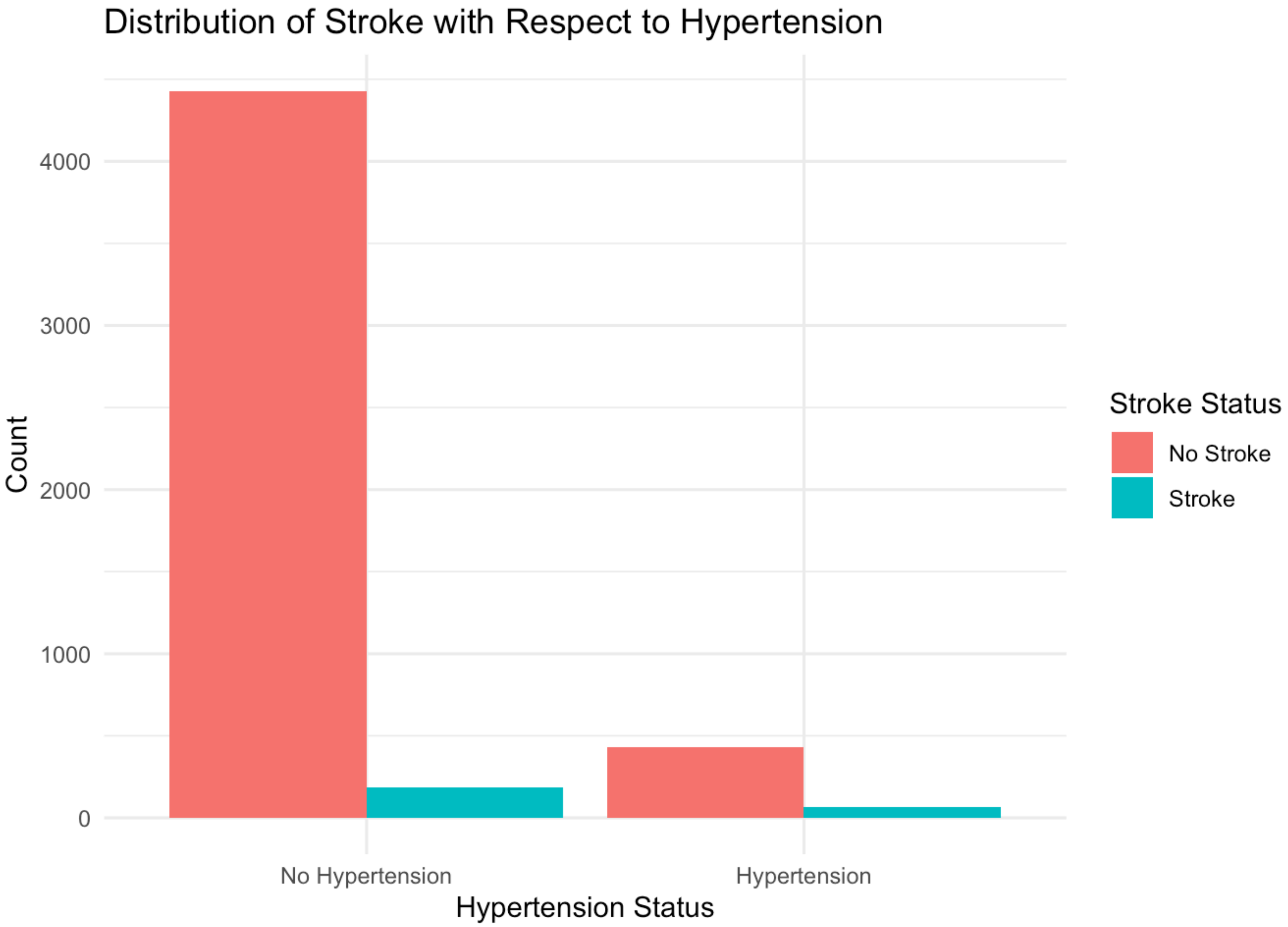
```
# Age distribution
ggplot(stroke_data_raw, aes(x = factor(stroke), y = age, fill = factor(stroke))) +
  geom_boxplot() +
  labs(title = "Distribution of Age with Respect to Stroke",
       x = "Stroke Status",
       y = "Age") +
  scale_x_discrete(name = "Stroke Status", labels = c("No Stroke", "Stroke")) +
  theme_minimal()
```



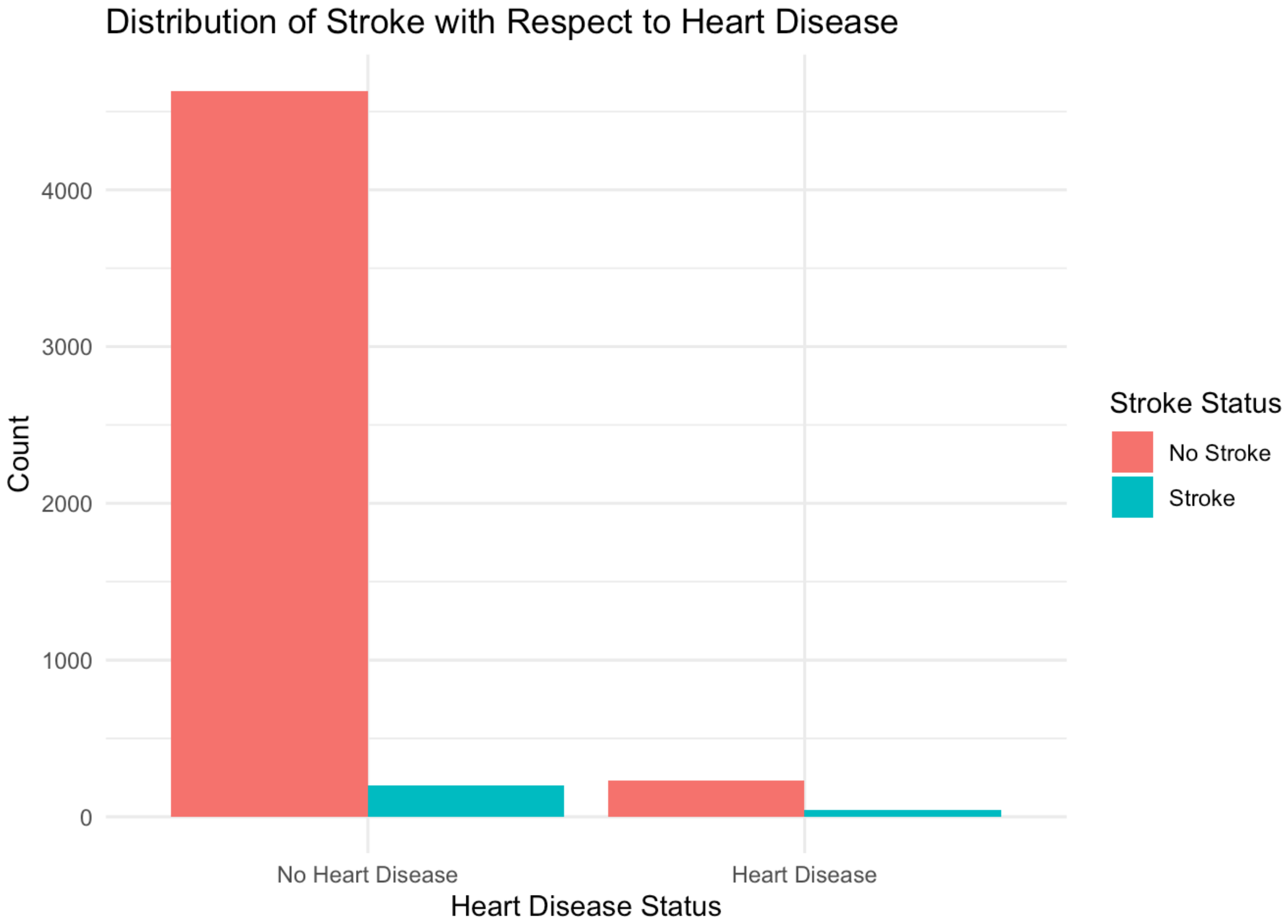
```
# Stroke by gender
ggplot(stroke_data_raw, aes(x = gender, fill = as.factor(stroke))) +
  geom_bar(position = "dodge") +
  theme_minimal() +
  xlab("Gender") +
  ylab("Frequency") +
  ggtitle("Stroke Incidence by Gender") +
  scale_fill_discrete(name = "Stroke", labels = c("No", "Yes"))
```



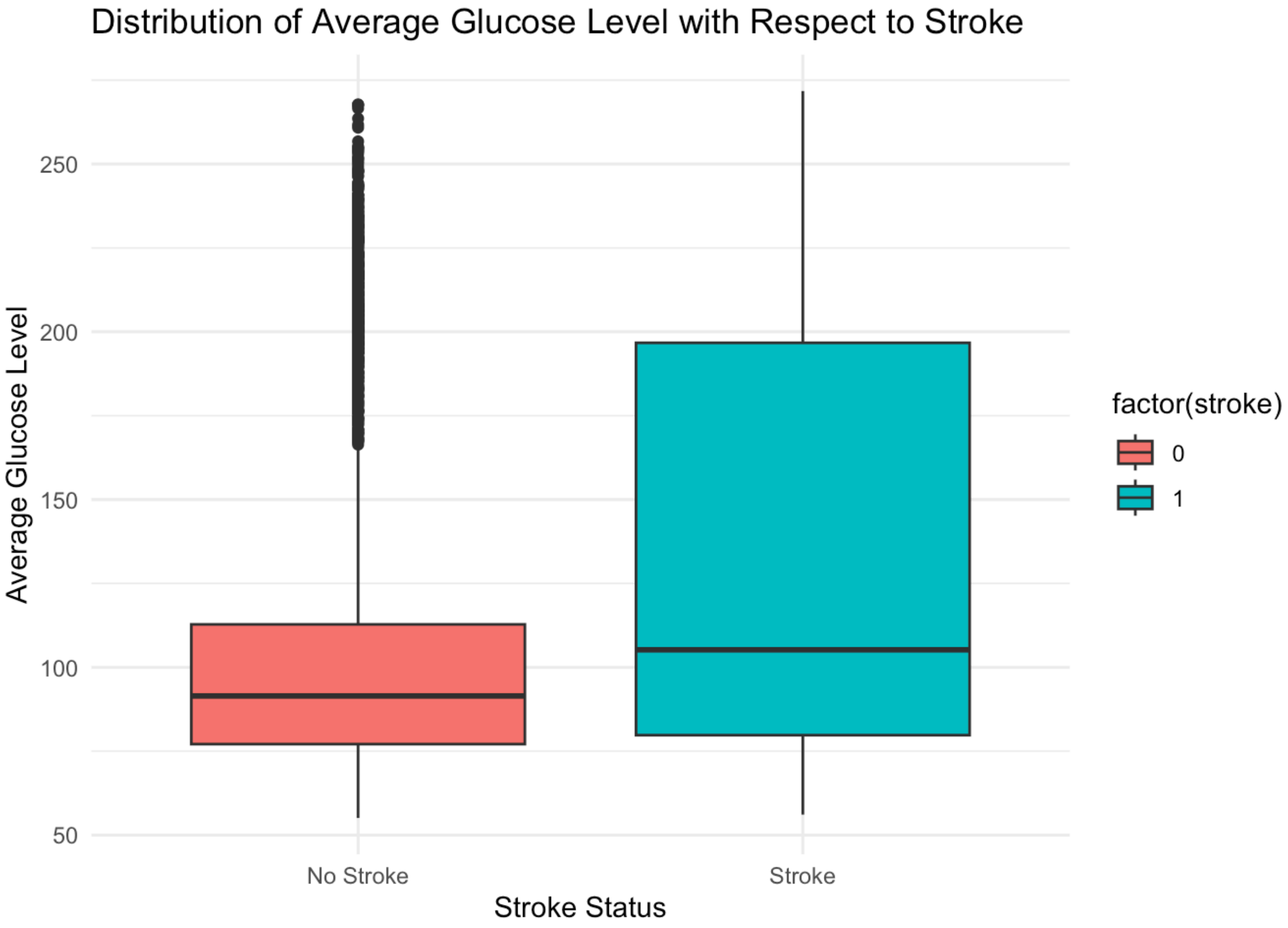
```
# Hypertension
ggplot(stroke_data_raw, aes(x = factor(hypertension), fill = factor(stroke))) +
  geom_bar(position = "dodge") +
  labs(title = "Distribution of Stroke with Respect to Hypertension",
       x = "Hypertension Status",
       y = "Count") +
  scale_x_discrete(labels = c("No Hypertension", "Hypertension")) +
  scale_fill_discrete(name = "Stroke Status", labels = c("No Stroke", "Stroke")) +
  theme_minimal()
```



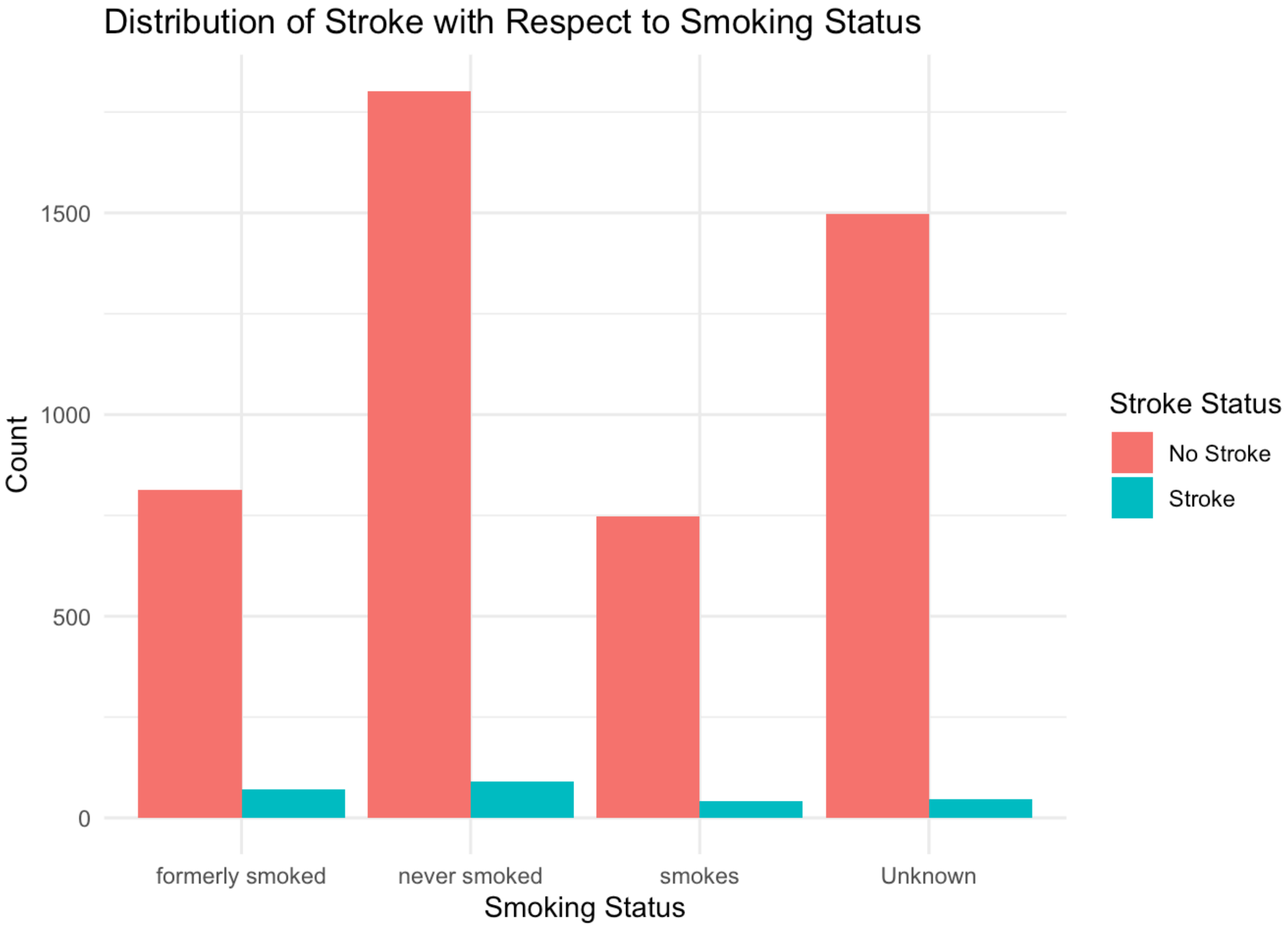
```
# Heart Disease
ggplot(stroke_data_raw, aes(x = factor(heart_disease), fill = factor(stroke))) +
  geom_bar(position = "dodge") +
  labs(title = "Distribution of Stroke with Respect to Heart Disease",
       x = "Heart Disease Status",
       y = "Count") +
  scale_x_discrete(labels = c("No Heart Disease", "Heart Disease")) +
  scale_fill_discrete(name = "Stroke Status", labels = c("No Stroke", "Stroke")) +
  theme_minimal()
```



```
# Average Glucose Level
ggplot(stroke_data_raw, aes(x = factor(stroke), y = avg_glucose_level, fill = factor(
stroke))) +
  geom_boxplot() +
  labs(title = "Distribution of Average Glucose Level with Respect to Stroke",
       x = "Stroke Status",
       y = "Average Glucose Level") +
  scale_x_discrete(name = "Stroke Status", labels = c("No Stroke", "Stroke")) +
  theme_minimal()
```



```
# Smoking Status
ggplot(stroke_data_raw, aes(x = smoking_status, fill = factor(stroke))) +
  geom_bar(position = "dodge") +
  labs(title = "Distribution of Stroke with Respect to Smoking Status",
       x = "Smoking Status",
       y = "Count") +
  scale_fill_discrete(name = "Stroke Status", labels = c("No Stroke", "Stroke")) +
  theme_minimal()
```



Upon visualizing the variables of the stroke dataset, it’s clear that only a small percentage of patients (4.9%) have experienced a stroke, while the majority (95.1%) have not. These proportions are displayed in a bar chart indicating the distribution of patients with and without a stroke.

Age distribution data was analyzed using boxplots. The results indicate that strokes can occur across a wide range of age groups, but there is a higher incidence among individuals aged between 60 and 80. This suggests that preventive measures and stroke prediction strategies should be particularly focused on these age groups.

The data also presented variations in stroke incidence between genders, suggesting potential differences in risk factor exposure or susceptibility between males and females. This implies that gender could play a critical role in developing stroke prevention strategies.

Upon comparing stroke incidences with the presence of hypertension, it was found that individuals with hypertension had a higher incidence of strokes. This finding confirms the known link between hypertension and stroke risk, underscoring the importance of hypertension management in stroke prevention.

There were no discernable trends concerning stroke incidence among individuals with heart disease in this dataset. Although heart disease is known to be a risk factor for stroke, additional analysis may be required to understand the specific relationship in this dataset.

In terms of average glucose levels, there seems to be a different pattern among patients who have experienced a stroke compared to those who haven't. This could highlight the importance of regular glucose monitoring and control in preventing strokes.

Additionally, the data showed a higher stroke incidence among former smokers and those who never smoked, compared to current smokers or those with unknown smoking status. This highlights the importance of tobacco control strategies in mitigating the risk of stroke.

In summary, the exploratory data analysis uncovers several factors that might influence stroke risk. Further comprehensive analysis is recommended to understand the specific role and significance of these factors in contributing to stroke risk.

Modifying the dataset to balance the proportion of strokes.

In the preliminary examination of the dataset, we observe a considerable imbalance in the distribution of stroke and no-stroke cases. Specifically, the dataset contains 4860 records of no-stroke instances against a substantially lower count of 249 stroke instances. This imbalance can significantly skew the results of any predictive model built on this dataset.

Given that the prevalence of no-stroke cases dominates the dataset, any classification model built on this data would invariably yield a low error rate. However, this does not necessarily signify a well-fitted model. The preponderance of no-stroke cases could mask an inadequate prediction of positive stroke cases, potentially leading to an undervaluation of the model's error rate.

To circumvent this issue, we propose to balance the dataset proportionally between stroke and no-stroke instances. To achieve this, we plan to retain a sample of 350 no-stroke cases while incorporating all the 249 stroke cases. This approach, while not typically recommended for real-world data applications, can offer a more balanced and representative dataset for our modelling purposes.

The method we are implementing here is referred to as stratified sampling. It involves dividing the population into homogeneous subgroups or strata and sampling individual groups separately. This process ensures that our sample will adequately represent both stroke and no-stroke instances, thus enhancing the reliability of our subsequent analysis. Please note that this new balanced dataset will provide an estimate of stroke risk based on the inputs for predictors and is not an absolute reflection of the original population.

```r
set.seed(2500)
idx=sampling:::strata(data=stroke_data_raw,stratanames = c("stroke"),size=c(249,350),
method="srswor")

#Creating the test and train data based on the criteria
stroke_data=stroke_data_raw[idx$ID_unit,]
```

```r
#Here we want to make sure that the stroke column for the new dataset works fine.
table(stroke_data$stroke)
```

```
##
##   0   1
## 350 249
```

Spliting data into training and test set to build a model.

We have 599 records (350 without stroke and 249 with stroke). We are preparing our dataset by creating training and testing sets. Using stratified sampling, we divide the dataset, with approximately 80% used for training and the rest for testing. This method ensures similar ratios of stroke to no-stroke cases in both sets, maintaining the original data distribution. The specific samples are selected based on a seed value to allow reproducibility. The ‘stroke_train’ subset will be used to build our model, and ‘stroke_test’ will evaluate its performance.

```
#Removing id columns from the dataset
stroke_data <- stroke_data[, -which(names(stroke_data) %in% c("id"))]
#The "Never_worked" class in work_type column has one record that exist only in the t
rain set. So, to avoid errors while predicting, we keep it away from the data.
stroke_data <- stroke_data[ !(stroke_data$work_type =="Never_worked"), ]
```

```
dim(stroke_data)
```

```
## [1] 598  11
```

```
table(stroke_data$stroke)
```

```
##
##   0   1
## 349 249
```

```
#Drawing 3880 "0" and 198 "1" from the "stroke" column, the train set will be includi
ng around 80% of the total data. The ratio of Stroke/no-Stroke in the entire dataset
and both splits are almost the same.
set.seed(2500)
stroke_idx=sampling:::strata(data=stroke_data,stratanames = c("stroke"),size=c(187,26
2),method="srswor")

#Creating the test and train data based on the criteria
stroke_test=stroke_data[-stroke_idx$ID_unit,]
stroke_train=stroke_data[stroke_idx$ID_unit,]
```

Making sure we have the required number of observations in train and the test data:

```
table(stroke_train$stroke)
```

```
##
##   0   1
## 262 187
```

```
table(stroke_test$stroke)
```

```
##
##  0  1
## 87 62
```

Checking multicollinearity:

Before using a generalized linear model or LDA, it is better to search for multicollinearity between the predictors. Here we have a simple test from the car library. Multicollinearity will reduce the stability of the glm model. So, before creating any model, we check for the existence of this condition.

Conducting VIF test to check for multicollinearity

Null Hypothesis (H0): There is no multicollinearity among the predictors in the dataset.

Alternative Hypothesis (H1): There is multicollinearity among the predictors in the dataset.

```
Stroke_LRM_model_1<-glm(stroke~., family=binomial, data=stroke_data)
VIF_values<- vif(Stroke_LRM_model_1)
VIF_values
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## gender            1.063107  1        1.031071
## age               1.625273  1        1.274862
## hypertension      1.068229  1        1.033551
## heart_disease     1.148022  1        1.071458
## ever_married      1.238991  1        1.113100
## work_type         1.727981  3        1.095443
## Residence_type    1.033817  1        1.016768
## avg_glucose_level 1.159358  1        1.076735
## bmi               1.134758  1        1.065250
## smoking_status    1.260501  3        1.039339
```

Using the whole dataset, we observed no sign of multicollinearity, since the result of the VIF test does not show any number greater than 5, or better to say, all of the VIF factors are less than 2. We can assume that the correlations between the predictor do not affect the stability of the model. The last column of the VIF test

(GVIF^(1/(2*Df))) accounts for the different categories in attributes, and it is kind of a standardized value for this test.

Logestic Regression Model

We aim to predict whether a patient will have a stroke or not based on the input parameters like gender, age, hypertension, heart disease, ever married, work type, residence type, average glucose level, body mass index (BMI), and smoking status.

Null Hypothesis (H0): None of predictor variables (gender, age, hypertension, heart disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status) have an effect on the outcome variable (stroke).

Alternative Hypothesis (H1): At least one of the predictor variables (gender, age, hypertension, heart disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status) is significantly associated with the outcome variable (stroke).

```
#Converting categorical variables to factors for the trian set
stroke_train$gender <- as.factor(stroke_train$gender)
stroke_train$ever_married <- as.factor(stroke_train$ever_married)
stroke_train$work_type <- as.factor(stroke_train$work_type)
stroke_train$Residence_type <- as.factor(stroke_train$Residence_type)
stroke_train$smoking_status <- as.factor(stroke_train$smoking_status)
stroke_train$stroke <- as.factor(stroke_train$stroke)
```

```
#Making sure that r dummifies them correctly
contrasts(stroke_train$smoking_status)
```

```
##                 never smoked smokes Unknown
## formerly smoked            0      0       0
## never smoked               1      0       0
## smokes                     0      1       0
## Unknown                    0      0       1
```

Building a model using the training set.

```
stroke_glm_model <- glm(stroke ~ ., family = binomial, data = stroke_train)
summary(stroke_glm_model)
```

```
##
## Call:
## glm(formula = stroke ~ ., family = binomial, data = stroke_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1940  -0.7102  -0.2548   0.7717   2.6086
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -3.576607   0.969442  -3.689 0.000225 ***
## genderMale                  0.288177   0.254238   1.133 0.257007
## age                         0.079864   0.009985   7.999 1.26e-15 ***
## hypertension                0.621526   0.322346   1.928 0.053839 .
## heart_disease               0.159904   0.380424   0.420 0.674244
## ever_marriedYes             0.522516   0.415492   1.258 0.208543
## work_typeGovt_job          -2.090229   1.039899  -2.010 0.044428 *
## work_typePrivate           -2.264150   1.015560  -2.229 0.025783 *
## work_typeSelf-employed     -2.382336   1.066848  -2.233 0.025545 *
## Residence_typeUrban         0.123470   0.245559   0.503 0.615097
## avg_glucose_level           0.002033   0.002317   0.877 0.380234
## bmi                         0.003264   0.019202   0.170 0.865010
## smoking_statusnever smoked -0.413418   0.316088  -1.308 0.190900
## smoking_statussmokes       -0.233934   0.382896  -0.611 0.541227
## smoking_statusUnknown      -0.258304   0.388810  -0.664 0.506470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.86  on 448  degrees of freedom
## Residual deviance: 425.70  on 434  degrees of freedom
## AIC: 455.7
##
## Number of Fisher Scoring iterations: 5
```

Summary of the logistic regression model:

The logistic regression model was built using the ‘stroke_train’ dataset to predict the likelihood of having a stroke. The model summary provides evidence against the null hypothesis and supports the alternative hypothesis, indicating a significant association between certain predictor variables and the likelihood of having a stroke.

The intercept term represents the log-odds of having a stroke when all other predictors are zero. The coefficients for the remaining variables indicate the change in log-odds of having a stroke associated with a one-unit increase in each predictor variable.

Based on the above summary output, the following conclusions can be drawn:

- The variable ‘age’ has a significant positive effect on the probability of having a stroke. As the age increases, the log-odds of having a stroke also increase.
- The variable ‘work_type’ is also significant, but it is negatively associated with the probability of having a stroke. Specifically, having a ‘Govt_job’, being ‘Private’ or being ‘Self-employed’ reduces the log-odds of having a stroke compared to the reference category.
- ‘smoking_statusnever_smoked’ is significant and negatively related to the probability of having a stroke. This means that those who have never smoked have lower log-odds of having a stroke compared to the reference category.

However, variables such as ‘gender’, ‘hypertension’, ‘heart_disease’, ‘ever_married’, ‘Residence_type’, ‘avg_glucose_level’, ‘bmi’, and other ‘smoking_status’ categories do not appear to have a significant effect on the probability of having a stroke based on this model.

Using GLM model to predict stroke status in the test set

```
#Converting categorical variables to factors for the test set
stroke_test$gender <- as.factor(stroke_test$gender)
stroke_test$ever_married <- as.factor(stroke_test$ever_married)
stroke_test$work_type <- as.factor(stroke_test$work_type)
stroke_test$Residence_type <- as.factor(stroke_test$Residence_type)
stroke_test$smoking_status <- as.factor(stroke_test$smoking_status)
stroke_test$stroke <- as.factor(stroke_test$stroke)
```

```
stroke_pred_prob <- predict(stroke_glm_model, newdata = stroke_test, type = "response
")
```

Since the outcome of a logistic regression is a probability, we need to convert these probabilities to a binary outcome. Using 0.5 as the cut-off for the same.

```
stroke_pred <- ifelse(stroke_pred_prob > 0.5, 1, 0)
```

Creating a confusion matrix and calculating the misclassification rate.

```
conf_matrix <- table(Predicted = stroke_pred, Actual = stroke_test$stroke)
print(conf_matrix)
```

```
##          Actual
## Predicted  0  1
##         0 70 15
##         1 17 47
```

```
misclassification_rate_glm_full <- 1 - sum(diag(conf_matrix)) / sum(conf_matrix)
print(paste("Misclassification Rate: ", round(misclassification_rate_glm_full * 100,
2), "%"))
```

```
## [1] "Misclassification Rate:  21.48 %"
```

Upon applying the model to the ‘stroke_test’ dataset, we predict stroke occurrences and compare these predictions to the actual values to assess the model’s performance.

The confusion matrix gives a count of correct and incorrect predictions. In our case, it correctly identified 66 cases without a stroke, and misclassified 16 actual stroke cases as non-stroke. The model correctly predicted 46 cases with a stroke and misclassified 20 non-stroke cases as having a stroke.

The misclassification rate, calculated from the confusion matrix, is 21.48%. This means that approximately 21.48% of the total predictions were incorrect.

While this rate seems moderate, it’s important to consider that in medical scenarios like this, false negatives (incorrectly predicting no stroke when there actually is one) and false positives (incorrectly predicting a stroke when there isn’t one) can have serious health implications. Therefore, improving the model’s sensitivity and specificity to accurately predict stroke occurrences is critical.

Reduced Logistic Regression Model

When a logistic regression model includes many predictors, some of which are not statistically significant, it can lead to overfitting. This situation often results in the model performing well on the training data but poorly on new, unseen data. Consequently, reducing the model by removing non-significant predictors can help to mitigate overfitting and improve the model’s generalizability.

From the earlier full model, the variables ‘gender’, ‘heart_disease’, ‘ever_married’, ‘Residence_type’, ‘avg_glucose_level’, ‘bmi’, and’smoking_status’ have p-values greater than 0.05, indicating that they are not statistically significant predictors. Thus, these variables can be excluded to form a reduced model.

```
# Reduced logistic regression model
stroke_glm_model_reduced <- glm(stroke ~ age + hypertension + work_type, family = bin
omial, data = stroke_train)
summary(stroke_glm_model_reduced)
```

```
##
## Call:
## glm(formula = stroke ~ age + hypertension + work_type, family = binomial,
##     data = stroke_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1558  -0.7290  -0.2802   0.7737   2.5851
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.416979   0.740646  -4.614 3.96e-06 ***
## age                    0.084525   0.009238   9.149  < 2e-16 ***
## hypertension           0.620967   0.314947   1.972   0.0486 *
## work_typeGovt_job     -1.794393   0.908708  -1.975   0.0483 *
## work_typePrivate      -1.914343   0.881312  -2.172   0.0298 *
## work_typeSelf-employed -2.123046   0.936041  -2.268   0.0233 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.86  on 448  degrees of freedom
## Residual deviance: 433.37  on 443  degrees of freedom
## AIC: 445.37
##
## Number of Fisher Scoring iterations: 5
```

Using Reduced GLM model to predict stroke status in the test set

Since the outcome of a logistic regression is a probability, we need to convert these probabilities to a binary outcome. Using 0.5 as the cut-off for the same.

```
stroke_pred_prob_reduced <- predict(stroke_glm_model_reduced, newdata = stroke_test,
type = "response")
stroke_pred_reduced <- ifelse(stroke_pred_prob_reduced > 0.5, 1, 0)
```

Creating a confusion matrix and calculating the misclassification rate for Reduced GLM model.

```
conf_matrix <- table(Predicted = stroke_pred_reduced, Actual = stroke_test$stroke)
print(conf_matrix)
```

```
##          Actual
## Predicted  0  1
##         0 70 19
##         1 17 43
```

```
misclassification_rate_glm_full <- 1 - sum(diag(conf_matrix)) / sum(conf_matrix)
print(paste("Misclassification Rate: ", round(misclassification_rate_glm_full * 100,
2), "%"))
```

```
## [1] "Misclassification Rate:  24.16 %"
```

The reduced logistic regression model, which includes age, work type, and hypertension as significant predictors, achieves a misclassification rate of 24.16% on the test set. This indicates that approximately 24.16% of the predictions made by the model are incorrect.

The misclassification rate of the reduced model (24.16%) is higher than the full model (21.48%). Given the nature of the data (stroke prediction), optimizing for the highest possible accuracy is important since false negatives (not predicting a stroke when one will occur) could have serious health consequences.

Therefore, based on the misclassification rates and sensitivity nature of the data, the full logistic regression model can be selected as the preferred model for predicting stroke occurrences thus far.

Before finishing this part, we should not forget to check the normality of the residuals from the logestic regression full model.

H0: The residuals are normally distributed

H1: The residuals are not normally distributed

```
shapiro.test(stroke_glm_model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  stroke_glm_model$residuals
## W = 0.7677, p-value < 2.2e-16
```

The p-value from the above Shapiro test (For residuals) is small enough to reject the null hypothesis. We have to be very careful about using logistic regression for the test data and interpreting the outputs since the normality assumption is not met.

In the upcoming sections, we'll explore other generalized linear regression models as well as the classification algorithms such as Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Decision Tree algorithms. By comparing their respective accuracy and misclassification rates with the logistic regression model, we aim to identify the most suitable modeling approach for this problem. This iterative process of model selection and evaluation is crucial in ensuring we develop a model that provides the most reliable predictions possible for the incidence of stroke.

Binary regression model (identity link function)

In this section, a binary linear regression model using an identity link function is implemented to calculate the misclassification rate based on the cross-validation error. It is important to note that the identity and log link methods may encounter convergence issues for binary responses due to their inherent limitations. Nevertheless, two models have been identified as stable for these link functions after thorough testing.

To facilitate this, an ordering technique is employed to assign orders to ‘age’, ‘bmi’, and ‘avg_glucose’ attributes based on different ranges or intervals. Five ascending orders are assigned for each of these predictors.

The intervals and their corresponding orders for ‘age’ are:

Age:

[0,20) –> 2

[20,30)–> 4

[30,40) –> 6

[40,50) –> 8

[+50] –> 10

bmi:

[10,20) –> 11

[20,30)–> 12

[30,40) –> 13

[40,50) –> 14

[50-60] –> 15

avg_glucose_level:

[55,100) –> 1

[100,140)–> 3

[140,180) –> 5

[180,220) –> 7

[220,275] –> 9

```
# Create a new dataset stroke_data_ordered
stroke_data_ordered <- stroke_data

# Define age_order based on age intervals
stroke_data_ordered$age_order <- ifelse(stroke_data_ordered$age < 20, 2,
                                        ifelse(stroke_data_ordered$age >= 20 & stroke
_data_ordered$age < 30, 4,
                                               ifelse(stroke_data_ordered$age >= 30 &
stroke_data_ordered$age < 40, 6,
                                                      ifelse(stroke_data_ordered$age
>= 40 & stroke_data_ordered$age < 50, 8,
                                                             ifelse(stroke_data_order
ed$age >= 50, 10, NA)))))


# Define bmi_order based on BMI intervals
stroke_data_ordered$bmi_order <- ifelse(stroke_data_ordered$bmi >= 10 & stroke_data_o
rdered$bmi <= 20, 11,
                                        ifelse(stroke_data_ordered$bmi > 20 & stroke_
data_ordered$bmi <= 30, 12,
                                               ifelse(stroke_data_ordered$bmi > 30 &
stroke_data_ordered$bmi <= 40, 13,
                                                      ifelse(stroke_data_ordered$bmi
> 40 & stroke_data_ordered$bmi <= 50, 14,
                                                             ifelse(stroke_data_order
ed$bmi > 50 & stroke_data_ordered$bmi <= 60, 15, NA)))))

# Define avg_glucose_level_order based on average glucose level intervals
stroke_data_ordered$avg_glucose_level_order <- ifelse(stroke_data_ordered$avg_glucose
_level >= 55 & stroke_data_ordered$avg_glucose_level <= 100, 1,
                                                      ifelse(stroke_data_ordered$avg_
glucose_level > 100 & stroke_data_ordered$avg_glucose_level <= 140, 3,
                                                             ifelse(stroke_data_order
ed$avg_glucose_level > 140 & stroke_data_ordered$avg_glucose_level <= 180, 5,
                                                                    ifelse(stroke_dat
a_ordered$avg_glucose_level > 180 & stroke_data_ordered$avg_glucose_level <= 220, 7,
                                                                           ifelse(str
oke_data_ordered$avg_glucose_level > 220 & stroke_data_ordered$avg_glucose_level <= 3
00, 9, NA)))))

# View the updated dataset
head(stroke_data_ordered)
```

gen...	...	hypertension	heart_disease	ever_married	work_type	Residence_type	▸
<chr>	<dbl>	<int>	<int>	<chr>	<chr>	<chr>	

1	Male	67	0	1	Yes	Private	Urban
2	Female	61	0	0	Yes	Self-employed	Rural
3	Male	80	0	1	Yes	Private	Rural
4	Female	49	0	0	Yes	Private	Urban
5	Female	79	1	0	Yes	Self-employed	Rural
6	Male	81	0	0	Yes	Private	Urban

6 rows | 1-8 of 15 columns

Initially, when age was included in the model with the identity link function, we observed instability in the model’s predictions. To address this issue, we decided to exclude the age variable and include as many other relevant predictors as possible.

The resulting logistic regression model, utilizing the identity link function, was built using the following predictors: hypertension, average glucose level (represented by the variable avg_glucose_level_order), and marital status (represented by the variable ever_married).

```
binary_identity_model <- glm(stroke ~ hypertension+avg_glucose_level_order+ever_marri
ed, data = stroke_data_ordered, family = binomial(link = 'identity'))


# Displaying the coefficients
coef(binary_identity_model)
```

```
##             (Intercept)            hypertension avg_glucose_level_order 
##              0.10374931              0.21719056              0.03425827 
##         ever_marriedYes 
##              0.22952147 
```

The model equation, expressing the probability of stroke occurrence (P(Y=1|x)), is given by:

P(Y=1|x)= 0.10 + 0.217hypertension + 0.034 avg_glucose_level_order + 0.22ever_married

To evaluate the performance of the model, we conducted k-fold cross-validation with k=10. The misclassification error was used as the evaluation metric. The misclassification error measures the average proportion of misclassified instances.

```
# K-fold cross validation for the three models
set.seed(2500)
cost<-function(r,pi) mean(abs(r-pi)>0.5)
cv_identity_link<-cv.glm(stroke_data_ordered, binary_identity_model, cost=cost, K=10)
cat("The misclassification error for the model using the identity link function is\n"
,cv_identity_link$delta[1]*100,"%")
```

```
## The misclassification error for the model using the identity link function is
##  32.44147 %
```

After performing k-fold cross-validation, the misclassification error for the model using the identity link function was found to be 32.44%, indicating the overall accuracy and effectiveness of the model.

Binary regression model (log link function)

In this analysis, we utilized a binary regression model with a log link function to predict the occurrence of stroke. The aim was to examine the stability and significance of the variable age_order when included in the model. To ensure the model's comprehensiveness, we included as many relevant variables as possible.

The logistic regression model with the log link function was constructed using the following predictors: age_order and hypertension.

```
binary_log_model <- glm(stroke ~age_order+hypertension
, data = stroke_data_ordered, family = binomial(link = 'log'))


coef(binary_log_model)
```

```
##  (Intercept)    age_order hypertension 
##   -5.5095443    0.4973334    0.1863978
```

The model equation, expressed as the logarithm of the probability of stroke occurrence (Log(P(Y=1|X))), is given by:

Log(P(Y=1|X))= -5.5 + 0.50age_order + 0.186hypertension

To further evaluate the performance of the model, we conducted k-fold cross-validation with k=10. The misclassification error was used as the evaluation metric, which measures the average proportion of misclassified instances.

```
set.seed(2500)
cv_log_link<-cv.glm(stroke_data_ordered, binary_log_model, cost=cost, K=10)
cat("The misclassification error for the model using the log link function is\n",cv_l
og_link$delta[1]*100,"%")
```

```
## The misclassification error for the model using the log link function is
##  26.92308 %
```

After performing k-fold cross-validation, the misclassification error for the model using the log link function was found to be 29.92%, indicating the overall accuracy and effectiveness of the model.

Although we employed two different models, the log link function demonstrated better performance in terms of misclassification error for this dataset. However, it is important to note that the model utilizing the logit function, as discussed in the previous part, exhibited superior accuracy.

Comparison of Log and Logit Link Functions for Age Predictor

In this section, we aim to compare the performance of the log and logit link functions when considering age_order as the explanatory variable in predicting stroke occurrence.

Using the Log Function as the Link Function

We first fit a logistic regression model using the log function as the link function:

```
binary_log_model_2 <- glm(stroke ~ age_order, data = stroke_data_ordered, family = bi
nomial(link = 'log'))
```

The coefficients of the logistic regression model are:

```
coef(binary_log_model_2)
```

```
## (Intercept)   age_order
##  -5.5841456   0.5097537
```

```
table(stroke_data_ordered$age_order)
```

```
##
##   2   4   6   8  10
##  64  38  49  77 370
```

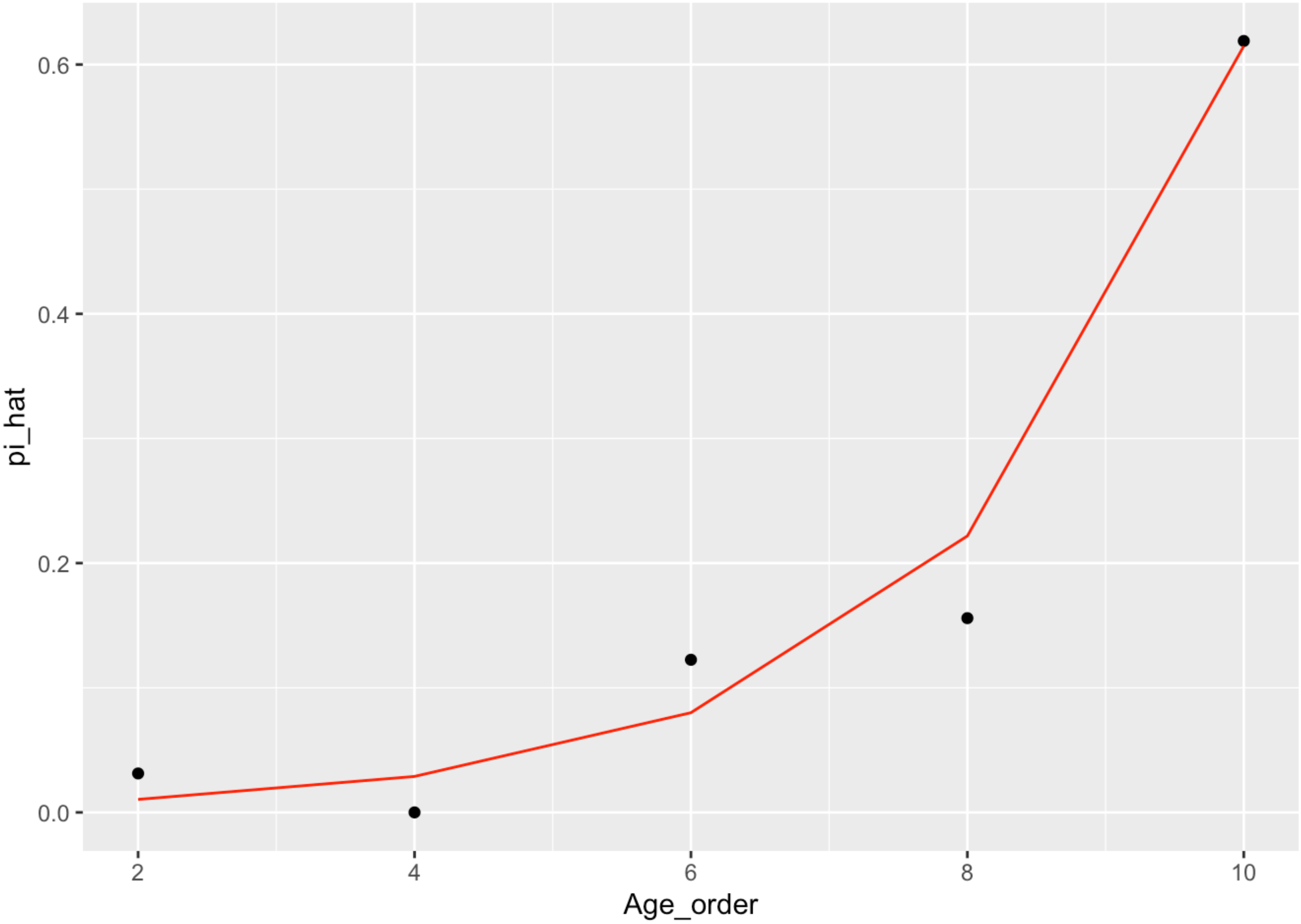
We then examine the distribution of the age_order variable:

```
table(age_order=stroke_data_ordered$age_order, Stroke=stroke_data_ordered$stroke)
```

```
##           Stroke
## age_order   0   1
##         2  62   2
##         4  38   0
##         6  43   6
##         8  65  12
##         10 141 229
```

To visualize the fitting result, we plot the observed proportions (pi_hat) against the fitted probabilities (pi_fit) based on the model:

```
#visualizting the fitting result
pi_hat<-c(2,0,6,12,229)/c(64,38,49,77,370)
Age_order=c(2,4,6,8,10)
pi_fit<- exp((-5.5841456) + 0.5097537*Age_order)
BIDATA<-data.frame(pi_hat, pi_fit)
ggplot()+geom_point(data = BIDATA, aes(x=Age_order, y=pi_hat))+geom_line(data = BIDAT
A, aes(x=Age_order, y=pi_fit), colour='red')
```



Lastly, we evaluate the model’s performance using k-fold cross-validation with k=10 and calculate the misclassification error:

```
set.seed(2500)
cost<-function(r,pi) mean(abs(r-pi)>0.5)
cv_fit_log<-cv.glm(stroke_data_ordered, binary_log_model_2, cost=cost, K=10)
cv_fit_log$delta[1]
```

```
## [1] 0.2692308
```

Using the Logit Function as the Link Function

Next, we fit a logistic regression model using the logit function as the link function:

```
binary_logit_model <- glm(stroke ~ age_order, data = stroke_data_ordered, family = bi
nomial)
```

The coefficients of the logistic regression model are:

```
coef(binary_logit_model)
```

```
## (Intercept)   age_order
##  -6.7350615   0.7156585
```

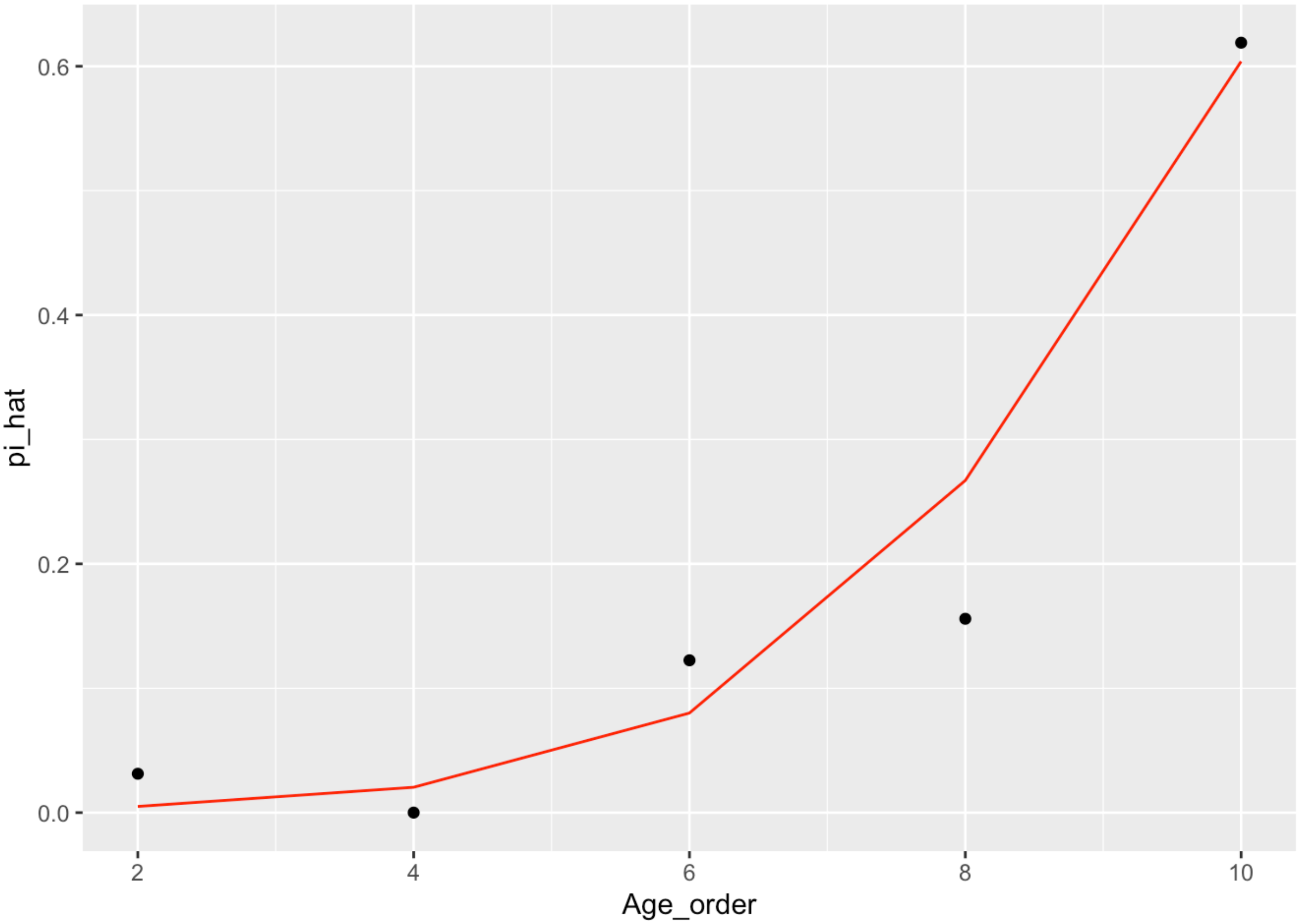
We examine the distribution of the age_order variable:

```
table(stroke_data_ordered$age_order)
```

```
##
##   2   4   6   8  10
##  64  38  49  77 370
```

Visualizing the fitting result, we plot the observed proportions (pi_hat) against the fitted probabilities (pi_fit) based on the model:

```
pi_hat<-c(2,0,6,12,229)/c(64,38,49,77,370)
predictor=Age_order
pi_fit<- exp(-6.7350615+0.7156585*Age_order)/(1+exp(-6.7350615+0.7156585*Age_order))
BIDATA<-data.frame(pi_hat, pi_fit)
ggplot()+geom_point(data = BIDATA, aes(x=Age_order, y=pi_hat))+geom_line(data = BIDAT
A, aes(x=Age_order, y=pi_fit), colour='red')
```



Evaluating the model’s performance using k-fold cross-validation with k=10 and calculate the misclassification error:

```
set.seed(2500)
cost<-function(r,pi) mean(abs(r-pi)>0.5)
cv_fit_logit<-cv.glm(stroke_data_ordered, binary_logit_model, cost=cost, K=10)
cv_fit_logit$delta[1]
```

```
## [1] 0.2692308
```

Based on this comparison, we can conclude that when considering only one predictor (age_order), there is almost no difference between using the log and logit functions as the link function. This is supported by the similar misclassification errors obtained from the cross-validation. Furthermore, the visualizations of the fitted probabilities for the two link functions exhibit close resemblance.

Linear discrimination Analysis

LDA assumptions test

Before conducting Linear Discriminant Analysis (LDA), it is important to ensure that the data satisfies certain assumptions. In LDA, there are two key assumptions: normality and variance homogeneity. Additionally, since we have multiple predictors, we need to check for multivariate normality, which is relevant for both LDA and Quadratic Discriminant Analysis (QDA) approaches.

The function fitted on the data in the posterior probability formula follows the normal distribution. Therefore it is important to check this condition.

Multivariate Normality:

To assess multivariate normality, we will use Mardia’s Test in R. However, it is important to note that the concept of multivariate normality applies only to continuous variables, and not to categorical variables. Therefore, in this step, we will focus on selecting the continuous predictors.

```
#stroke_test
#stroke_train
#stroke_data
# Subset dataset for all continous columns except id and stroke column
subset_for_normality_test <- stroke_data[, which(names(stroke_data) %in% c("age", "av
g_glucose_level","bmi"))]
```

To test the assumption of multivariate normality, we will utilize two different tests: Mardia’s Test and the energy package.

H0: The predictors follow a multivariate normal distribution. Ha: The predictors do not follow a multivariate normal distribution.

Mardia’s Test:

We perform the Mardia’s Test in R using the mult.norm() function:

```
#perform Multivariate normality test
mult.norm(subset_for_normality_test)$mult.test
```

```
##            Beta-hat       kappa     p-val
## Skewness  3.442352 343.0877444 0.0000000
## Kurtosis 15.071513   0.1596406 0.8731642
```

If any of the p-values for skewness is less than 0.05, we reject the null hypothesis, indicating a lack of multivariate normality based on the Mardia’s Test. Since one of the p value is less than 0.05 we reject our null hypothesis and conclude the lack of multivariate normality.

Energy Package Test:

We can also employ the mvnorm.etest() function from the energy package to further evaluate multivariate normality:

```
mvnorm.etest(subset_for_normality_test, R=50)
```

```
##
##  Energy test of multivariate normality: estimated parameters
##
## data:  x, sample size 598, dimension 3, replicates 50
## E-statistic = 15.445, p-value < 2.2e-16
```

By conducting 50 replicates, we obtain a p-value that allows us to make a conclusion regarding the existence of multivariate normality. Since p-value is less than 0.05, it provides strong evidence against the null hypothesis confirming our previous results (i.e, predictors do not follow a multivariate normal distribution).

It is important to note that normality tests for categorical variables in the dataset are not applicable since they are not continuous variables.

Based on the results of the multivariate normality tests, since we have evidence against the assumption of multivariate normality, we have to acknowledge that the assumption is not satisfied. However, for the purpose of comparing the accuracy of the LDA model with other methods, we are proceeding with creating the LDA model.

Linear Discriminant Analysis (LDA) Modeling and Evaluation

In this section, we proceed with the LDA model. However, it is important to note that the basic assumption of the LDA model, namely multivariate normality, is not satisfied due to the combination of categorical and continuous variables in the dataset.

To begin, we convert the categorical variables into proper numerical factors that can be used in the LDA formula:

```
#Converting categorical variables to factors for the trian set
stroke_train$gender <- as.factor(stroke_train$gender)
stroke_train$ever_married <- as.factor(stroke_train$ever_married)
stroke_train$work_type <- as.factor(stroke_train$work_type)
stroke_train$Residence_type <- as.factor(stroke_train$Residence_type)
stroke_train$smoking_status <- as.factor(stroke_train$smoking_status)
stroke_train$stroke <- as.factor(stroke_train$stroke)
```

```
#Making sure that r dummifies them correctly
contrasts(stroke_train$smoking_status)
```

```
##                  never smoked smokes Unknown
## formerly smoked            0      0       0
## never smoked               1      0       0
## smokes                     0      1       0
## Unknown                    0      0       1
```

Next, we create the LDA model using all the predictors:

```
#LDA model
LDA_model_1<-lda(stroke~., data = stroke_train)
LDA_model_1
```

```
## Call:
## lda(stroke ~ ., data = stroke_train)
## 
## Prior probabilities of groups:
##         0         1 
## 0.5835189 0.4164811 
## 
## Group means:
##   genderMale      age hypertension heart_disease ever_marriedYes
## 0  0.3625954 43.48153   0.08778626    0.05725191       0.6641221
## 1  0.4385027 67.74503   0.27807487    0.20320856       0.8983957
##   work_typeGovt_job work_typePrivate work_typeSelf-employed Residence_typeUrban
## 0         0.1412214        0.5916031              0.1564885           0.4885496
## 1         0.1336898        0.5828877              0.2727273           0.5561497
##   avg_glucose_level      bmi smoking_statusnever smoked smoking_statussmokes
## 0          105.9123 29.02758                 0.4045802            0.1793893
## 1          128.8099 30.06041                 0.3529412            0.1604278
##   smoking_statusUnknown
## 0             0.2709924
## 1             0.1711230
## 
## Coefficients of linear discriminants:
##                                      LD1
## genderMale                  0.182556899
## age                         0.056597225
## hypertension                0.517016261
## heart_disease               0.223838774
## ever_marriedYes            -0.037355812
## work_typeGovt_job          -1.176322451
## work_typePrivate           -1.213817431
## work_typeSelf-employed     -1.316655522
## Residence_typeUrban         0.018253375
## avg_glucose_level           0.002096535
## bmi                        -0.006062671
## smoking_statusnever smoked -0.375667596
## smoking_statussmokes       -0.293441784
## smoking_statusUnknown      -0.208776567
```

We convert the categorical variables to factors for the test set:

```
#Converting categorical variables to factors for the test set
stroke_test$gender <- as.factor(stroke_test$gender)
stroke_test$ever_married <- as.factor(stroke_test$ever_married)
stroke_test$work_type <- as.factor(stroke_test$work_type)
stroke_test$Residence_type <- as.factor(stroke_test$Residence_type)
stroke_test$smoking_status <- as.factor(stroke_test$smoking_status)
stroke_test$stroke <- as.factor(stroke_test$stroke)
```

Then, we make predictions using the LDA model and calculate the misclassification error:

```
#Making prediction
LDA_pred_1=predict(LDA_model_1, stroke_test)
#Calculating the misclassification error
test_set_actual=stroke_test$stroke

LDA_1_missclass_rate=mean(test_set_actual!=LDA_pred_1$class)*100
cat("misclassification rate based on the first LDA model LDA:\n",LDA_1_missclass_rate
,"%")
```

```
## misclassification rate based on the first LDA model LDA:
##  20.13423 %
```

```
table(Predicted=LDA_pred_1$class,Actual=stroke_test$stroke)
```

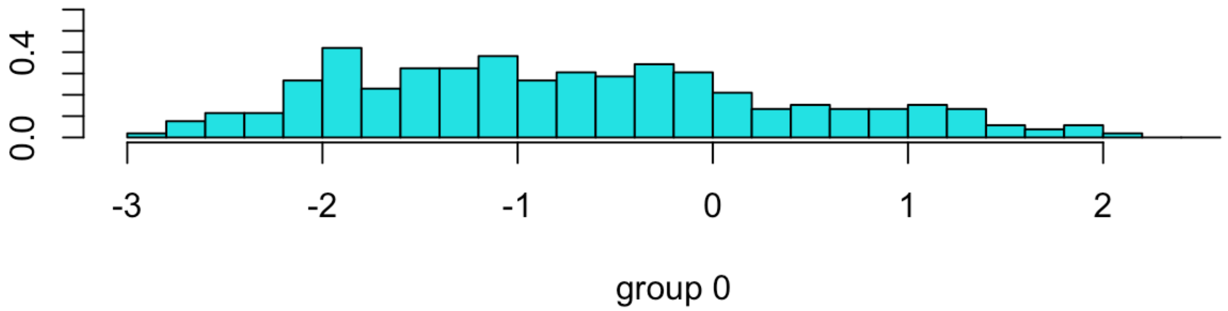
```
##          Actual
## Predicted  0  1
##         0 70 13
##         1 17 49
```

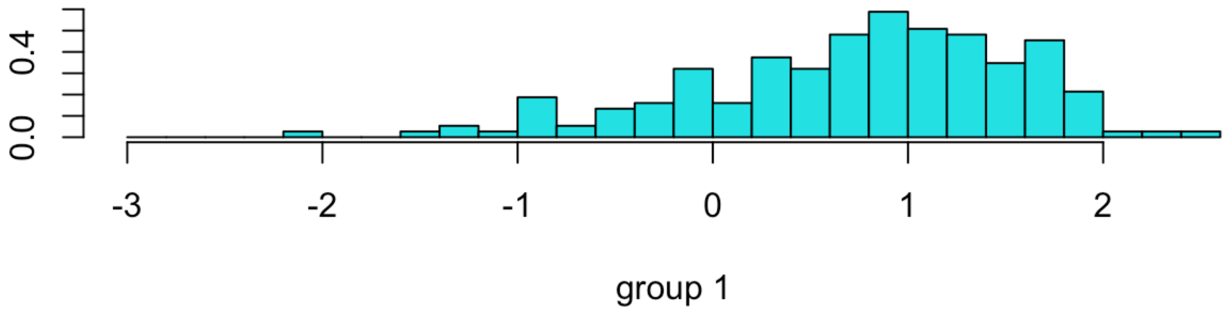
```
cat("misclassification rate based on the first LDA model LDA (table):\n",(13+17)/(49+
70+13+17)*100,"%")
```

```
## misclassification rate based on the first LDA model LDA (table):
##  20.13423 %
```

Visualizing the LDA model:

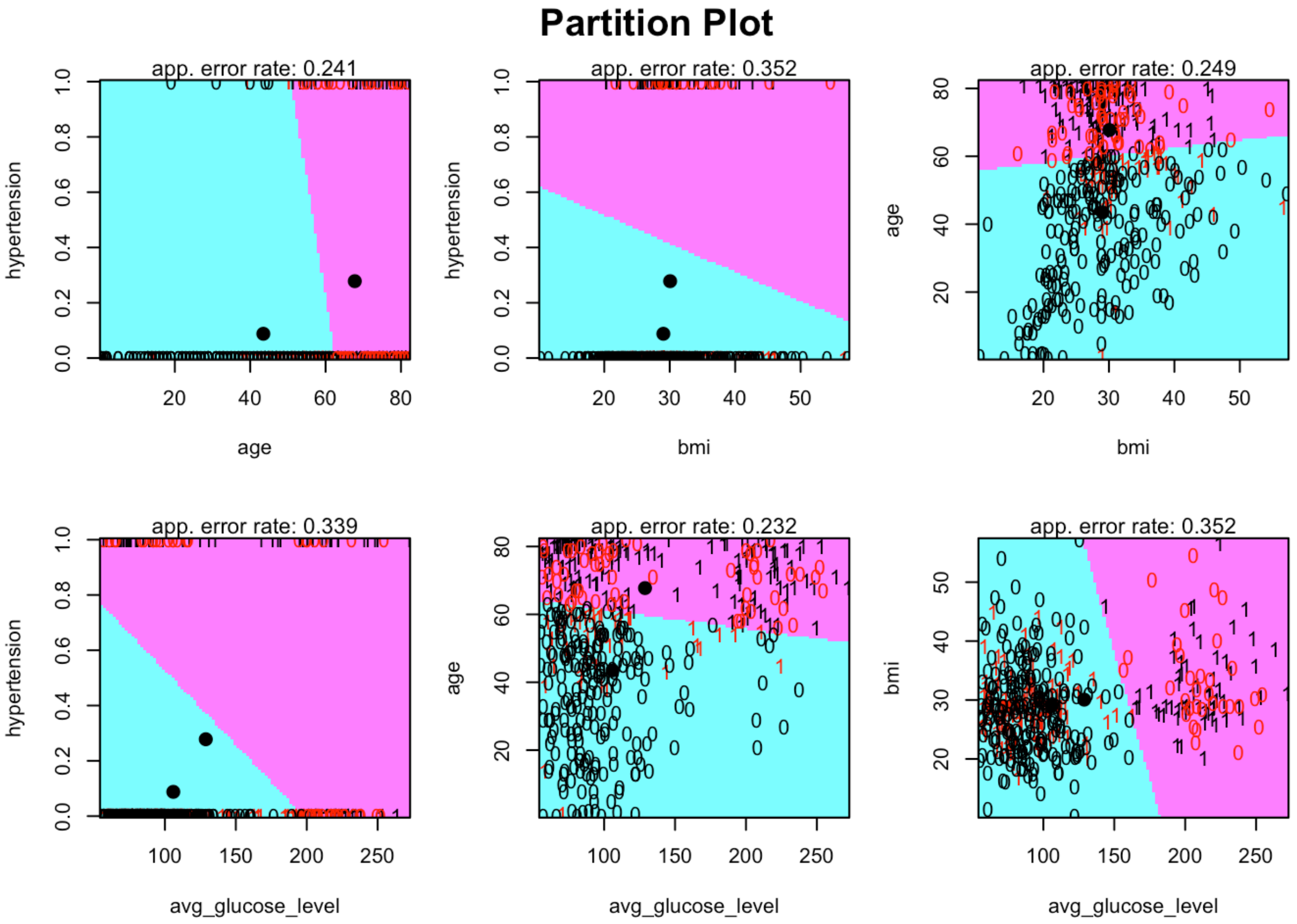
```
plot(LDA_model_1)
```





Using the partimat() function to examine the classification error rates for different predictor pairs:

```
partimat(stroke ~ hypertension + age + bmi + avg_glucose_level,data=stroke_train,meth
od="lda")
```



The misclassification error for the LDA model, considering all the predictors, is reported as 20.13% for the test set. However, it is important to note that there may be doubts about the stability of this method due to the violation of the assumption of multivariate normality and the presence of a combination of categorical and continuous data.

In the next section, we proceed with building the QDA model to compare its misclassification rate with the LDA model. However, it is important to note that due to the violation of the assumption of multivariate normality, as indicated by the tests conducted in the previous section, the results of the QDA models may not be reliable for prediction purposes.

Quadratic Discriminant Analysis (QDA) Modeling and Evaluation

In this part, we construct the QDA model using two groups of predictors. The first group contains only continuous variables, while the second group includes all predictors.

We begin by fitting the QDA model using only the continuous predictors:

```
qda_model_1<-qda(stroke~age+avg_glucose_level+bmi, data = stroke_train)
qda_model_1
```

```
## Call:
## qda(stroke ~ age + avg_glucose_level + bmi, data = stroke_train)
## 
## Prior probabilities of groups:
##         0         1 
## 0.5835189 0.4164811 
## 
## Group means:
##        age avg_glucose_level      bmi
## 0 43.48153          105.9123 29.02758
## 1 67.74503          128.8099 30.06041
```

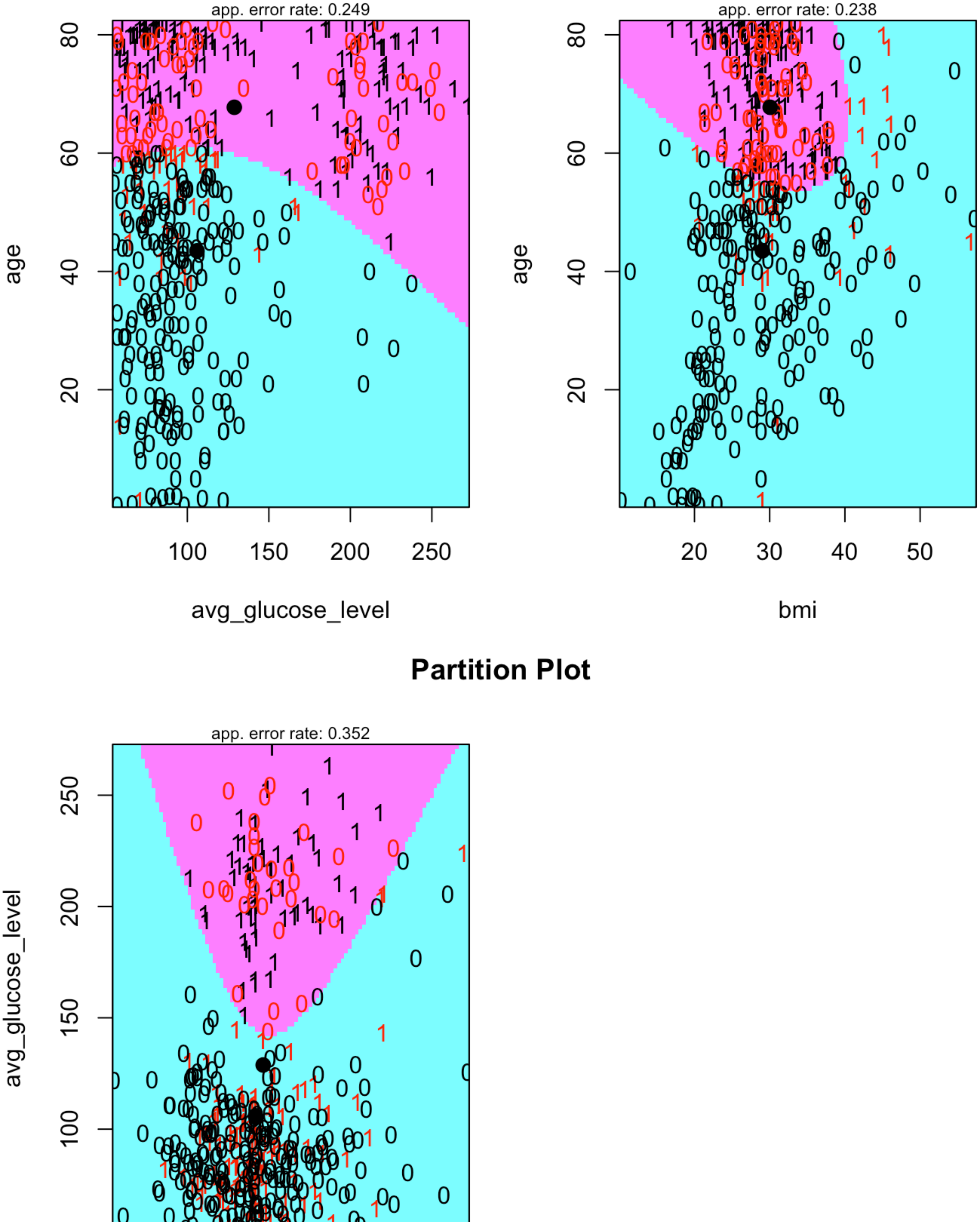
Next, we calculate the misclassification rate for the test set based on this model:

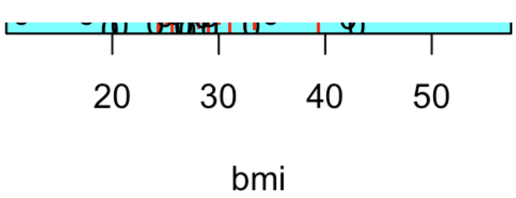
```
qda_class_1<-predict(qda_model_1, stroke_test)$class
qda_misclass_rate_1=mean(stroke_test$stroke!=qda_class_1)*100
cat("QDA misclassification rate obtained from continous predictors:\n",qda_misclass_r
ate_1,"%")
```

```
## QDA misclassification rate obtained from continous predictors:
##  20.13423 %
```

Using the partimat() function to visualize the classification error rates for different predictor pairs:

```
partimat(stroke ~ age + avg_glucose_level + bmi,data=stroke_train,method="qda")
```


app. error rate: 0.249
age
80
60
40
20
100
150
200
250
avg_glucose_level
app. error rate: 0.238
age
80
60
40
20
20
30
40
50
bmi
Partition Plot
app. error rate: 0.352
avg_glucose_level
250
200
150
100



As we can see in the above plots that the age-bmi pair seem to have the minimum error rate for classification on the train data.

QDA Model with All Predictors

```
qda_model_2<-qda(stroke~., data = stroke_train)
qda_class_2<-predict(qda_model_2, stroke_test)$class
qda_misclass_rate_2=mean(stroke_test$stroke!=qda_class_2)*100
cat("QDA misclassification rate obtained from all predictors:\n",qda_misclass_rate_2,
"%")
```

```
## QDA misclassification rate obtained from all predictors:
##  22.81879 %
```

In the first QDA model, which includes only the continuous predictors, the misclassification rate is reported as 20.13%. However, in the second QDA model, which includes all the predictors, the misclassification rate is reported as 22.81%. It is important to note that these results may not be reliable due to the violation of the assumption of multivariate normality, as indicated by the previous tests.

The inconsistency observed in the results of the QDA models could be attributed to the categorical variables, which may not align well with the assumption of a normal distribution for the fitted function in the QDA model.

The findings from this section indicate that for datasets containing both categorical and continuous variables, methods such as decision trees, which do not rely on normality or homogeneity assumptions, may yield better results.

Decision Tree

Decision trees can be used in machine learning for creating either classification or regression models. The structure is similar to a tree. Depending on the model, there can be different internal nodes with specific criteria (decisions). There are also terminal nodes (leaves) that represent the outcomes. Decision trees can provide us with valuable information about the underlying predictors and they are easy to interpret and comprehend. By examining the decision sequences we can determine the most significant variables that lead to predictions. An advantage of decision trees over linear regression models is that they can be used for non-linear relationships. However, decision treess are prone to complexity and overfitting the trainig data, which can lead to inaccurate predictions.

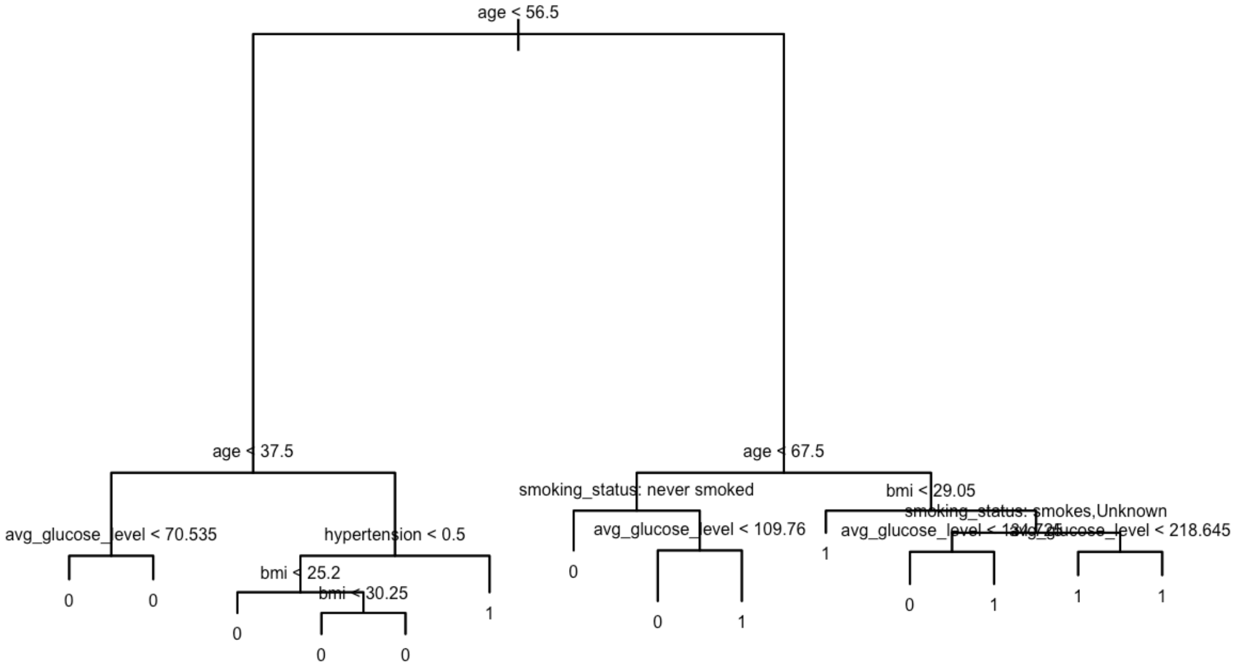
In the section below, we will build an initial regression tree to identify the variables used to predict stroke incidence. Then, we will use cross-validation to identify the best tree size for our data, and we will prune the tree if needed to minimize overfitting.

```
#build an initial tree model
tree_stroke<-tree(stroke~., data = stroke_train)
summary(tree_stroke)
```

```
##
## Classification tree:
## tree(formula = stroke ~ ., data = stroke_train)
## Variables actually used in tree construction:
## [1] "age"               "avg_glucose_level" "hypertension"
## [4] "bmi"               "smoking_status"
## Number of terminal nodes:  14
## Residual mean deviance:  0.7782 = 338.5 / 435
## Misclassification error rate: 0.1759 = 79 / 449
```

Based on the classification tree above, we found that age, average glucose level, hypertension, BMI and “smoking_status are the variables that were used in tree construction. In this initial tree model, there are 14 terminal nodes and the residual mean deviance is 0.7782.

```
#plot the tree
plot(tree_stroke)
text(tree_stroke, pretty = 0,cex=0.5)
```


age < 56.5
age < 37.5
age < 67.5
smoking_status: never smoked
bmi < 29.05
smoking_status: smokes,Unknown
avg_glucose_level < 70.535
hypertension < 0.5
avg_glucose_level < 109.76
avg_glucose_level < 13[illegible]725
avg_glucose_level < 218.645
bmi < 25.2
bmi < 30.25
0
0
0
0
0
1
0
0
1
1
0
1
1
1

```
tree_stroke
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 449 609.900 0 ( 0.58352 0.41648 )
##    2) age < 56.5 218 178.300 0 ( 0.85780 0.14220 )
##      4) age < 37.5 99  19.570 0 ( 0.97980 0.02020 )
##        8) avg_glucose_level < 70.535 16  12.060 0 ( 0.87500 0.12500 ) *
##        9) avg_glucose_level > 70.535 83   0.000 0 ( 1.00000 0.00000 ) *
##      5) age > 37.5 119 132.200 0 ( 0.75630 0.24370 )
##       10) hypertension < 0.5 105 102.300 0 ( 0.80952 0.19048 )
##         20) bmi < 25.2 26   8.477 0 ( 0.96154 0.03846 ) *
##         21) bmi > 25.2 79  87.160 0 ( 0.75949 0.24051 )
##           42) bmi < 30.25 38  50.020 0 ( 0.63158 0.36842 ) *
##           43) bmi > 30.25 41  30.410 0 ( 0.87805 0.12195 ) *
##       11) hypertension > 0.5 14  18.250 1 ( 0.35714 0.64286 ) *
##    3) age > 56.5 231 291.200 1 ( 0.32468 0.67532 )
##      6) age < 67.5 86 118.800 1 ( 0.46512 0.53488 )
##       12) smoking_status: never smoked 25  27.550 0 ( 0.76000 0.24000 ) *
##       13) smoking_status: formerly smoked,smokes,Unknown 61  78.550 1 ( 0.34426 0.
65574 )
##         26) avg_glucose_level < 109.76 35  48.260 0 ( 0.54286 0.45714 ) *
##         27) avg_glucose_level > 109.76 26  14.100 1 ( 0.07692 0.92308 ) *
##      7) age > 67.5 145 160.300 1 ( 0.24138 0.75862 )
##       14) bmi < 29.05 86  76.410 1 ( 0.16279 0.83721 ) *
##       15) bmi > 29.05 59  76.820 1 ( 0.35593 0.64407 )
##         30) smoking_status: smokes,Unknown 12  15.280 0 ( 0.66667 0.33333 )
##           60) avg_glucose_level < 134.725 7   0.000 0 ( 1.00000 0.00000 ) *
##           61) avg_glucose_level > 134.725 5   5.004 1 ( 0.20000 0.80000 ) *
##         31) smoking_status: formerly smoked,never smoked 47  55.430 1 ( 0.27660 0.
72340 )
##           62) avg_glucose_level < 218.645 37  47.970 1 ( 0.35135 0.64865 ) *
##           63) avg_glucose_level > 218.645 10   0.000 1 ( 0.00000 1.00000 ) *
```

Among all the terminal nodes, we observe the count of strokes and no-strokes are very close in the leaf 26. However, the number of zeros are slightly higher than ones, so it is classified as “no-stroke”.

```
stroke_predict<-(predict(tree_stroke,stroke_test, type="class"))
table(stroke_predict,stroke_test$stroke)
```

```
##
## stroke_predict  0  1
##              0 74 18
##              1 13 44
```

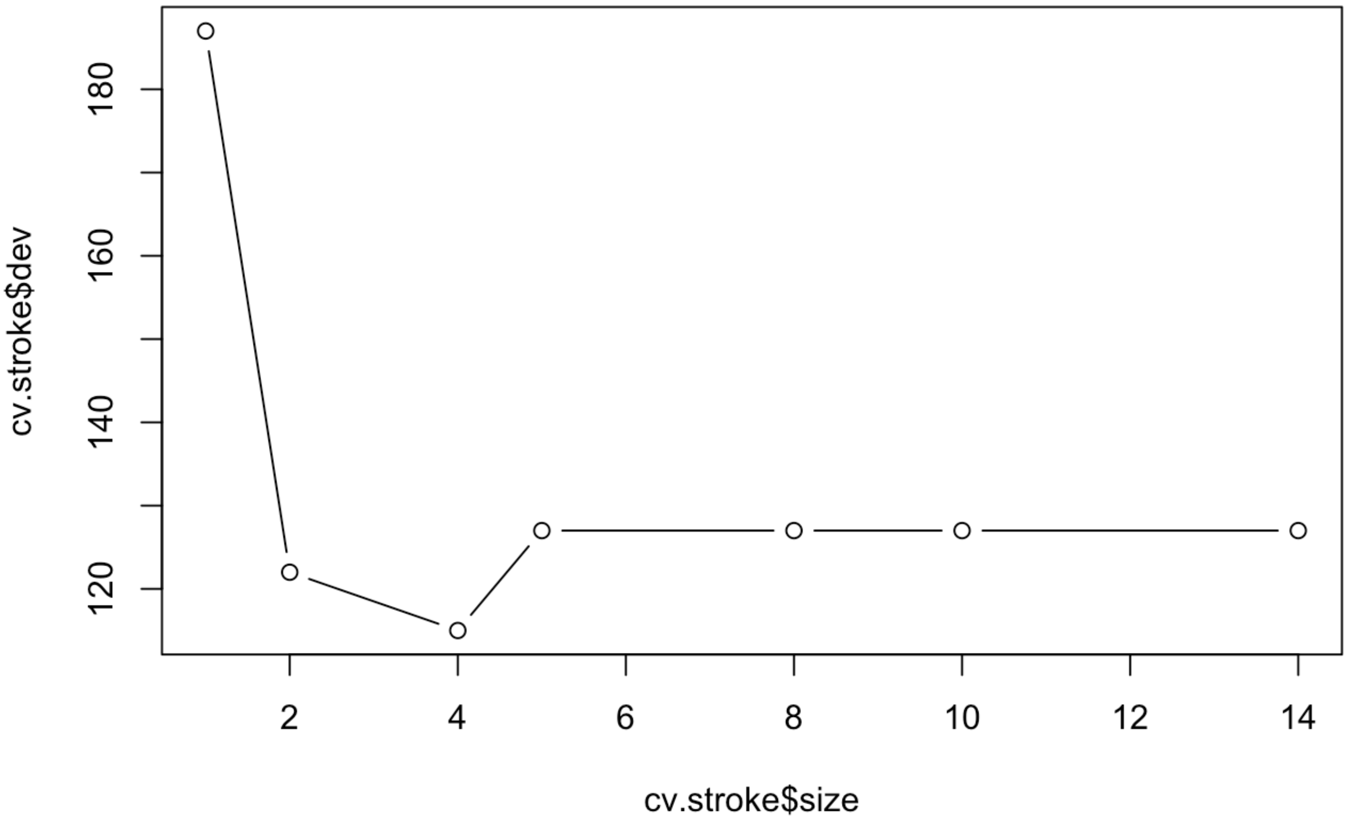
Based on the predicted values above, using the validation set, there are a total number of 149 predictions. Out of these, 118 are predicted correctly (74 are true negative and 44 true positive). However, a total number of 31 were falsely predicted (13 are false positive, and 18 false negative).

```
#misclassification rate for tree size =14

#values



cat("The misclassification error rate for a decision tree, size 14 is: ",mean(stroke_
predict!=stroke_test$stroke)*100, "%")
```

```
## The misclassification error rate for a decision tree, size 14 is:  20.80537 %
```

The misclassification error rate for our initial decision tree model, with having 14 terminal nodes was found to be 20.80%.

```
#cross-validation to choose the best tree size
set.seed(2500)
cv.stroke=cv.tree(tree_stroke, FUN=prune.misclass)
plot(cv.stroke$size,cv.stroke$dev, type="b")
```



```
cv_tree_dev<-data.frame(size=cv.stroke$size,deviance=cv.stroke$dev)
cv_tree_dev
```

size <int>	deviance <dbl>
14	127
10	127
8	127
5	127
4	115
2	122
1	187

7 rows

The graph referenced earlier depicts the cross-validation error (deviance) on the y-axis against the tree size (number of terminal nodes) on the x-axis. This graph aids in identifying the lowest error, which can be beneficial in tree pruning. By pruning, we could potentially simplify our decision tree without necessarily increasing the misclassification rate.

According to the plot, the cross-validation errors span from slightly above 187 to below 115, with tree sizes ranging from 1 to 14.

To optimize our decision tree, we attempt to prune it such that there are 4 terminal nodes. However, as demonstrated by the pruning results, a tree with only 4 terminal nodes could not be achieved. This result arises from the nature of the pruning algorithm, which employs the complexity parameter ‘alpha’ to find the best-fitted tree.

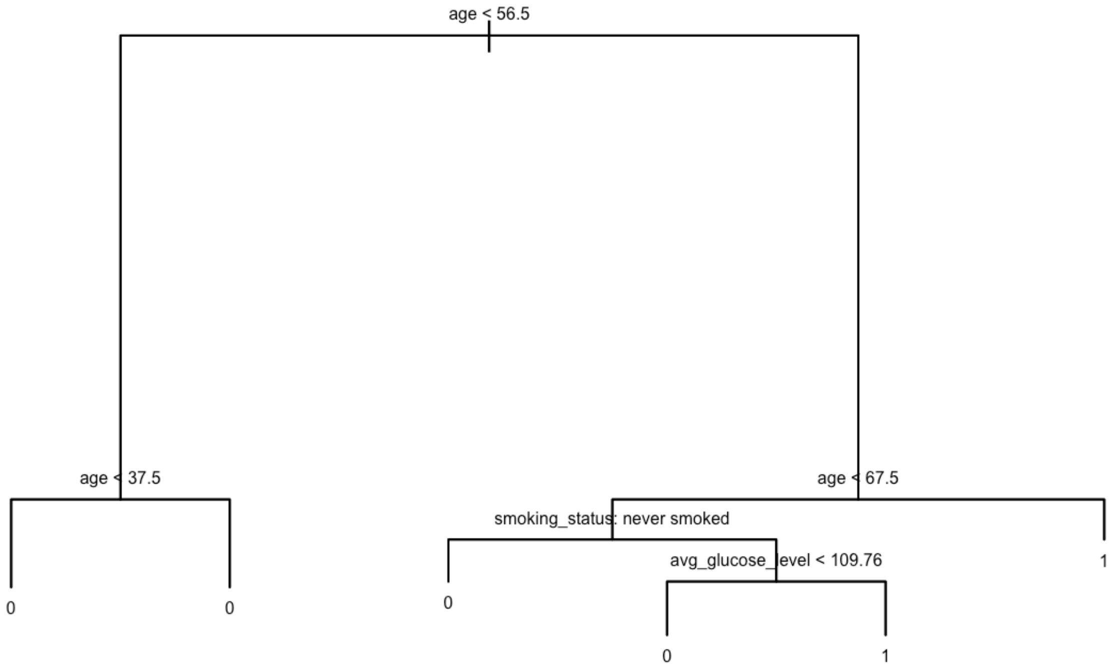
The code block provided prunes the decision tree ‘tree_stroke’ to what is determined as best, i.e., 4 nodes, and then provides a summary of the pruned tree ‘prune.stroke’.

```
prune.stroke=prune.tree(tree_stroke,best=4)
summary(prune.stroke)
```

```
##
## Classification tree:
## snip.tree(tree = tree_stroke, nodes = c(4L, 7L, 5L))
## Variables actually used in tree construction:
## [1] "age"               "smoking_status"    "avg_glucose_level"
## Number of terminal nodes:  6
## Residual mean deviance:  0.9073 = 401.9 / 443
## Misclassification error rate: 0.2004 = 90 / 449
```

However, the result shows that the pruned tree has 6 terminal nodes, not 4 as initially desired. The tree was constructed based on the ‘age’, ‘smoking_status’, and ‘avg_glucose_level’ variables. It yields a residual mean deviance of 0.9073, calculated as 401.9 divided by 443, and a misclassification error rate of 0.2004, calculated as 90 divided by 449.

```
plot(prune.stroke)
text(prune.stroke,pretty=0,cex=0.5)
```



```
prune.stroke
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 449 609.90 0 ( 0.58352 0.41648 )
##    2) age < 56.5 218 178.30 0 ( 0.85780 0.14220 )
##      4) age < 37.5 99  19.57 0 ( 0.97980 0.02020 ) *
##      5) age > 37.5 119 132.20 0 ( 0.75630 0.24370 ) *
##    3) age > 56.5 231 291.20 1 ( 0.32468 0.67532 )
##      6) age < 67.5 86 118.80 1 ( 0.46512 0.53488 )
##       12) smoking_status: never smoked 25  27.55 0 ( 0.76000 0.24000 ) *
##       13) smoking_status: formerly smoked,smokes,Unknown 61  78.55 1 ( 0.34426 0.6
5574 )
##         26) avg_glucose_level < 109.76 35  48.26 0 ( 0.54286 0.45714 ) *
##         27) avg_glucose_level > 109.76 26  14.10 1 ( 0.07692 0.92308 ) *
##      7) age > 67.5 145 160.30 1 ( 0.24138 0.75862 ) *
```

The output above presents a pruned decision tree model with terminal nodes derived from the ‘age’, ‘smoking_status’, and ‘avg_glucose_level’ variables. A comparison of the probabilities between most terminal nodes indicates they are quite different, barring node 26.

The misclassification rate on the test data is then calculated for decision trees with 2, 3, and 6 terminal nodes using the following code:

```
T_nodes=c(2,3,6)
for (node in T_nodes){
  prune.stroke=prune.tree(tree_stroke,best=node)
  pred_pruned<-predict(prune.stroke,stroke_test, type="class")
  cat("The error rate for the test set based on",node,"terminal nodes is",
      mean(pred_pruned!=stroke_test$stroke)*100,"% \n")
  }
```

```
## The error rate for the test set based on 2 terminal nodes is 22.14765 %
## The error rate for the test set based on 3 terminal nodes is 22.14765 %
## The error rate for the test set based on 6 terminal nodes is 18.79195 %
```

From this, we can observe that the error rate on the test set decreases with a greater number of terminal nodes. The initial decision tree model, with 14 terminal nodes, yielded a misclassification rate of 20.80%. However, by pruning the tree to 6 terminal nodes, we were able to reduce the error rate on unseen test data to 18.79%. This shows that pruning the tree has led to a less complex model with improved results on the test data.

Cross Validation

In the previous analyses, models were trained and tested on specific splits of the dataset. However, to ensure a more robust and reliable performance estimation, we applied 10-fold cross-validation to the entire dataset. This technique provides a more comprehensive analysis by using different subsets of the data for training and testing, thereby reducing the likelihood of overfitting and allowing for a more accurate assessment of each model’s performance.

The logistic regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Decision Tree models were all subjected to this process. Each model was trained and tested on different data folds, and the misclassification rate was calculated for each fold as shown below.

```
stroke_data$gender <- as.factor(stroke_data$gender)
stroke_data$ever_married <- as.factor(stroke_data$ever_married)
stroke_data$work_type <- as.factor(stroke_data$work_type)
stroke_data$Residence_type <- as.factor(stroke_data$Residence_type)
stroke_data$smoking_status <- as.factor(stroke_data$smoking_status)
stroke_data$stroke <- as.factor(stroke_data$stroke)
```

Creating 10 Folds which will be applied to all the models.

```
set.seed(2500)
folds<-createFolds(stroke_data$stroke, k=10)
```

Logistic Regression Model

```
set.seed(2500)
misclassificationLogistic<-function(idx){
  Train<-stroke_data[-idx,]
  Test<-stroke_data[idx,]
  fit <- glm(as.factor(stroke) ~ ., family = binomial, data = Train)
  pred<-predict(fit, newdata=Test, type = "response")
  stroke_pred <- ifelse(pred > 0.5, 1, 0)
  conf_matrix <- table(Predicted = stroke_pred, Actual = Test$stroke)
  return(1 - sum(diag(conf_matrix)) / sum(conf_matrix))
}
mis_rate=lapply(folds,misclassificationLogistic)

cat("The Misclassification rate for 10 fold Cross Validation on Logistic Regression M
odel is",mean(as.numeric(mis_rate))*100,"%")
```

```
## The Misclassification rate for 10 fold Cross Validation on Logistic Regression Mod
el is 23.91525 %
```

LDA Model

```
set.seed(2500)
misclassificationLDA<-function(idx){
  Train<-stroke_data[-idx,]
  Test<-stroke_data[idx,]
  fit<-lda(factor(stroke)~., data=Train)
  pred<-predict(fit,Test)
  return(1-mean(pred$class==Test$stroke))
}
mis_rate=lapply(folds,misclassificationLDA)
cat("The Misclassification rate for 10 fold Cross Validation on LDA Model(Full model)
is" ,mean(as.numeric(mis_rate))*100,"%")
```

```
## The Misclassification rate for 10 fold Cross Validation on LDA Model(Full model) i
s 23.58475 %
```

QDA Model

```
set.seed(2500)
misclassificationQDA<-function(idx){
  Train<-stroke_data[-idx,]
  Test<-stroke_data[idx,]
  fit<-qda(factor(stroke)~., data=Train)
  pred<-predict(fit,Test)
  return(1-mean(pred$class==Test$stroke))
}
mis_rate=lapply(folds,misclassificationQDA)
cat("Misclassification rate for 10 fold Cross Validation on QDA Model is", mean(as.numeric(mis_rate))*100,"%")
```

```
## Misclassification rate for 10 fold Cross Validation on QDA Model is 25.24576 %
```

Decision Tree

```
set.seed(2500)
misclassificationTree<-function(idx){
  Train<-stroke_data[-idx,]
  Test<-stroke_data[idx,]
  fit<-tree(stroke~., data=Train)
  pred<-predict(fit,Test,type='class')
  return(mean(pred!=Test$stroke))
}
mis_rate=lapply(folds,misclassificationTree)
cat("Misclassification rate for 10 fold Cross Validation on Decision Tree Model is ", mean(as.numeric(mis_rate))*100,"%")
```

```
## Misclassification rate for 10 fold Cross Validation on Decision Tree Model is  22.91525 %
```

The model with the lowest misclassification rate (22.91%) was the Decision Tree model, confirming its superior performance in this dataset.

Contingency Table

We investigated the associations between stroke occurrence and various demographic and health-related factors within our dataset through contingency tables. The dependency of stroke on each factor was tested using chi-squared tests, with the following findings:

Exploring the relationship between Stroke and Gender.

Are they independent or not?

H0: Stroke and Gender are Independent

Ha: Stroke and Gender are not Independent

The contingency table and chi-squared test result for the relationship between gender and stroke are as follows:

```
tab1=table(Gender=stroke_data$gender, Stroke=stroke_data$stroke)
tab1
```

```
##         Stroke
## Gender     0   1
##   Female 225 141
##   Male   124 108
```

```
chisq.test(tab1)
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  tab1
## X-squared = 3.4419, df = 1, p-value = 0.06356
```

Based on the p-value obtained from the chi-squared test (p=0.06356 > 0.05) , we fail to reject the null hypothesis. This suggests that gender and stroke are independent variables.

Exploring the relationship between Stroke and Hypertension.

Are they independent or not?

H0: Stroke and Hypertension are Independent

Ha: Stroke and Hypertension are not Independent

The contingency table and chi-squared test result for the relationship between hypertension and stroke are as follows:

```
tab2=table(Hypertension=as.factor(stroke_data$hypertension), Stroke=as.factor(stroke_
data$stroke))
tab2
```

```
##             Stroke
## Hypertension   0   1
##            0 316 183
##            1  33  66
```

```
chisq.test(tab2)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab2
## X-squared = 29.36, df = 1, p-value = 6.011e-08
```

The chi-squared test result indicates a significant relationship between hypertension and stroke occurrence. The p-value (p-value=6.011e-08) is less than 0.05, leading us to reject the null hypothesis. Therefore, stroke is dependent on hypertension.

Heart Disease and Stroke

The contingency table and chi-squared test result for the relationship between heart disease and stroke are as follows:

```
riskdifference(316,33,316+183,33+66, conf.level = 0.95)
```

```
##                  Cases People at risk        Risk
## Exposed   316.0000000    499.0000000   0.6332665
## Unexposed  33.0000000     99.0000000   0.3333333
## Total     349.0000000    598.0000000   0.5836120
```

```
##
##  Risk difference and its significance probability (H0: The difference
##  equals to zero)
##
## data:  316 33 316 + 183 33 + 66
## p-value = 8.338e-09
## 95 percent confidence interval:
##  0.1979006 0.4019658
## sample estimates:
## [1] 0.2999332
```

P_value is small enough to reject the Null hypothesis. So, here the difference is not equal to zero. Also the 95% confidence interval did not capture zero. So, we can say that there is a difference between the probability of stroke between those people who had hypertension and those who did not.

```
riskratio(316,33,316+183,33+66, conf.level = 0.95, p.calc.by.independence = TRUE)
```

```
##             Disease Nondisease Total
## Exposed        316        183   499
## Nonexposed      33         66    99
```

```
## 
##  Risk ratio estimate and its significance probability
## 
## data:  316 33 316 + 183 33 + 66
## p-value = 3.287e-08
## 95 percent confidence interval:
##  1.426582 2.529991
## sample estimates:
## [1] 1.8998
```

The risk ratio states the concept. However, the ratio between the probabilities matters here. Since it did not capture 1 in 95% of the times, we can say that the ratio between the probabilities is not 1.

```
oddsratio(as.matrix(tab2), conf.level = 0.95, p.calc.by.independence = TRUE)
```

```
##             Disease Nondisease Total
## Exposed        316        183   499
## Nonexposed      33         66    99
## Total          349        249   598
```

```
## Warning in N1 * N0 * M1 * M0: NAs produced by integer overflow
```

```
## 
##  Odds ratio estimate and its significance probability
## 
## data:  as.matrix(tab2)
## p-value = NA
## 95 percent confidence interval:
##  2.189332 5.447790
## sample estimates:
## [1] 3.453552
```

The other method is the Odds ratio which works with both of the probabilities of success and failure. The confidence interval also indicates that the odds ratio did not capture 1, these two categories are not independent. Also, it seems that in the calculation of odds ratio there was a division by zero or other errors. So it was not able to calculate the p_value.

Exploring the relationship between Stroke and heart_disease.

Are they independent or not?

H0: Stroke and heart_disease are Independent

Ha: Stroke and heart_disease are not Independent

```
tab3<-table(Heart_Disease= stroke_data$heart_disease, stroke=stroke_data$stroke)
tab3
```

```
##              stroke
## Heart_Disease   0   1
##             0 331 202
##             1  18  47
```

```
chisq.test(tab3)
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  tab3
## X-squared = 26.829, df = 1, p-value = 2.223e-07
```

The chi-squared test yields a p-value of 2.223e-07 (< 0.05), indicating a significant relationship between stroke and heart disease. Therefore, we conclude that stroke is dependent on the presence of heart disease.

Exploring the relationship between Stroke and ever_married.

Are they independent or not?

H0: Stroke and ever_married are Independent Ha: Stroke and ever_married are not Independent

```
tab4<-table(ever_married=stroke_data$ever_married, Stroke=stroke_data$stroke)
tab4
```

```
##             Stroke
## ever_married   0   1
##          No  118  29
##          Yes 231 220
```

```
chisq.test(tab4)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab4
## X-squared = 37.321, df = 1, p-value = 1.002e-09
```

The chi-squared test results in a p-value of 1.002e-09 (< 0.05), leading to the rejection of the null hypothesis. Thus, we conclude that stroke is dependent on the marital status of an individual.

Exploring the relationship between Stroke and work_type.

Are they independent or not?

H0: Stroke and work_type are Independent

Ha: Stroke and work_type are not Independent

```
tab5<-table(work_type=stroke_data$work_type, Stroke=stroke_data$stroke)
tab5
```

```
##                  Stroke
## work_type           0   1
##   children         38   2
##   Govt_job         49  33
##   Private         210 149
##   Self-employed    52  65
```

```
chisq.test(tab5)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab5
## X-squared = 31.489, df = 3, p-value = 6.704e-07
```

```
chisq.residuals(tab5, std=TRUE)
```

```
##                  Stroke
## work_type             0     1
##   children         4.87 -4.87
##   Govt_job         0.28 -0.28
##   Private          0.08 -0.08
##   Self-employed -3.40  3.40
```

The chi-squared test yields a p-value of 6.704e-07 (< 0.05), indicating a significant relationship between stroke and work type. However, the relatively large chi-squared value is influenced by the category of children. Hence, further investigation is required.

Exploring the relationship between Stroke and Residence_type.

Are they independent or not?

H0: Stroke and Residence_type are Independent

Ha: Stroke and Residence_type are not Independent

```
tab6<-table(Residence_type=stroke_data$Residence_type, stroke= stroke_data$stroke)
tab6
```

```
##               stroke
## Residence_type   0   1
##          Rural 180 114
##          Urban 169 135
```

```
chisq.test(tab6)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab6
## X-squared = 1.7262, df = 1, p-value = 0.1889
```

Based on the high p-value=0.1889 (> 0.05), we fail to reject the null hypothesis. Therefore, we conclude that stroke is independent of whether a person resides in an urban or rural area.

Exploring the relationship between Stroke and smoking_status.

Are they independent or not?

H0: Stroke and smoking_status are Independent

Ha: Stroke and smoking_status are not Independent

```
tab7<-table(stroke_data$smoking_status, stroke_data$stroke)
tab7
```

```
##
##                        0   1
##   formerly smoked  52  70
##   never smoked    136  90
##   smokes           60  42
##   Unknown         101  47
```

```
chisq.test(tab7)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab7
## X-squared = 18.698, df = 3, p-value = 0.0003156
```

```
chisq.residuals(tab7, std=TRUE)
```

```
##
##                          0     1
##   formerly smoked -3.95  3.95
##   never smoked     0.70 -0.70
##   smokes           0.10 -0.10
##   Unknown          2.81 -2.81
```

The chi-squared test results in a p-value of 0.0003156 (< 0.05), leading us to reject the null hypothesis. Thus, we conclude that stroke is dependent on whether a person is a smoker or not.

Here are the summarized findings:

1. **Gender**: No significant correlation was found between gender and stroke occurrences.
2. **Hypertension**: A significant relationship was discovered, indicating that individuals with hypertension have a higher risk of experiencing a stroke.
3. **Heart Disease**: The analysis revealed a strong correlation, signifying that people with heart diseases are more likely to have a stroke.
4. **Marital Status**: The results suggest a significant relationship between marital status and stroke occurrences, with people who have been married at least once having a higher stroke risk.
5. **Work Type**: A significant correlation was found between work type and stroke risk.
6. **Residence Type**: No significant relationship was found between residence type (urban or rural) and stroke occurrences.
7. **Smoking Status**: A significant relationship was discovered, showing that smokers are at higher risk for

a stroke.

This analysis provides a better understanding of the various factors influencing stroke risk, which can guide interventions and preventive measures.

Conclusion and Limitations

Based on the analysis conducted on the given dataset, it can be concluded that the Decision Tree model is the most effective choice for predicting stroke risk. This conclusion is supported by the model's outstanding performance, as evidenced by the lowest misclassification error rate of 18.79% and a 10-fold cross-validation error rate of 22.91%. Conversely, alternative models like logistic regression, LDA, and QDA exhibited instability when applied to the stroke dataset, which consisted of both quantitative and qualitative variables. These findings emphasize the importance of selecting an appropriate modeling technique based on the nature of the data and the specific prediction task at hand. Additionally, when comparing different link functions in binary regression models, it was observed that the logit function outperformed others by accommodating more predictors and demonstrating a higher converging rate. The superiority of the Decision Tree model can be attributed to its ability to handle complex relationships without making assumptions about data distribution and its resilience to outliers. Thus, the Decision Tree model emerges as the most reliable choice for predicting stroke risk based on the given dataset.

Future Scope

1. Include other important factors like Blood Pressure, Cholesterol Levels and Family History and various psychological factors to better understand stroke incidence.
2. Examine other sexes and genders. While this binary categorization is common in many health-related datasets, it does not fully represent the diversity of sex and gender identities.
3. Use more advance ML algorithm to reduce classification error rate
4. Utilize real world data to train the models and make better prediction

Refrencess

1. Canada, P. H. A. of. (2022). Stroke in Canada. Government of Canada. Retrieved from: https://www.canada.ca/en/public-health/services/publications/diseases-conditions/stroke-in-canada.html (https://www.canada.ca/en/public-health/services/publications/diseases-conditions/stroke-in-canada.html)

2. World Stroke Organization. (2023). Learn about stroke. Retrieved from: https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters/learn-about-stroke (https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters/learn-about-stroke)

3. World Health Organization. (2022). World stroke day 2022. Retrieved from: https://www.who.int/srilanka/news/detail/29-10-2022-world-stroke-day-2022 (https://www.who.int/srilanka/news/detail/29-10-2022-world-stroke-day-2022)

4. Kaggle. (2021). Stroke prediction dataset. Retrieved from https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset (https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset)
5. Zach. (202). Multivariate Normality Test in R. Retrieved from https://www.statology.org/multivariate-normality-test-r/ (https://www.statology.org/multivariate-normality-test-r/)
6. R-Bloggers. (2021). Equality of Variances in R: Homogeneity Test Quick Guide. Retrieved from https://www.r-bloggers.com/2021/06/equality-of-variances-in-r-homogeneity-test-quick-guide/ (https://www.r-bloggers.com/2021/06/equality-of-variances-in-r-homogeneity-test-quick-guide/)