Stroke Predictions

Analytical Theory and Methods
Professor : **Anne VANHEMS**

GROUP 9:

- **Patricia Fonseca GOMES**
- **Ngoc Uyen PHUNG**
- **Renzo RAUSCHENBERG**
- **Evan Brahma Hughie AZHABUR**

Table of contents

01 Project Goals

This project will identify a prevailing world problem, namely a health issue about **stroke**. Specifically, in this project, a patient will be predicted whether they are likely to be diagnosed with stroke based on input parameters.

The hypothesis of this project is

1. Input factor such as Age and Hypertension have significant effect on stroke
2. Input factor such as work type and residence type have no significant effect on stroke

The purpose of this project is to create a model which suitable with the data using several methods such as Logistic regression and Generalized Additive Model

Additionally, this project aims to provide more insight about health so that there will be more policies related to stroke developed in the future.

02 Dataset

Link : https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data

Stroke Prediction Dataset:
Comprises 11 clinical features for predicting stroke events, including patient's age, gender, health conditions like hypertension and heart disease, marital status, work type, residence type, average glucose level, body mass index, and smoking status (contains both quantitative and qualitative variables).

Dataset Context:
Developed in response to strokes being the 2nd leading cause of death globally (World Health Organization. 2020), this dataset aims to predict the likelihood of stroke in patients based on various input parameters. Each data row provides relevant patient information for stroke prediction.

Features

1. id: unique identifier
2. gender: "Male", "Female" or "Other"
3. age: age of the patient
4. hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5. heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6. ever_married: "No" or "Yes"
7. work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8. Residence_type: "Rural" or "Urban"
9. avg_glucose_level: average glucose level in blood
10. bmi: body mass index
11. smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12. stroke: 1 if the patient had a stroke or 0 if not

Data Exploration: Data Type

	id	gender	age	hypertension	heart_disease	ever_married	work_type
1	9046	Male	67	0	1	Yes	Private
2	51676	Female	61	0	0	Yes	Self-employed
3	31112	Male	80	0	1	Yes	Private
4	60182	Female	49	0	0	Yes	Private
5	1665	Female	79	1	0	Yes	Self-employed

Residence_type	avg_glucose_level	bmi	smoking_status	stroke
Urban	228.69	36.6	formerly smoked	1
Rural	202.21	N/A	never smoked	1
Rural	105.92	32.5	never smoked	1
Urban	171.23	34.4	smokes	1
Rural	174.12	24	never smoked	1

- Dataset with 12 columns and 5110 rows (observations).
- The dataset includes a mix of genders, ages, and health conditions. Variables such as hypertension, heart disease, and smoking status vary among individuals.
- ID column can be deleted.
- We have already seen missing data at first sight.

Data Exploration: Data Type

```
> str(data)
'data.frame':   5110 obs. of  12
 $ id               : int  9046 5
53882 10434 27419 60491 ...
 $ gender           : chr  "Male'
...
 $ age              : num  67 61
 $ hypertension     : int  0 0 0
 $ heart_disease    : int  1 0 1
 $ ever_married     : chr  "Yes"
 $ work_type        : chr  "Priva
e" "Private" ...
 $ Residence_type   : chr  "Urbar
...
 $ avg_glucose_level: num  229 20
 $ bmi              : chr  "36.6'
 $ smoking_status   : chr  "forme
"never smoked" "smokes" ...
 $ stroke           : int  1 1 1
```

- In order to continue the data, we need to change some variable type, changing into factor might help
- Also some of these variables might be better in a binary form. We further need to explore that.

Data Exploration: Data Summary

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

- To make further explorations we need to first change the data type for gender, hypertension, heart_disease, ever_married, work_type, Residence_type, bmi and smoking_status.

03 Data Cleaning

```r
## Changing data type
data$gender <- as.factor(data$gender)
data$hypertension <- factor(data$hypertension, levels=c(0, 1),
                            labels=c('No Hypertension', 'Hypertension'))
data$heart_disease <- factor(data$heart_disease, levels=c(0, 1),
                             labels=c('No Heart Disease', 'Heart Disease'))
data$ever_married <- as.factor(data$ever_married)
data$work_type <- as.factor(data$work_type)
data$Residence_type <- as.factor(data$Residence_type)
data$smoking_status <- as.factor(data$smoking_status)
data$stroke <- as.factor(data$stroke)
```

Changing the Data Type:

1. Data including gender, ever_married, work_type, residence_type, smoking_status, heart_disease, hypertension, and stroke change into factor in order to represent categorical variable. Factors in R are designed to store and represent these categories efficiently.
2. Change the BMI into numeric, to notify R changing the from character to numeric

```r
data$bmi <- as.numeric(data$bmi)
```

Data Exploration: Data Summary after changing

```
> summary(data)
       id            gender          age                   hypertension             heart_disease  ever_married
 Min.   :   67   Female:2994   Min.   : 0.08   No Hypertension:4612   No Heart Disease:4834   No :1757
 1st Qu.:17741   Male  :2115   1st Qu.:25.00   Hypertension   : 498   Heart Disease   : 276   Yes:3353
 Median :36932   Other :   1   Median :45.00
 Mean   :36518                 Mean   :43.23
 3rd Qu.:54682                 3rd Qu.:61.00
 Max.   :72940                 Max.   :82.00

         work_type    Residence_type avg_glucose_level       bmi                 smoking_status stroke
 children     : 687   Rural:2514     Min.   : 55.12    Min.   :10.30   formerly smoked: 885   0:4861
 Govt_job     : 657   Urban:2596     1st Qu.: 77.25    1st Qu.:23.50   never smoked   :1892   1: 249
 Never_worked :  22                  Median : 91.89    Median :28.10   smokes         : 789
 Private      :2925                  Mean   :106.15    Mean   :28.89   Unknown        :1544
 Self-employed: 819                  3rd Qu.:114.09    3rd Qu.:33.10
                                     Max.   :271.74    Max.   :97.60
                                                       NA's   :201
```

As we can see in the image, we need to continue cleaning the new dataset

- Gender: delete „other“ (one value is not representative).
- Binary variables: hypertension, heart_disease, ever_married, Residence_type.
- Bmi: need to deal with missing values.
- Smoking_status: unknown need to be dealt with too.

Data Cleaning: After Changing Type (1)

Change of data type into binary is not necessary, models can deal with it. For having a uniform dataset, the binary variables hypertension and heart_disease are transformed into factors.

```
data$hypertension <- factor(data$hypertension, levels=c(0, 1),
                            labels=c('No Hypertension', 'Hypertension'))
data$heart_disease <- factor(data$heart_disease, levels=c(0, 1),
                             labels=c('No Heart Disease', 'Heart Disease'))
```

Deleting ID column

```
# deleting ID column
cleandata <- subset(cleandata, select = -id)
```

Data Cleaning: After Changing Type (2)

Delete the „other“ from gender column:

```
# drop row of gender = "other"
cleandata <- subset(cleandata, gender != "Other")
cleandata$gender <- factor(cleandata$gender, levels = c("Male", "Female"))
```

Missing values:

BMI: due to the fact, that only 4% of the data has is NA, we will delete the missing values and later do a sensitivity analysis

Smoking_status: due to the fact, that 30% of the data points are unknown, we can‘t delete them. We will treat them as an own category.

```
# Drop the NA
cleandata <- data[complete.cases(data),]
```

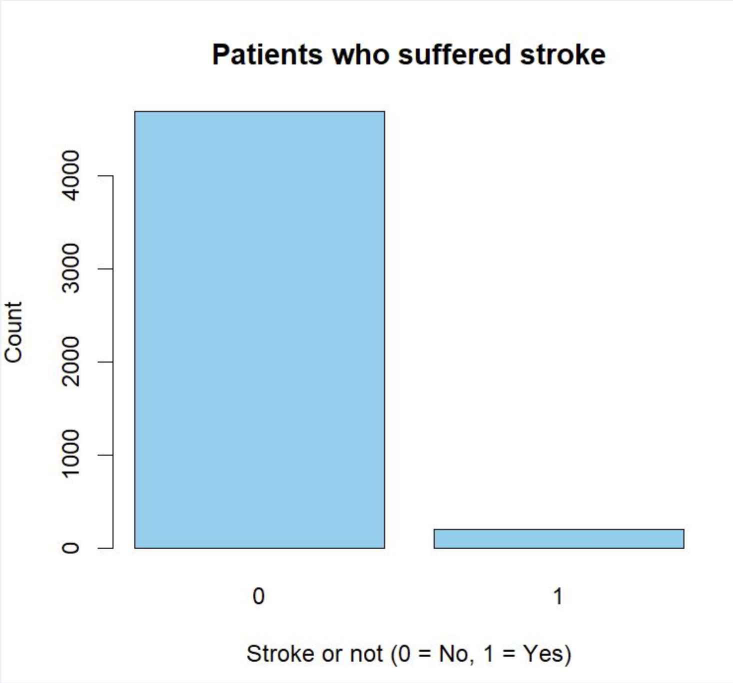
Data Cleaning: Summary of Cleaned Data

```
> summary(cleandata)
    gender           age                   hypertension
 Male  :2011   Min.   : 0.08   No Hypertension:4457
 Female:2897   1st Qu.:25.00   Hypertension   : 451
               Median :44.00
               Mean   :42.87
               3rd Qu.:60.00
               Max.   :82.00
          heart_disease  ever_married        work_type
 No Heart Disease:4665   No :1704     children      : 671
 Heart Disease   : 243   Yes:3204     Govt_job      : 630
                                      Never_worked  :  22
                                      Private       :2810
                                      Self-employed: 775

 Residence_type avg_glucose_level      bmi
 Rural:2418     Min.   : 55.12    Min.   :10.30
 Urban:2490     1st Qu.: 77.07    1st Qu.:23.50
                Median : 91.68    Median :28.10
                Mean   :105.30    Mean   :28.89
                3rd Qu.:113.50    3rd Qu.:33.10
                Max.   :271.74    Max.   :97.60
          smoking_status stroke
 formerly smoked: 836   0:4699
 never smoked   :1852   1: 209
 smokes         : 737
 Unknown        :1483
```

- Gender: more men than women, still appropriate proportion.
- Most people have no hypertension, heart disease, were married and haven't had a stroke.
- Most of the people in the data are the ones who not having stroke (4699), and the ones who suffer stroke within the rate of 4% (209 people). So the proportion is not balanced

Following: Distribution Graphs of Age, work type, avg glucose level, bmi and smoking status.

04 Descriptive Analytics

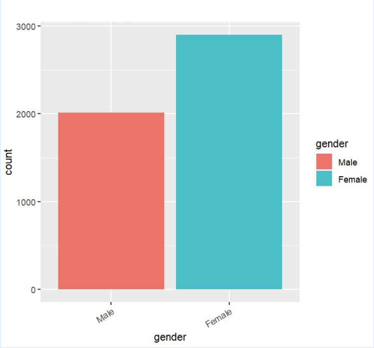


Descriptive for target variable

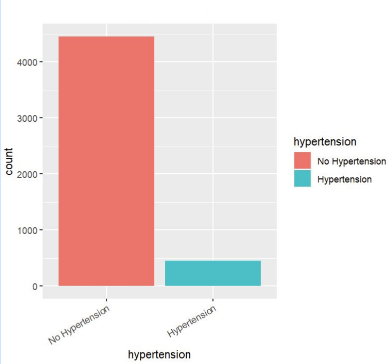
- The data has the result of people not suffered from stroke way more than the people who suffered the stroke
- Our limitations cause by the data quality since the number of people suffered from stroke only 4%

Distribution Graphs

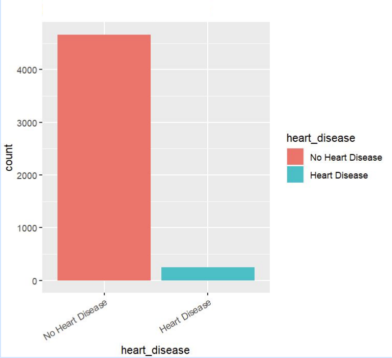
Gender



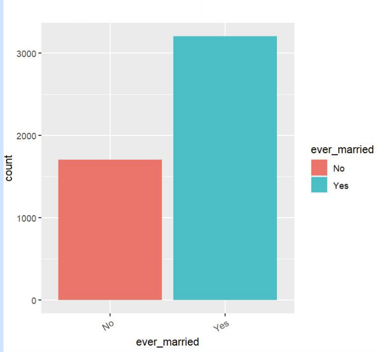
Hypertension



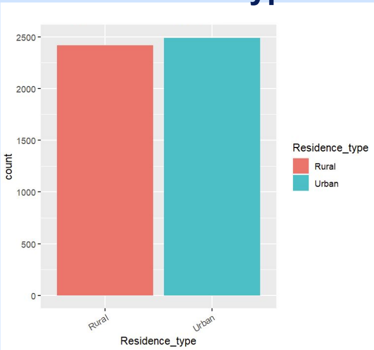
Heart disease



Marital Status

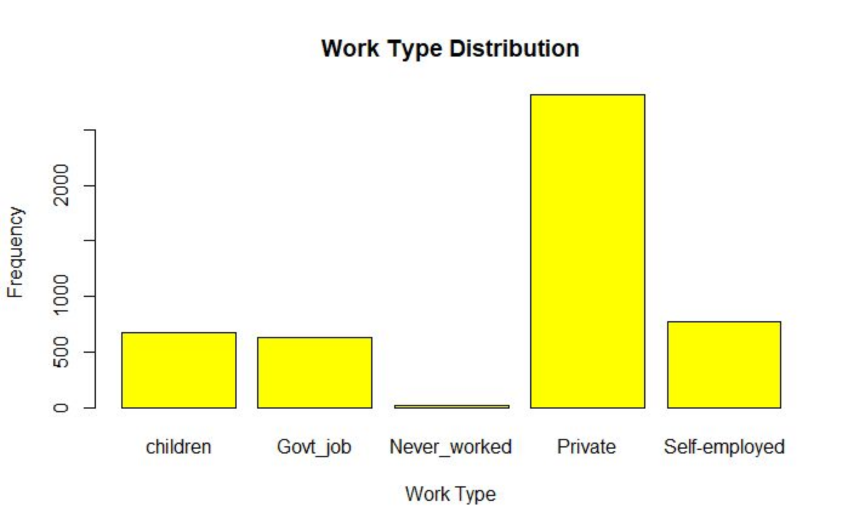


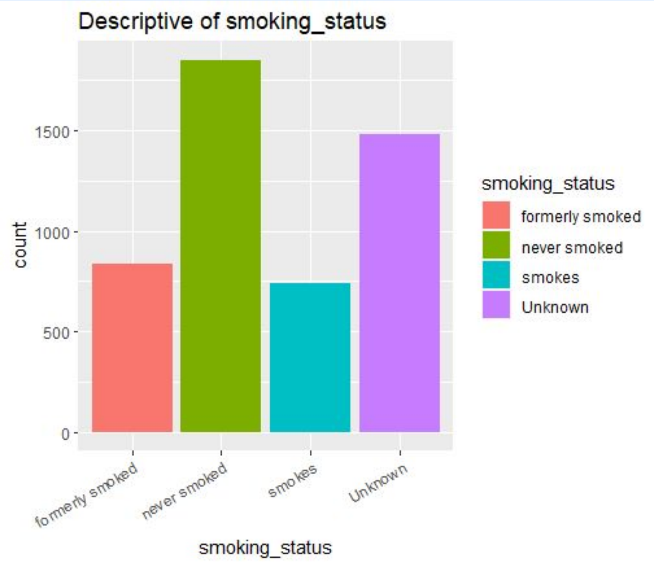
Residence type



- The dataset reveals a higher prevalence of individuals without hypertension or heart disease compared to those affected.
- More participants have been married at least once, and the distribution between urban and rural residents is evenly split.
- Additionally, there is a slight majority of female participants over males.

Distribution Graphs





- Overwhelming majority works in the Private industry. Almost no people who have never worked.

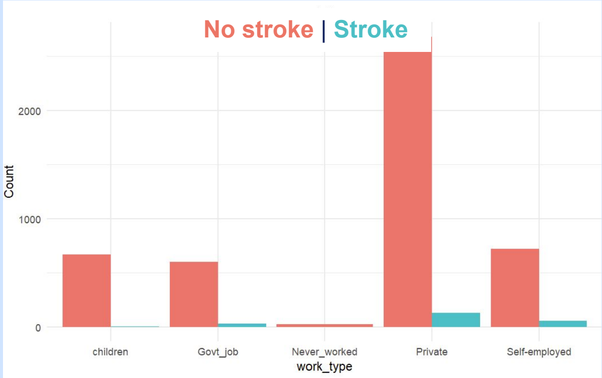
- There is a great number of unknown
- From the people who we know the status form: Most people have never smoked, the number of people who still smoke is fairly low (about 30%)

Distribution Graphs: Categorical variables

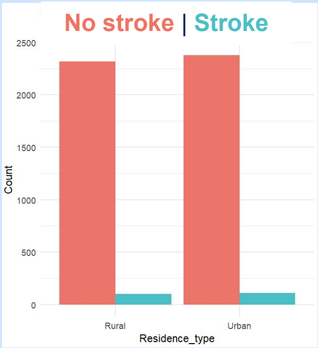
Gender risk

Marital status risk

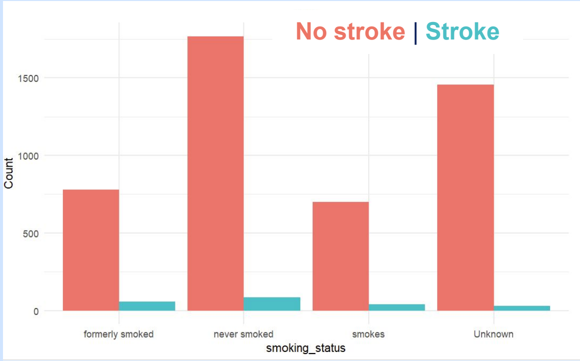
Employment risk



Residence type

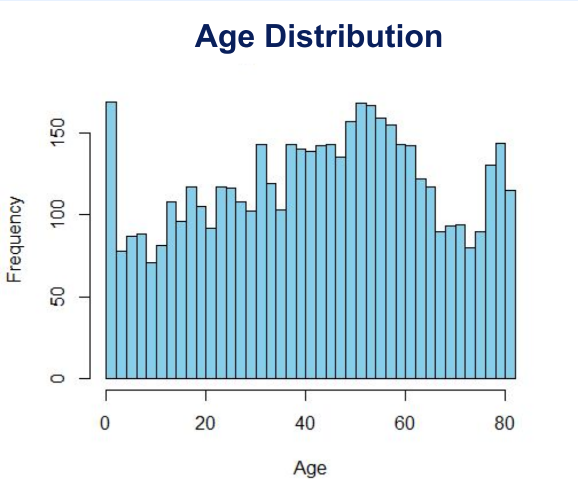


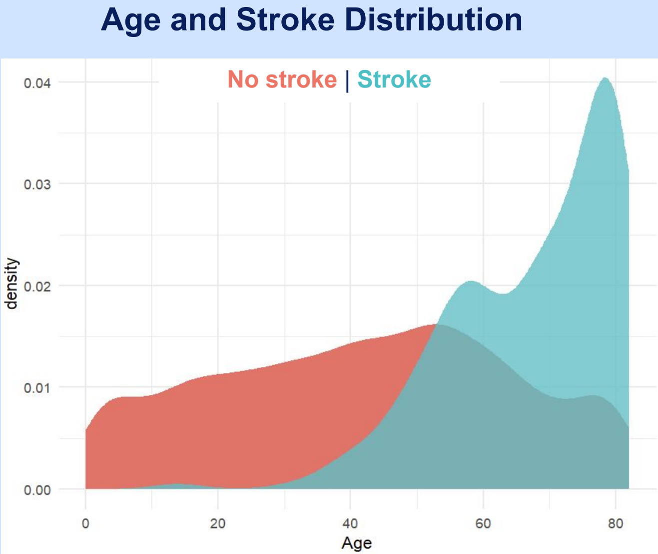
Smoking status risk



This is an overview of the categorical features, displaying the number of strokes and no strokes for the category features. Since there is a unbalanced values between categories, we cannot give any immediately assumption about the other variables.

Distribution Graphs: Age

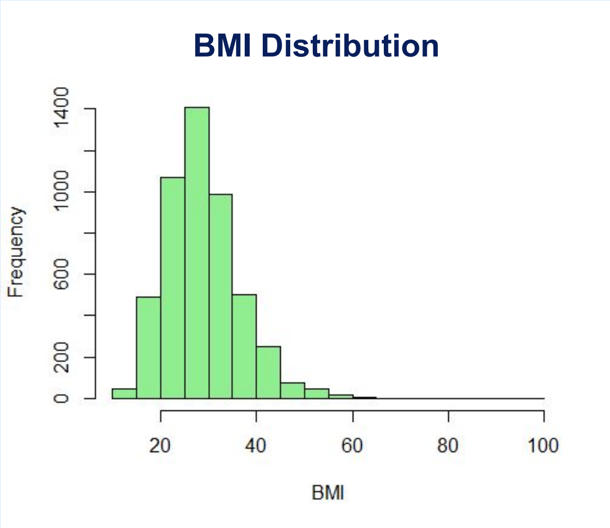




From those features, we can see that:

- High number of babies and old seniors, normal distribution towards the middle, Mean = 42.87
- Old people are mostly having strokes, compared to younger ones.

Distribution Graphs: BMI

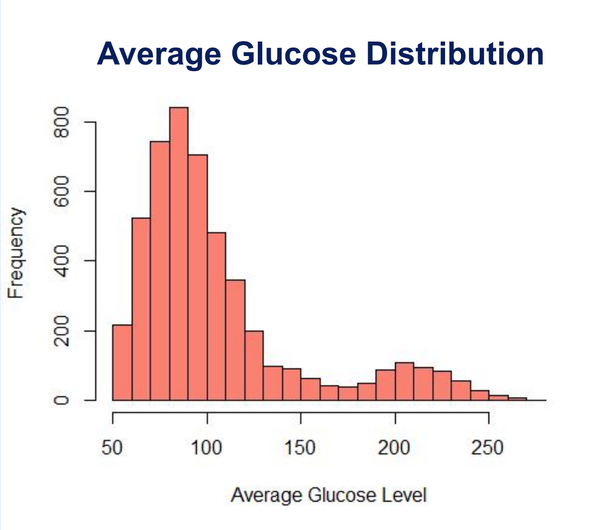


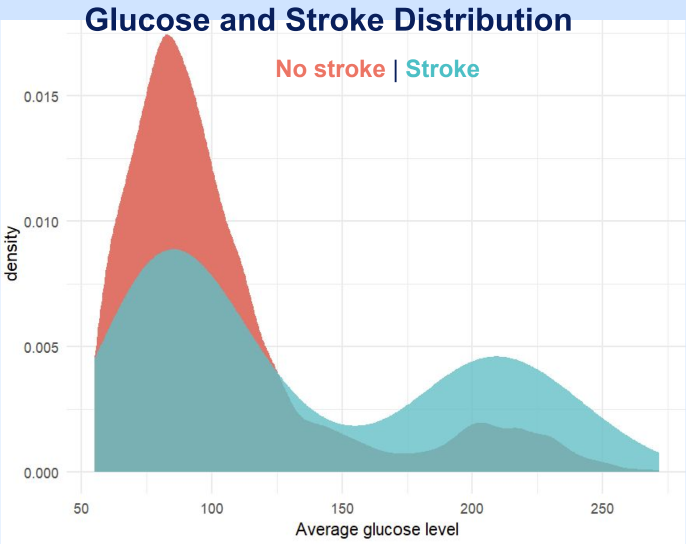
BMI and Stroke Distribution

From those features, we can see that:

- Mostly normal distribution with some obese patients.
- The higher the BMI, the higher possibility that a person have strokes.

Distribution Graphs: Glucose level





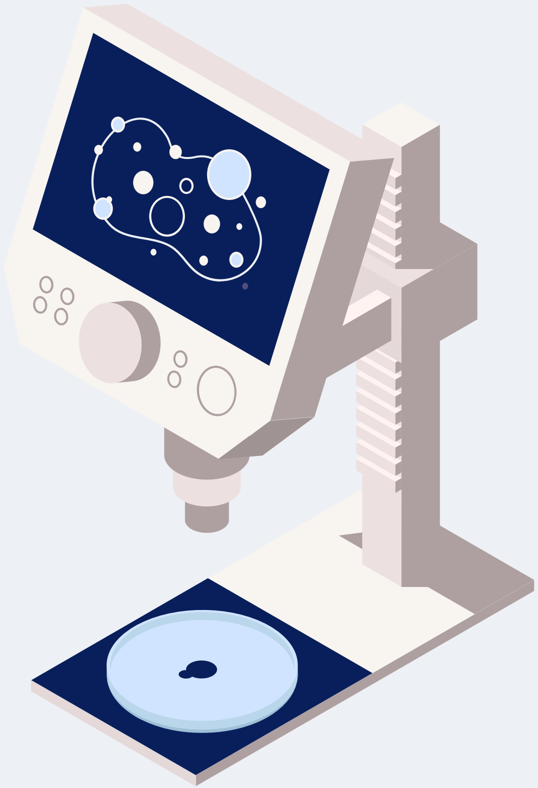
From those features, we can see that:

- Mostly normal distribution with some high glucose level patients (maybe diabetes).
- Strokes can be seen with people have higher average glucose level.

05

Methodology

Describe the model and evaluation



Methodology

- In this case, a **logit regression model** is tried out first due to the cause that the target variable is binary.
- Furthermore , a **GAMs model** is applied to test, to see if it performs better than the logit models.
- We divided the dataset into 2 parts: training and testing. We used 80% of the data to train the model, while the rest is for the model evaluation.
- Moreover, we set seed of 123 for reproducibility.

```
# Create train and test data
set.seed(123)  # Set seed for reproducibility
# 80% for data training
train_indices <- sample(1:nrow(cleandata), 0.8 * nrow(cleandata))
train_data <- cleandata[train_indices, ]
test_data <- cleandata[-train_indices, ]
```

Method 1: Logistics regression (model 1)

```
> mod.1 <- glm(stroke~
+              gender + age + hypertension + heart_disease + avg_glucose_level+
+              bmi + smoking_status, family=binomial, data=cleandata)
> summary(mod.1)

Call:
glm(formula = stroke ~ gender + age + hypertension + heart_disease +
    avg_glucose_level + bmi + smoking_status, family = binomial,
    data = cleandata)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -7.835140   0.586329 -13.363  < 2e-16 ***
genderFemale                0.011195   0.154002   0.073 0.942052
age                         0.069041   0.005846  11.810  < 2e-16 ***
hypertensionHypertension    0.517649   0.174433   2.968 0.003001 **
heart_diseaseHeart Disease  0.372836   0.206067   1.809 0.070405 .
avg_glucose_level           0.004697   0.001289   3.644 0.000268 ***
bmi                         0.003458   0.011744   0.294 0.768426
smoking_statusnever smoked -0.057792   0.187965  -0.307 0.758493
smoking_statussmokes        0.321264   0.228501   1.406 0.159736
smoking_statusUnknown      -0.256978   0.245238  -1.048 0.294697
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1728.3  on 4907  degrees of freedom
Residual deviance: 1369.4  on 4898  degrees of freedom
AIC: 1389.4

Number of Fisher Scoring iterations: 7
```

- Using logistic regression: Variables age, hypertension and avg_glucose_level are highly significant (p<0,05)
- Age: positive relationship
- Hypertension: strong positive relationship
- Avg_glucose: positive relationship
- The higher the age and glucose level and people who have hypertension are more likely to suffer a stroke.
- Deviance: Model better than Null model, because deviance decreases
- AIC = 1369.4

Method 1: Logistics regression (model 2)

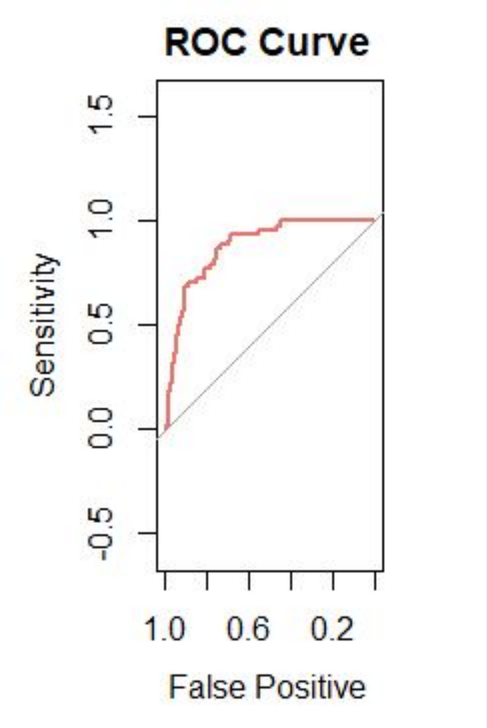
```
> mod.2 <- glm(stroke~
+                age + hypertension + avg_glucose_level, family=binomial,
+              data=cleandata)
> summary(mod.2)

Call:
glm(formula = stroke ~ age + hypertension + avg_glucose_level,
    family = binomial, data = cleandata)

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)               -7.766457   0.385221 -20.161  < 2e-16 ***
age                        0.069565   0.005490  12.671  < 2e-16 ***
hypertensionHypertension   0.547010   0.172629   3.169  0.00153 **
avg_glucose_level          0.005047   0.001245   4.052 5.07e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1728.3  on 4907  degrees of freedom
Residual deviance: 1378.4  on 4904  degrees of freedom
AIC: 1386.4

Number of Fisher Scoring iterations: 7
```

- Model 2 is created as the improved version of model 1 since the insignificant variables is excluded.
- Only significant variables are included in model 2 ($p<0.5$).
- Similar deviance values.
- AIC = 1386.4, which indicates a better model compare to the previous one.

Model 2 is preferred because AIC is slightly smaller.

We choose to continue with Model 2.

Method 1: Model 2 Predictive power



The image is the ROC Curve from method 1, model 2 using logistic regression

As we can see the image represent:

- The x-axis is labeled "False Positive," which represents the false positive rate (FPR). The y-axis is labeled "Sensitivity," which is another term for the true positive rate (TPR).
- The curve is not smooth, which suggests that it might be representing the performance of a classifier over a discrete set of threshold values rather than a continuous probability distribution.

Based on the image, the curve represent a good quality of predictive power inside the model even though it seem not as smooth but the curve is ideally close.

Method 1: Model 2 Predictive power

```
> print(paste("Best threshold:", best_threshold)) # 0.037
[1] "Best threshold: 0.0373359194892259"
> print(paste("Youden's index:", youden_index)) # 0.619
[1] "Youden's index: 0.619839116107773"
```

By using youden’s index we try to find the balanced threshold, it result in 0.037336. which might suggest that the model is set to predict 'stroke' even if the probability is very low. This could lead to a higher sensitivity (more true positives) but also potentially more false positives. → *try the threshold = 0.037*

For the result of the youden’s index is relatively high with the result of 0.61984, indicating that the model has a good ability to discriminate between the outcomes.

```
> print(paste('Prediction accuracy =', 1-misclass.err))
[1] "Prediction accuracy = 0.739307535641548"
```

The prediction accuracy we get is 0.7393. It means that by using classic logistics regression, the model correctly predicted the outcome approximately 73.93% of the time .

Method 1: Confusion Matrix

```
> conf_matrix <- table(test_data$stroke, pred_test)
> print(conf_matrix)
   pred_test
      0   1
  0 687 251
  1   5  39
```

- True Negatives (TN): The upper left cell (687) which is the number of instances that were correctly predicted as not having a stroke.
- False Positives (FP): The upper right cell (251) where the model incorrectly predicted a stroke when there was not one.
- False Negatives (FN): The lower left cell (5) which occur when the model predicts no stroke, but the patient actually did have a stroke.
- True Positives (TP): The lower right cell (39) where the model correctly predicted a stroke.

Based only from the confusion matrix, we can conclude that:

- The model is more often correct on negative cases (no stroke) but it has a significant number of false positives.
- The number of false negatives is low maybe it is due to the unbalanced dataset predicting stroke
- The true positives number is moderate. This suggests that when the model predicts a stroke, it's more likely to be correct

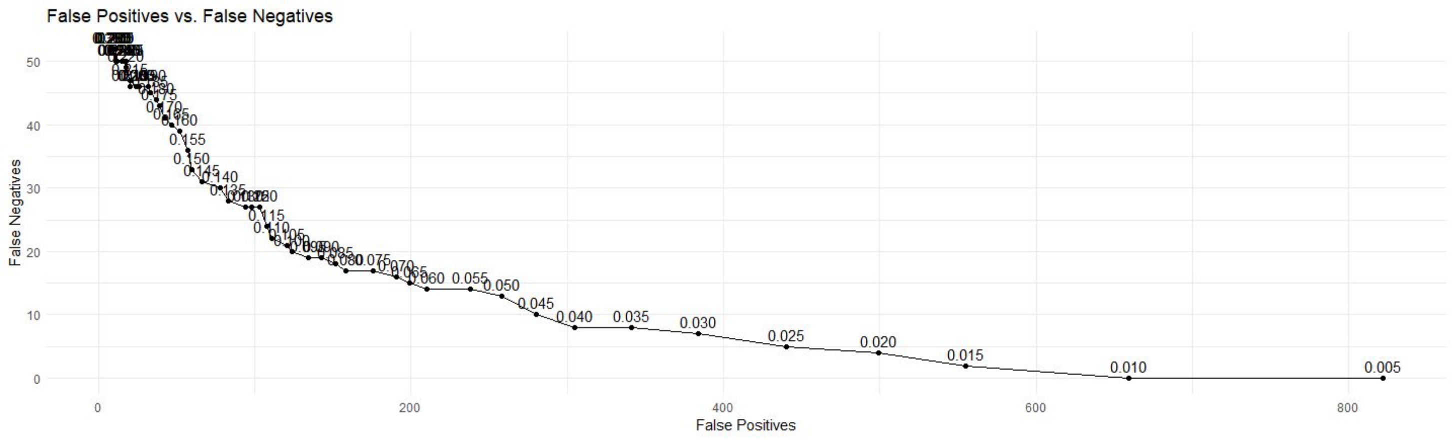
Method 1: Confusion Matrix (Calculate metrics)

```
> # Print metrics
> print(paste("Sensitivity (True Positive Rate):", sensitivity)) # 88.63%
[1] "Sensitivity (True Positive Rate): 0.886363636363636"
> print(paste("Specificity (True Negative Rate):", specificity)) # 73.24%
[1] "Specificity (True Negative Rate): 0.732409381663113"
> print(paste("Precision (Positive Predictive Value):", precision)) #13.44%
[1] "Precision (Positive Predictive Value): 0.13448275862069"
> print(paste("Recall (Sensitivity):", recall)) # 88.63%
[1] "Recall (Sensitivity): 0.886363636363636"
```

- Sensitivity (True Positive Rate) : the sensitivity is about 88.64%. This indicates that the model is good at identifying positive cases (in this case, identifying patients who did have a stroke)
- Specificity (True Negative Rate) : The specificity is about 73.24%. This suggests that the model is reasonably good but not excellent at identifying negative cases (patients who did not have a stroke).
- Precision (Positive Predictive Value) : The precision here is about 13.44%. This low precision indicates that when the model predicts a stroke, it is correct only about 13.44% of the time.

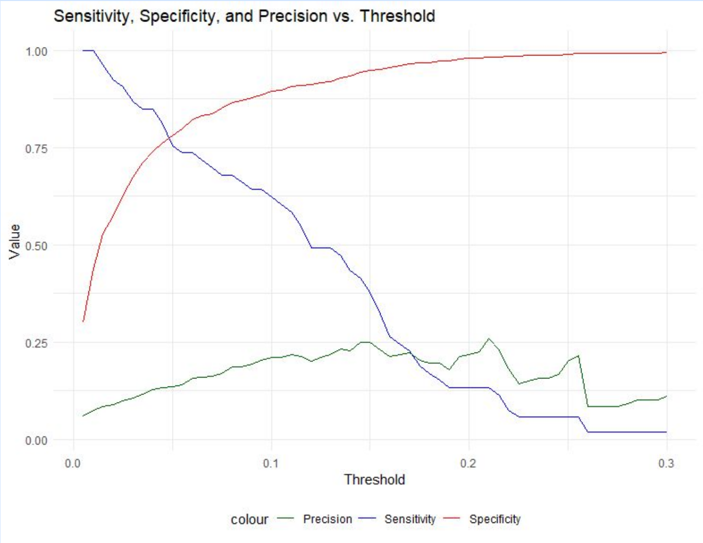
Method 1: Finding the appropriate threshold

To improve the model, we tried to find the best threshold:



- Only at a very low threshold of 0.01 there is no false negative
- The accuracy of the model to predict that will be very low because the number of false positives increases drastically.
- A threshold of 0.145 would have been better to minimize number of false positives too

Method 1: Finding the appropriate threshold



It seems that there is a trade-off happening as the threshold is adjusted:

- When the threshold is low, sensitivity is high — the classifier is capturing most of the positive cases, but the precision is low, indicating many false positives.
- As the threshold increases, specificity increases (fewer false negatives), but sensitivity decreases (more false negatives).
- Precision improves with the threshold but is not as high or stable as the other metrics, indicating that even as the classifier becomes more selective, it struggles to maintain precision.
- The ideal balance between these metrics depends on the specific application and the cost of false positives vs. false negatives. In medical diagnostics, high sensitivity (fewer false negatives) is often prioritized to ensure that most disease cases are caught, even if it means more false positives (lower precision).
- The graph doesn't show the entire range of threshold values, only up to 0.3, which implies that we're seeing a partial view of the classifier's performance. It would be useful to look at these metrics across the entire range of possible threshold values to make a comprehensive evaluation of the classifier's performance.

Method 1: Improved model

```
> print(conf_matrix)
  pred_test
      0   1
 0 393 547
 1   3  39
```

We try the other threshold value, which is 0.01

- Accuracy = approximately 44% → lower than the previous threshold
- Higher TP rate, Sensitivity; but Lower TN rate and precision.

- True Negatives (TN): The upper left cell (393) which is the number of instances that were correctly predicted as not having a stroke.
- False Positives (FP): The upper right cell (547) where the model incorrectly predicted a stroke when there was not one.
- False Negatives (FN): The lower left cell (3) which occur when the model predicts no stroke, but the patient actually did have a stroke.
- True Positives (TP): The lower right cell (39) where the model correctly predicted a stroke.

Method 1: Improve model

```
> print(paste("Sensitivity (True Positive Rate):", sensitivity)) # 92.85%
[1] "Sensitivity (True Positive Rate): 0.928571428571429"
> print(paste("Specificity (True Negative Rate):", specificity)) # 41.80%
[1] "Specificity (True Negative Rate): 0.418085106382979"
> print(paste("Precision (Positive Predictive Value):", precision)) # 6.65%
[1] "Precision (Positive Predictive Value): 0.0665529010238908"
> print(paste("Recall (Sensitivity):", recall)) # 92.85%
[1] "Recall (Sensitivity): 0.928571428571429"
```

- **Sensitivity** (True Positive Rate): indicating that the model correctly identifies 92.85% of the positive cases (e.g., people who have had a stroke, if we are talking about a stroke prediction model).
- **Specificity** (True Negative Rate): suggesting that the model correctly identifies 41.80% of the negative cases (e.g., people who have not had a stroke)
- **Precision** (Positive Predictive Value): The precision is about 6.65%, which is quite low. This means that of all the cases the model predicted as positive, only 6.65% were actually positive

Method 2: Generalized Additive Model

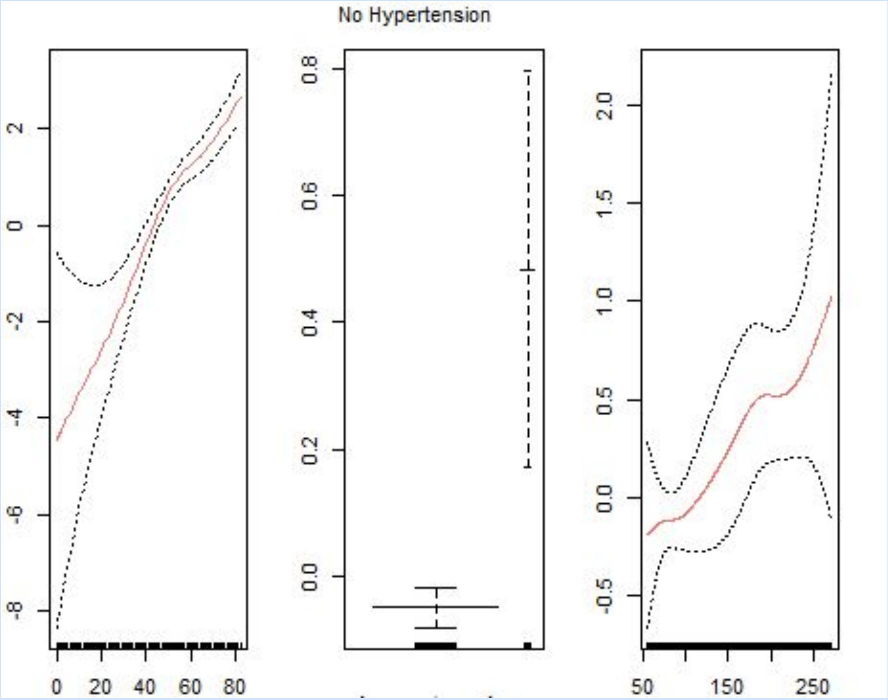
```
##---------------------------GAM model building-----------------------
# GAM
mod.gam <- gam(stroke ~ s(age) + hypertension + s(avg_glucose_level),
               family = binomial, data = cleandata)
summary(mod.gam)
# AIC: 1390.273
# Logistics model is preferred according to the AIC.

par(mfrow=c(1,3))
plot.Gam(mod.gam, se=TRUE, col="#ec756b")
```

```
Call: gam(formula = stroke ~ s(age) + hypertension + s(avg_glucose_level),
    family = binomial, data = cleandata)
Deviance Residuals:
     Min       1Q   Median       3Q      Max
-0.99192 -0.31695 -0.15755 -0.04828  3.87479

(Dispersion Parameter for binomial family taken to be 1)

    Null Deviance: 1728.299 on 4907 degrees of freedom
Residual Deviance: 1370.273 on 4898 degrees of freedom
AIC: 1390.273

Number of Local Scoring Iterations: NA

Anova for Parametric Effects
                       Df Sum Sq Mean Sq F value    Pr(>F)
s(age)                  1  154.0 153.953 144.895 < 2.2e-16 ***
hypertension            1   14.0  14.032  13.207 0.0002818 ***
s(avg_glucose_level)    1   16.6  16.641  15.661 7.683e-05 ***
Residuals            4898 5204.2   1.063
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Anova for Nonparametric Effects
                     Npar Df Npar Chisq P(Chi)
(Intercept)
s(age)                     3     5.7981 0.1218
hypertension
s(avg_glucose_level)       3     1.7648 0.6226
```

- Creating GAM model with the target variable of stroke. The independent variable are defined from previous code which is proven have significant effect to the model. The independent variable are age, hypertension and avg_glucose _level.
- AIC: 1390.273, higher than logistic.

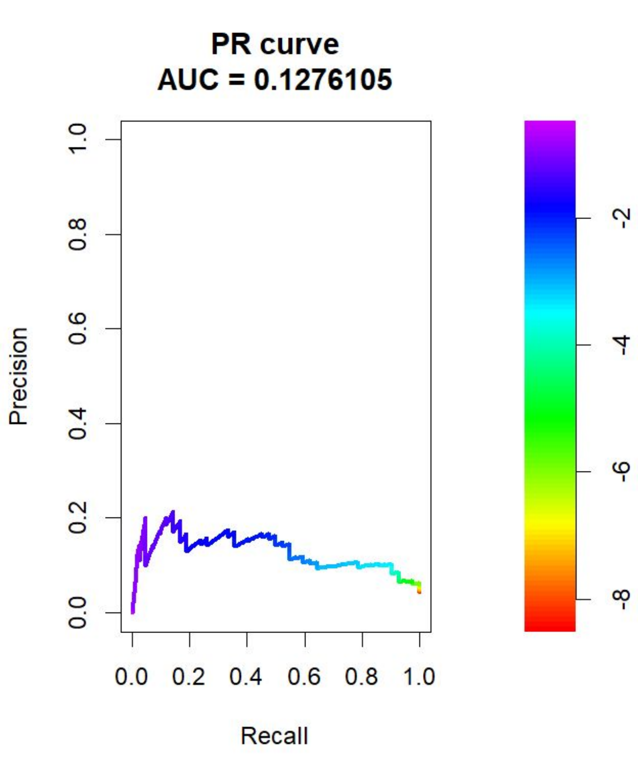
→ According to AIC, the logistic model is preferred.

Method 2: Generalized Additive Model



- Left Plot (age): This plot shows the smooth term for age (s(age)), indicating the relationship between age and the response variable (probably the probability of having a stroke). The curve appears to go below and above zero, indicating varying effects of age across its range.
- Middle Plot (hypertension): Since hypertension is likely a binary variable (as indicated by the earlier code snippet that did not apply a smoothing function to it), this plot displays its estimated effect on the log-odds of the response.
- Right Plot (avg_glucose_level): This is similar to the first plot for age, showing the smooth term for average glucose level (s(avg_glucose_level)). The non-linear relationship suggests that the effect of average glucose level on the likelihood of a stroke is not constant and varies across the range of glucose levels.

Method 2: Generalized Additive Model (PR Curve)



Precision-Recall (PR) curve, which is a graph that illustrates the tradeoff between precision (positive predictive value) and recall (true positive rate) for a binary classifier system as its discrimination threshold is varied.

The AUC (Area Under the Curve) value of 0.2117983, which is quite low, which suggests that the model has poor performance in terms of precision and recall across different thresholds. This value indicates that **model is not performing well** at distinguishing between "stroke" and "no stroke" classes.

→ there is a lot of room for improvement in the classifier's performance.

Method 2: Generalized Additive Model

```
> table(pred.gam, test_data$stroke)

pred.gam              No stroke Stroke
  Predicted no stroke       940     42
  Predicted stroke            0      0
> prop.table(table(pred.gam, test_data$stroke),2)

pred.gam              No stroke Stroke
  Predicted no stroke         1      1
  Predicted stroke            0      0
```

We first try with threshold = 0.037 to see how the model performed.

- Misclassification Error = 0.042
- Prediction accuracy = 0.9572301 ~ 95.72%
- Confusion matrix shows that the model does not perform great with the prediction of strokes (only one is classified correctly)
- We want to reduce the number of false negative predictions in order to identify high-risk patients.

Method 2: Generalized Additive Model

Threshold 0.01

- No false negative anymore but a lot of false positives!
- Sensitivity (true positive rate): 1, which is the desired outcome
- Specificity: 0.43, because we have a lot of false positives due to the very low threshold
- Accuracy: 0.493, low because of the many false positives

	pred_test	
	0	1
0	515	659
1	0	53

Conclusion

Using the data data given of stroke prediction dataset, we made 2 type of final models. First one is generating the logistic regression model and the second one is using generalized additive model. Based on our discussion we choose the first one (logistic regression) since it have number of AIC (1386.4) lower than the second model (GAM).

Based on our chosen model of Logistic Regression, there are some significant variable that affect the model which are age, hypertension, and average glucose level. Other variable is discarded due to insignificance. Based on our hypothesis of this project from the beginning.

The hypothesis of this project is:

1. Input factor such as Age and Hypertension have significant effect on stroke
2. Input factor such as work type and residence type have no significant effect on stroke

Hypothesis number 1 is accepted due to proven and hypothesis number 2 is rejected due to the insignificant result.

Conclusion

Conclusion given by model 2 (logistic regression),

- Based on ROC, the shape of the curve is good even though it is not smooth, this indicates that this model have good predictive power
- The ideal threshold is 0.037336, which is ideal threshold based on youden index. But since the model is used to detect stroke (medical usage) which is sensitive to mistake than the model needs to be more conservative meaning the lower the threshold might be better. But after we tried to improve the model by lowering the threshold, the accuracy become lower and the true positive also become higher so it is better to stick with the 0.037336 threshold.

Unique findings: even though the result of the model is good enough with also quite good predictive power such as AUC, but due to the unbalanced data, (only 4% people in the data which suffer from stroke) it effect the outcome of the result including:

Precision (Positive Predictive Value) : The precision here is about 13.44%. This low precision indicates that when the model predicts a stroke, it is correct only about 13.44% of the time.

This indicates that further improvement of the model need high quality data.

Way forward

What we did to improve the models:

- Experiment with different thresholds.

To meet the project goals as we mentioned previous, there is some further steps to enhance the model quality:

- Address class imbalance: The dataset is highly imbalanced, consider techniques such as SMOTE for oversampling the minority class, adjusting class weights in the model, or using anomaly detection methods.
- Consider model refinement: Try different types of models or combine the models (bagging, boosting, ...)

References

World Health Organization. (2020). *The Top 10 Causes of Death*. Available at: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

THANK YOU

Have a Good Day!