Understanding Strokes

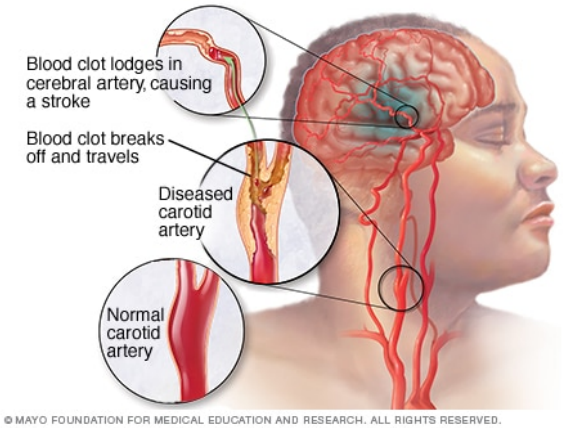
Nhi Vu
Math 448
Instructor: Dr. Tao He

I. INTRODUCTION.

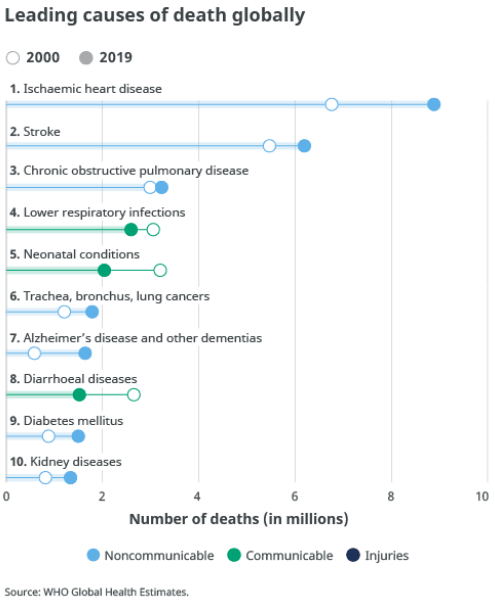
1. Problems working on.

What is a stroke?

A stroke happens when there is a loss of blood flow to part of the brain. Your brain cells cannot get the oxygen and nutrients they need from blood, and they start to die within a few minutes. This can cause lasting brain damage, long-term disability, or even death.



According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths, and from 2000 to 2019, the number of people who die due to stroke rises by more than 2 million to 8.9 million.



Who is at risk for a stroke?

Certain factors can raise your risk of a stroke. The major risk factors include

- High blood pressure. This is the primary risk factor for a stroke.
- Diabetes.
- Heart diseases. Atrial fibrillation and other heart diseases can cause blood clots that lead to stroke.
- Smoking. When you smoke, you damage your blood vessels and raise your blood pressure.
- A personal or family history of stroke or TIA.
- Age. Your risk of stroke increases as you get older.
- Race and ethnicity. African Americans have a higher risk of stroke.

There are also other factors that are linked to a higher risk of stroke, such as

- Alcohol and illegal drug use
- Not getting enough physical activity
- High cholesterol
- Unhealthy diet
- Having obesity

2. Objective & Strategy.

In my project, I want to understand more about stroke that who is likely to get a stroke based on the input variable like gender, age, various diseases, and smoking status of patients.

Besides, I visualize the feature of patients to see risk factors related to strokes, and see if we can successfully detect stroke on some features using classification methods in R.

3. The Data

The data was a free available data set provided by Kaggle website, and the last update is 01-26-2021. https://www.kaggle.com/fedesoriano

By Collection methodology, the data contains 5110 observations with 12 attributes (12 columns), which is 11 clinical features for predicting stroke events, including patients' demographic data (gender, age, marital status, type of work and residence type) and health records (hypertension, heart disease, average glucose level measured after meal, Body Mass Index (BMI), smoking status and experience of stroke).

Generally, the data set is almost clean, only there are 201 "N/A" values in the bmi column, which makes this column to be parsed as character, although it should be numerical, and a lot of "Unknown" values in smoking status which we have to take care of too. Also, the common feature of attributes are categorical data, only 3 columns (age, average glucose level and bmi) are numeric variables. Therefore, the task is classification.

Attribute Information:

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

4. Data preparation.

Change to appropriate format.

The "bmi" column, variables supposed to be numeric value but its class is characters showing in the "summary table".

Cleaning data.

Outlier data: only 1 patient categorize "Other" in gender column.

Missing data:

3.9%.

bmi	Smoking_stattus
201 N/A	1544 "unkown"

30%

Remove Keep.

Since there are 201 N/A in bmi columns which is a small data, but the number of patients categorized as "unknown" in smoking_status column takes 30% of the dataset. It was a huge proportion of the dataset, then we keep them, rather than dropping them altogether.

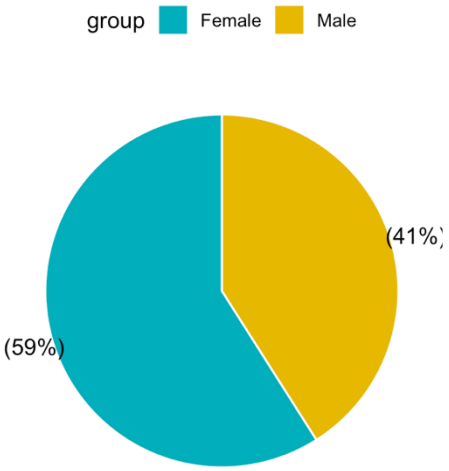
After remove outlier and missing data, we have a total 4908 observation.

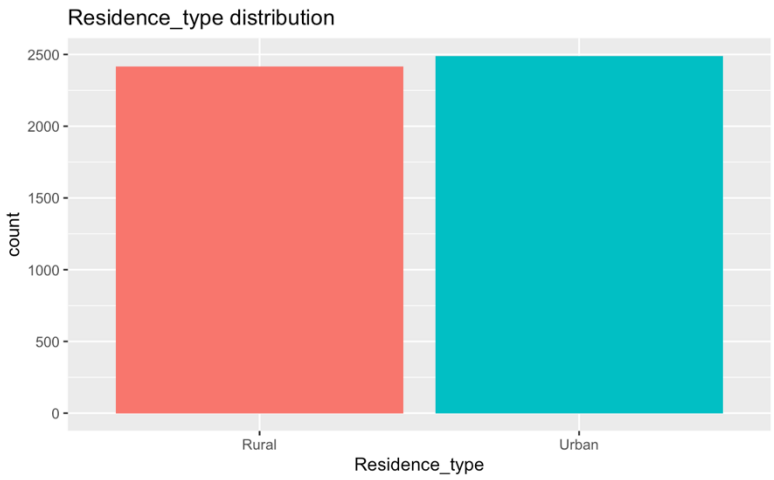
```
       id            gender               age          hypertension     heart_disease
 Min.   :   77   Length:4908        Min.   : 0.08   Min.   :0.00000   Min.   :0.00000
 1st Qu.:18602   Class :character   1st Qu.:25.00   1st Qu.:0.00000   1st Qu.:0.00000
 Median :37580   Mode  :character   Median :44.00   Median :0.00000   Median :0.00000
 Mean   :37060                      Mean   :42.87   Mean   :0.09189   Mean   :0.04951
 3rd Qu.:55182                      3rd Qu.:60.00   3rd Qu.:0.00000   3rd Qu.:0.00000
 Max.   :72940                      Max.   :82.00   Max.   :1.00000   Max.   :1.00000
 ever_married        work_type         Residence_type     avg_glucose_level      bmi
 Length:4908        Length:4908        Length:4908        Min.   : 55.12    Min.   :10.30
 Class :character   Class :character   Class :character   1st Qu.: 77.07    1st Qu.:23.50
 Mode  :character   Mode  :character   Mode  :character   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
 Length:4908        Min.   :0.00000
 Class :character   1st Qu.:0.00000
 Mode  :character   Median :0.00000
                    Mean   :0.04258
                    3rd Qu.:0.00000
                    Max.   :1.00000
```

II. VISUALIZATION.

1. Categorical variables.

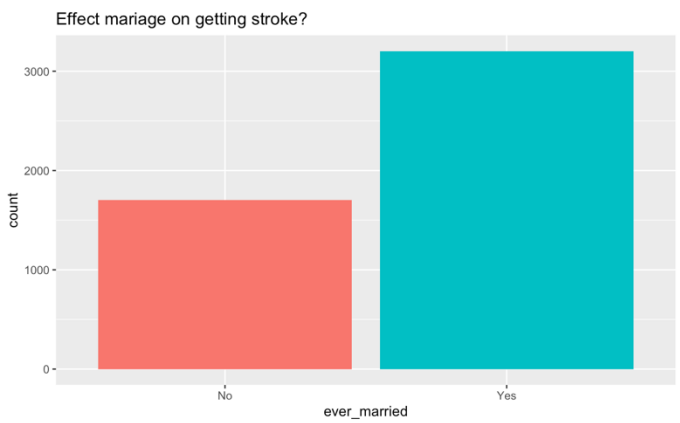
a) Gender & Residence type.

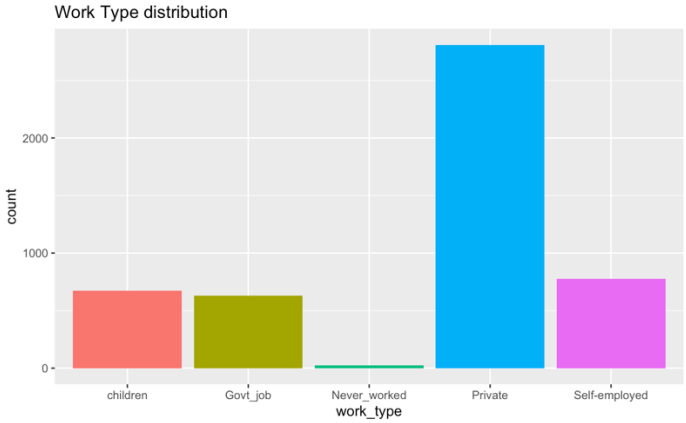




Insight #1: Regardless of patient's gender, and where they stayed, they have the same likelihood to experience stroke.

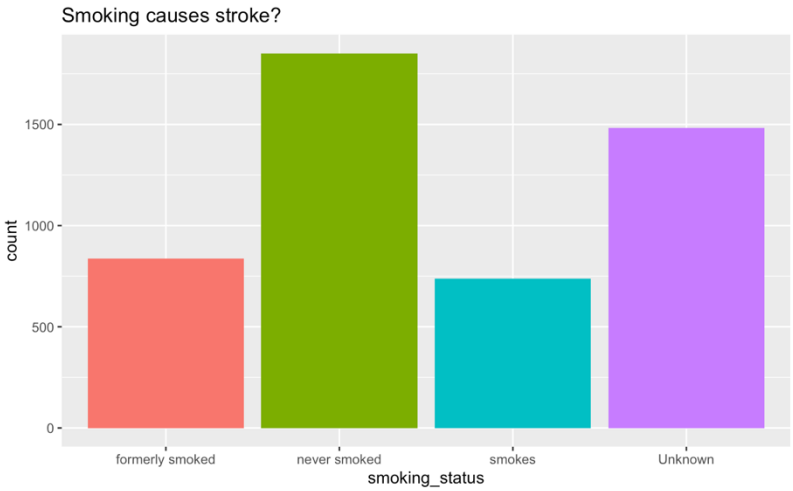
b) Marital status & Work type.





Insight #2: Married people and people who are self-employed most likely to have strokes than others. These results lead us to another hypothesis that stroke relative to mental health? because maybe these patients are suffering depression from work stress and conflicts in marriage.

c) Smoking_status
Does smoking habit cause strokes?



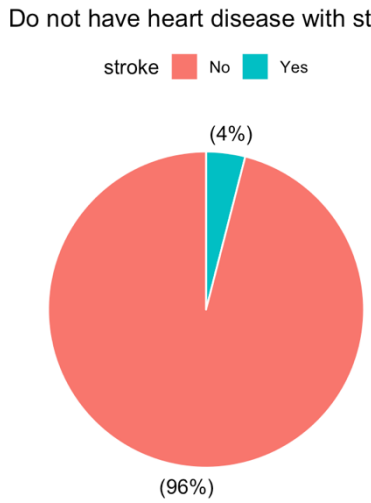
Insight #3: The number of patients who never smoke is a tallest columns. But the total of formerly smoke and smokes is reasonable enough to conclude that smoking raises the risk of stroke.

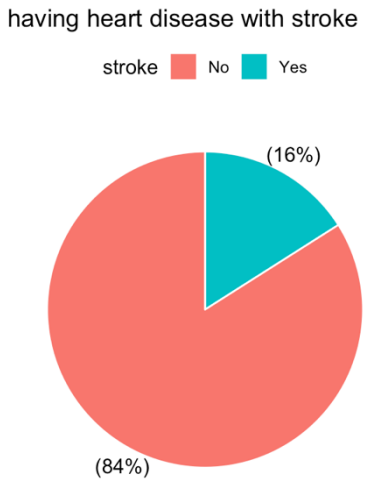
"Unknown" column is high which indicates that it takes a significant amount of data.

d) Heart disease.

We know that "Atrial fibrillation and other heart diseases can cause blood clots that lead to stroke." , so to verify the information, we visualize the stroke distribution in two group of patients, who have heart problems and do not .

stroke	heart_disease 0	1	Sum
0	4496	203	4699
1	169	40	209
Sum	4665	243	4908



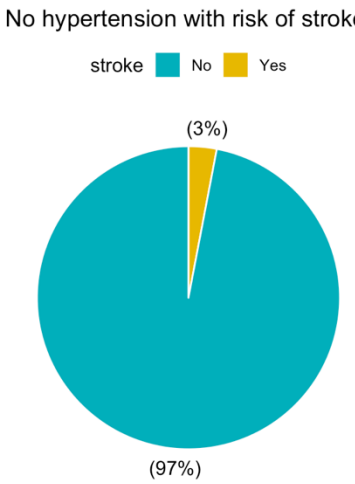


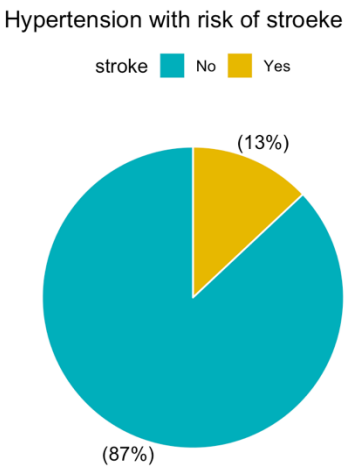
Insight #4: While the percentage of getting stroke for people having healthy heart is small, the percentage for people having heart disease is bigger than and siginificant enough. Therfore, patients who have heart disease have a higher chance to get strokes than patients who have a healthy heart.

e) Hypertention

We found the similar trend in hypertension that patients have hypertension will have higher risk of stroke than people do not have hypertension.

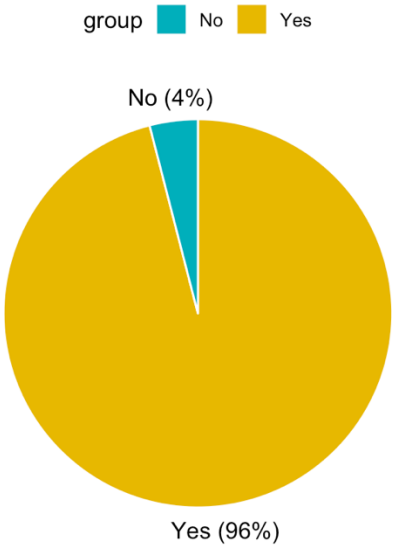
stroke	hypertension 0	1	Sum
0	4308	391	4699
1	149	60	209
Sum	4457	451	4908





Insight #5: Hypertension has significant effect on blood-preasure then it raises the chance of having stroke.

f) Patients suffering stroke before

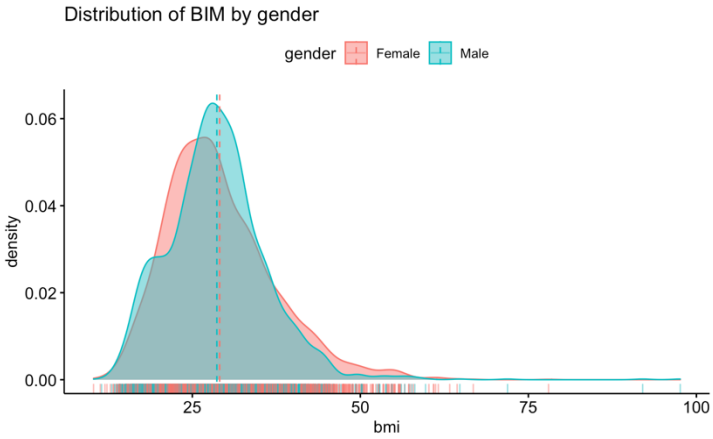


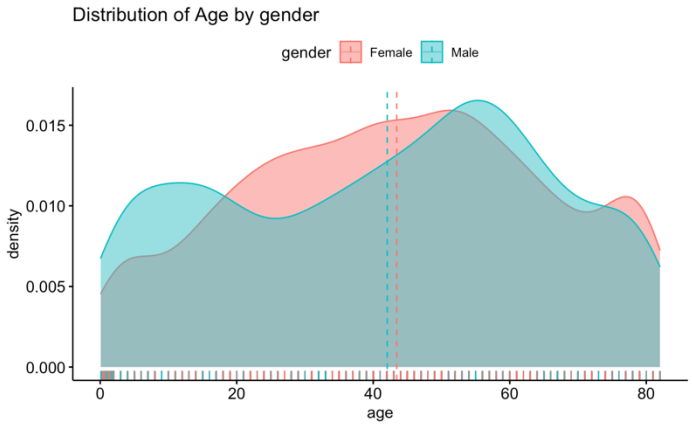
We can see the imbalance in the data distribution which is approximately 95% of patients not suffering strokes in the past. which implies any dump model should randomly predictions of stroke could reach accuracy of 95%. Therefore, in the building model, I pay more attention in the sensitivity, predict the true positive to evaluate the model.

2. Numerical attributes.

a) BIM and Age

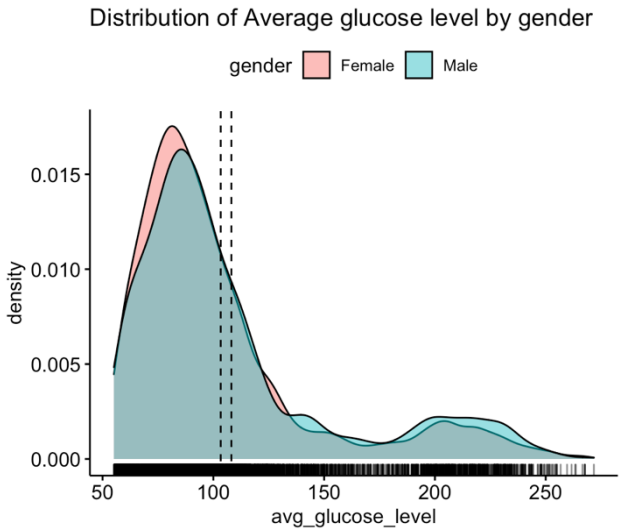
Both atributes have important effect on the risk of stroke. Let's see their distibutions to find who are more likley to getting stroke.





Bmi and age both have normal distribution. While bmi is highly skewed toward to left side and average is around 30, Age is toward to a right side, and risk of stroke is high for elderly people and mid age adults for both genders, even though male has average's age is younger than of female.

b) Average gluscose level

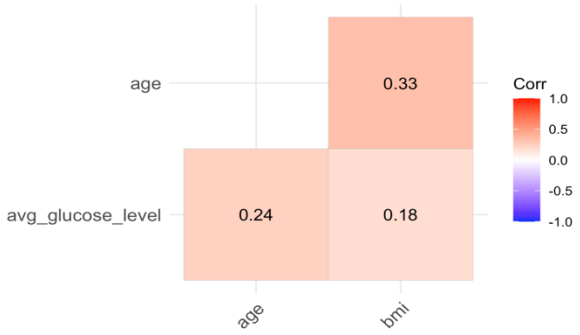


Glucose level distribution is skewed towards left and it can be seen people with regular glucose levels, around 90 to 110 mg/dL

c) Correlation for numerical variable.

The correlation matrix on the numerical attributes to see their relationships

It seemed like Age were positively correlated with BMI and avg_glucose level, though the association was not strong.



III. MODELING

The problem of our data set is imbalance in the distribution of response variable (stroke) as I mentioned in the visualization part. We will always have nearly 95% of accuracy for any models but very low correct rate of predicting true value (since patients having strokes takes only 5% in the dataset). Therefore, to address this problem, I will pick a optimal cutoff for class to get the maximun sensitivity + specify.

Also, because my problem is health diagnosis, I interest in make a correct predict a patients who have disease, then we more care about the sensitive rate to evaluate the efficiency of model.

Data slips into training data (70%) and test data (30%) and does not inclue id column.

Predited variables: patients' gender, age, hypertension, heart_disease, ever_married, work_type, residence_type, bmi, avg_glucose_level.

Response variable: stroke.

1. Logistic Regression

Since the response variable is binary values, we are use the logistic regression. I will fit the model with all the features of patients (remove only id column).

```
Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -6.908045   1.095480  -6.306 2.86e-10 ***
genderMale                   0.053145   0.183033   0.290 0.771543
age                          0.070800   0.007611   9.302  < 2e-16 ***
hypertensionYes              0.453022   0.211083   2.146 0.031859 *
heart_diseaseYes             0.339750   0.242728   1.400 0.161599
ever_marriedYes              0.090933   0.315027   0.289 0.772847
work_typeGovt_job           -1.050027   1.167253  -0.900 0.368348
work_typeNever_worked      -10.049802 417.585143  -0.024 0.980800
work_typePrivate            -0.831386   1.147615  -0.724 0.468792
work_typeSelf-employed      -1.249721   1.173602  -1.065 0.286940
Residence_typeUrban          0.010753   0.179415   0.060 0.952209
avg_glucose_level            0.005527   0.001547   3.573 0.000353 ***
bmi                         -0.001385   0.014764  -0.094 0.925256
smoking_statusnever smoked  -0.287882   0.221751  -1.298 0.194211
smoking_statussmokes         0.272602   0.264617   1.030 0.302928
smoking_statusUnknown       -0.369465   0.296937  -1.244 0.213405
```

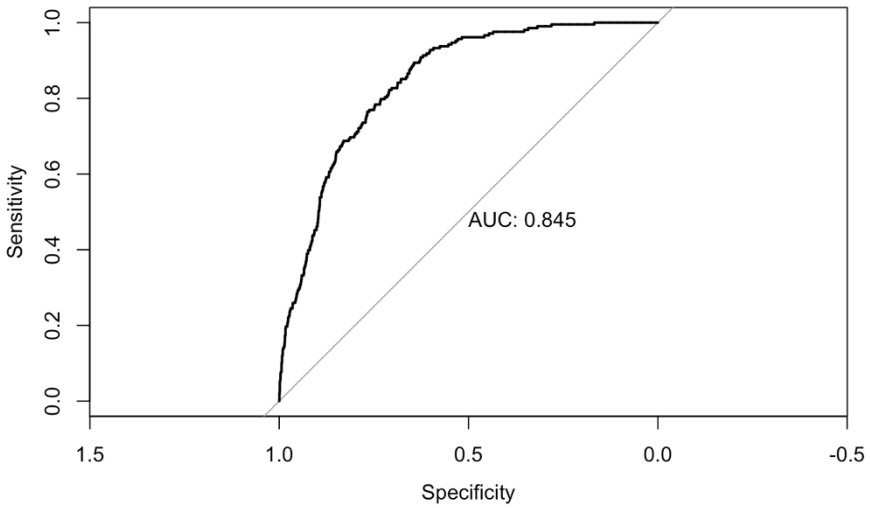
From the summary table, based on p_value < 0.5, the atributes effecting on stroke is age, hypertension, and avg_glucose_level. Besides, stroke have positive relationship with risk factors: age, heart disease, hypertension, avg_glucose_level, smoking_statussmoke, which means that if the patients have these conditions they will rasise their risk of stroke.

At the threshold 0.5, the result of prediction on the test data is

Accuracy	Sensitivity	Specificity
0.95922460	0.0161293	1.00000000

The accurate rate and specificity is high over 95% as we predict from the visualization of stroke distributiom, but the sensitivity is very low.

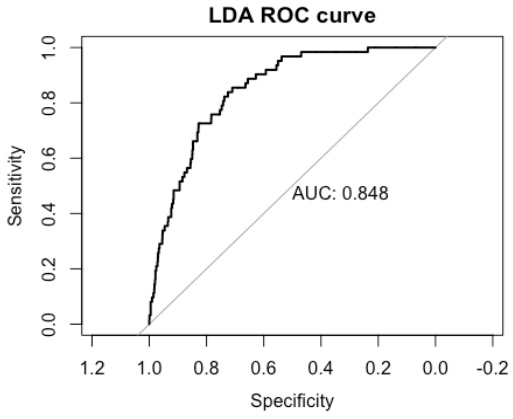
The optimal threshold that maximaze the sensitivity is 0.02969906 , which means the predicted probability of stroke is greater than 0.02969906, the predicted class for the observation will be “Stroke”. Besides the AUC is high enough to conclude that the model performs well at distinguish between positive and negative classes.

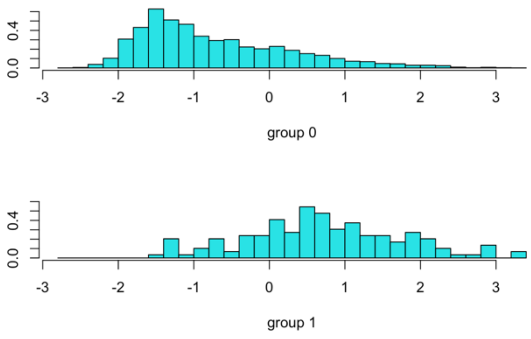


Accuracy	Sensitivity	Specificity	AUC
0.6838235	0.8709677	0.6757322	0.845

2. Linear Discriminant Analysis (LDA)

Simiarly to Logistic Regression, LDA performs well at distinguish between positive and negative classes with the 84.8% of AUC rate.



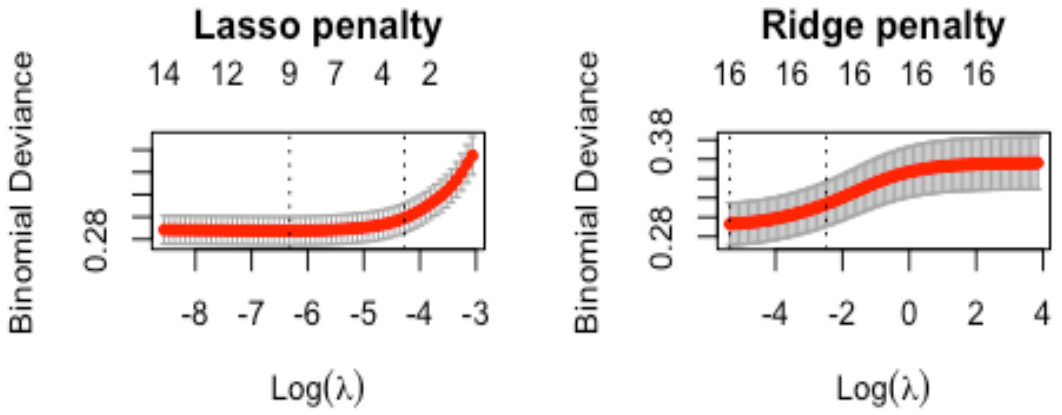


To handle the imbalance in the distribution of response variables, I change the cutoff where the sensitivity + Specificity is maximum.

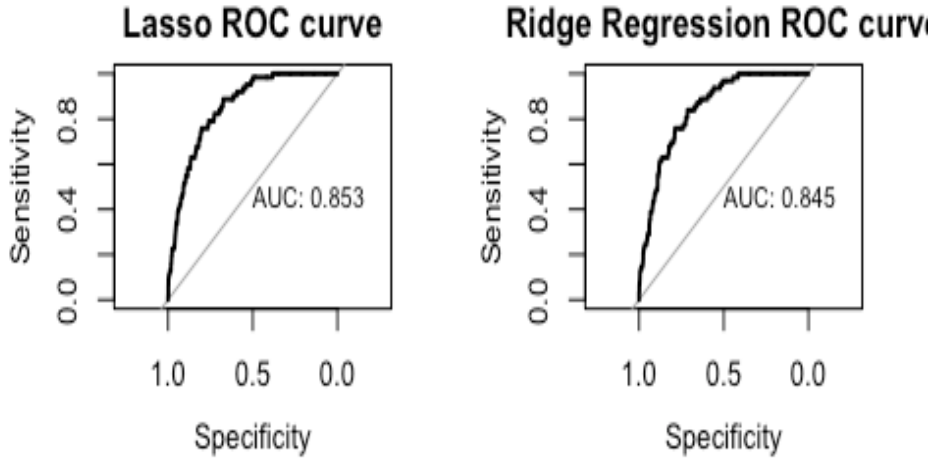
Method	Accuracy	Sensitivity	Specificity
LDA (threshold=0.0252)	0.7159091	0.8548387	0.7099024
LDA (a threshold =0.5)	0.95454545	0.0806456	0.99232915

3. Penalized logistic regression: Lasso method vs Ridge Regression

I specify a constant lambda to adjust the amount of the coefficient shrinkage that minimize the cross-validation prediction error rate.



The plot displays the cross-validation error according to the log of lambda. The left dashed vertical line indicates that the log of the optimal value of lambdas. For Lasso method, the optimal lambda is 0.00179601and 0.00466068 for ridge regression. From both plots, I found that Lasso performs better in reducing prediction error than Ridge.

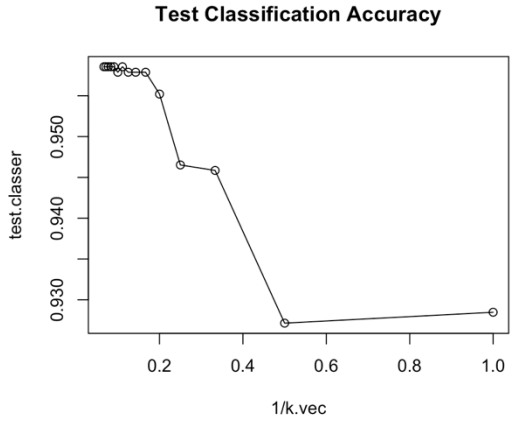


I also find the optimal threshold for lasso is 0.03277425 and for ridge is 0.03982708 to improve their sensitive rates.

Method	Accuracy	Sensitivity	Specificity	AU
Lasso	0.6838235	0.8870968	0.6750349	0.8
Ridge Regression	0.7192513	0.8387097	0.7140865	0.8

Even though both methods give us similar results and both perform well at prediction, Lasso gives a higher Sensitivitive rate and performs better in reducing prediction error.

4. K-Nearest Neighbor Regression



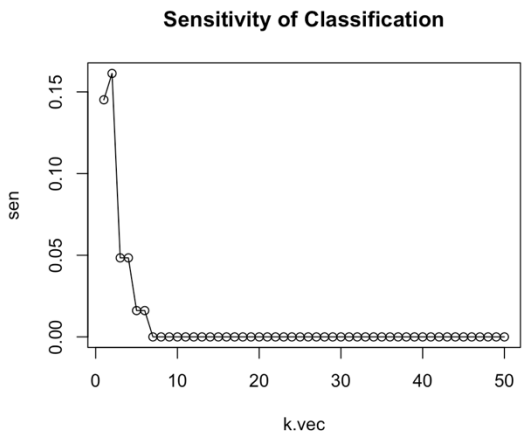
KNN classifies the data point on how its neighbor is classified; therefore, in the imbalance dataset, it will result into undesirable performance.

Furthermore, I found that the optimal K = 9,

```
 Accuracy Sensitivity Specificity
0.9578877   0.0000000   0.9993026
```

As predictions, KNN model predict well on negative values but wrong on positive values due to th imbalance of data set. Also, I calculate the Cost of Misclassification when I use KNN methods.

k	Accuracy	Sensitivity	Specificity	Cost
1	0.9284759	0.1451613	0.9623431	2.602173
2	0.9224599	0.1290323	0.9567643	2.656139
3	0.9458556	0.0483871	0.9846583	2.870180
4	0.9445187	0.0483871	0.9832636	2.871575
5	0.9552139	0.0161290	0.9958159	2.955797
6	0.9545455	0.0161290	0.9951185	2.956494
7	0.9578877	0.0000000	0.9993026	3.000697
8	0.9578877	0.0000000	0.9993026	3.000697
9	0.9585561	0.0000000	1.0000000	3.000000
10	0.9585561	0.0000000	1.0000000	3.000000



The optimal K =9 at the maximum accuracy but from the cost of misclassification and the plot of sensitivity vs K, I realize that I can get the maximum sensitive rate at k=1 but the value of sensitive rate is not significant, only 1.45%. When k >10, I get 0% of sensitivity. For the imbalance dataset, KNN is absolutely not a choice, unless we should handle the imbalance dataset by oversampling or undersampling before using KNN to fit the dataset.

IV. COMPARISON AND CONCLUSION.

1. Insights from visualization.

- It seemed like Age were positively correlated with BMI and avg_glucose level, though the association was not strong.
- Older patient was more likely to suffer a stroke than a younger patient.
- Higher BMI does not increase the stroke risk.
- Higher proportion of patients who suffered from hypertension or heart disease experienced a stroke, all else being equal.
- Regardless of patient's gender, and where they stayed, they have the same likelihood to experience stroke

1. **Comparison of modeling.**
   Results.

Method	Accuracy	Sensitivity	Specificity	AUC
Logistic Regression (with best threshold)	0.6838235	0.8709677	0.6757322	0.845
Logistic Regression (with a threshold =0.5)	0.95922460	0.0161293	1.00000000	
LDA (with a best threshold)	0.7159091	0.8548387	0.7099024	0.848
LDA (a threshold =0.5)	0.95454545	0.0806456	0.99232915	
Lasso	0.6838235	0.8870968	0.6750349	0.853
Ridge Regression	0.7192513	0.8387097	0.7140865	0.845
KNN	0.957887	0.0000	0.9993062	

Comment: To improve sensitivity, we need to change a threshold for predicted class. All methods have similar results, after use the optimal threshold, the sensitivity is improved, and other rates are high enough. The method gives the highest rate in sensitivity and AUC rate is Lasso but Logistic Regression is good too. Besides, due to the imbalance of the data set, then the KNN does not perform well, which predicts 0.0% correct true value, not good for health prediction.

Future direction: To have a better result in prediction, the imbalance of data set needs to be handled before modeling. Even though, I changed the threshold, the sensitivity was improved a lot and the accuracy was still significant enough but there are other ways to get higher accuracy and sensitive rate, especially they will work well with KNN methods. The recommended methods that make the data balance are oversampling or under sampling.

Undersampling: "This method works with majority class. It reduces the number of observations from majority class to make the data set balanced. This method is best to use when the data set is huge and reducing the number of training samples helps to improve run time and storage troubles." (Manish)

Oversampling: "This method works with minority class. It replicates the observations from minority class to balance the data." (Manish)

References

Topics, Health. "Stroke | CVA | Cerebrovascular Accident | Medlineplus". *Medlineplus.Gov*, 2021, https://medlineplus.gov/stroke.html.

"Imbalanced Classification Problems In R". *Analytics Vidhya*, 2021, https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/.

CODE

```
library(naniar)
library(ggplot2)
library(ggpubr)
library(ggcorrplot)
library(bestglm)
library(pROC) #for ROC curve
library(caret)
library(glmnet)
library(MASS)
library(class)
library(knitr)

## Read data into R
set.seed(88)
Stroke_data<-data.frame(read.csv("/Users/nhivutu/Desktop/Math
448/Rproject/healthcare-dataset-stroke-data.csv"))
head(Stroke_data)
summary(Stroke_data)

####################
## Cleaning data.##
####################
# Change to appropriate format
Stroke_data$bmi <- as.numeric(Stroke_data$bmi)
Stroke_data <- replace_with_na(data = Stroke_data, replace =
list(gender=c("Other")))
```

```
# Handling missing data
sum(is.na(Stroke_data))

#remove all missing value.
data_clean<-na.omit(Stroke_data)

#remove id attribute.
data_clean<- subset(data_clean, select = -id )
summary(data_clean)

####################
## Visualization ##
####################


#### Looking at the attributes of data

attach(data_clean)
# check unique values of categorical values by using "unique()"
unique(gender)
unique(ever_married)
unique(work_type)
unique(Residence_type)
unique(smoking_status)

##  For classification variables.
table(gender)
df   <-   data.frame(group   =   c("Male",   "Female"),value   =
c(2011,2897))
labs <- paste0(df$group, " (", df$value,")")
ggpie(df,  "value",  label  =  labs,fill  =  "group",  color  =
"white",palette = c("#00AFBB", "#E7B800"))

ggplot(data_clean,
       aes(x = Residence_type, fill = Residence_type)) +
  geom_bar() +
  labs(title = "Effect of Residence type on getting stroke?") +
  theme(legend.position = "none")

ggplot(data_clean,
       aes(x = ever_married, fill = ever_married)) +
  geom_bar() +
  labs(title = "Effect mariage on getting stroke?") +
  theme(legend.position = "none")

ggplot(data_clean,
       aes(x = work_type, fill = work_type)) +
  geom_bar() +
```

```
  labs(title = "Work Type distribution") +
  theme(legend.position = "none")

addmargins(table( stroke,heart_disease))

slices<-c(4496,169)
pct <- round(slices/sum(slices)*100)
df <- data.frame(stroke = c("No", "Yes"),value = pct)
labs <- paste0( " (", df$value,"%",")")
ggpie(df, "value", label = labs,title="Do not have heart disease
with stroke",fill = "stroke", color = "white")

slices<-c(203,40)
pct <- round(slices/sum(slices)*100)
df <- data.frame(stroke = c("No", "Yes"),value = pct)
labs <- paste0( " (", df$value,"%",")")
ggpie(df, "value", label = labs,title="having heart disease with
stroke",fill = "stroke", color = "white")

addmargins(table(hypertension, stroke))

slices<-c(149,60)
pct <- round(slices/sum(slices)*100)
df <- data.frame(group = c("No", "Yes"),value = pct)
labs <- paste0( " (", df$value,"%",")")
ggpie(df, "value", label = labs,title=" Hypertension with
experience of stroke",fill = "group", color = "white",palette =
c("#00AFBB", "#E7B800"))

slices<-c(4308,391)
pct <- round(slices/sum(slices)*100)
df <- data.frame(group = c("No", "Yes"),value = pct)
labs <- paste0( " (", df$value,"%",")")
ggpie(df, "value", label = labs,title="Hypertension with no
stroke",fill = "group", color = "white",palette = c("#00AFBB",
"#E7B800"))

table(stroke)
slices <- c(4699,209)
pct <- round(slices/sum(slices)*100)
df <- data.frame(group = c("Yes", "No"),value = pct)
labs <- paste0(df$group, " (", df$value,"%",")")
ggpie(df, "value", label = labs,fill = "group", color =
"white",palette = c("#00AFBB", "#E7B800"))

## For numberical values
ggdensity(data_clean, x = "bmi",
          add = "mean", rug = TRUE,
          color = "gender", fill = "gender",
```

```
          main="Distribution of BMI by gender")
ggdensity(data_clean, x = "age",
          add = "mean", rug = TRUE,
          color = "gender", fill = "gender",
          main="Distribution of Age by gender")
ggdensity(data_clean, x = "avg_glucose_level",
          add = "mean", rug = TRUE,
          fill = "gender",
          main="Distribution of Average glucose level by gender")

#Correlation for numerical variable.

df <-subset(data_clean, select = age,bmi,avg_glucose_level )
r <- cor(df, use="complete.obs")
round(r,2)
ggcorrplot(r,
           hc.order = TRUE,
           type = "lower",
           lab = TRUE)
#############
## MODELING##
#############

## Slip data into traing set and test set (70-30 split )
set.seed(1)
subset=sample(c(TRUE ,FALSE), nrow(data_clean ), replace=TRUE,
prob=c(0.7, 0.3))
train= data_clean[subset,]
test=data_clean[!subset,]

#create matrix for training set and test set
train.X<-model.matrix(stroke~.,data=train)[,-1]
train.Y<-train$stroke
test.X<-model.matrix(stroke~.,data=test)[,-1]
test.Y<-test$stroke

## LOGISTIC REGRESSION .
set.seed(2)
fit.all=glm(stroke~.,data=train,family=binomial)
summary(fit.all)
probs_predict=predict(fit.all,newdata=test, type="response")
roc.glm = roc(test.Y ~ probs_predict, plot = TRUE, print.auc =
TRUE)
c.glm=coords(roc.glm, "best", ret = "threshold")
pred.glm <- ifelse(probs_predict > c.glm$threshold,1,0)


##  Accurate rate, Sensitivity, Specificity table.
tab.glm=table(pred.glm, test.Y)
```

```
train_con_mat = confusionMatrix(tab.glm, positive = "1")
c(train_con_mat$overall["Accuracy"],
  train_con_mat$byClass["Sensitivity"],
  train_con_mat$byClass["Specificity"])

# Accurate rate table with a threshold =0.5
pred.glm2 <- ifelse(probs_predict > 0.5,1,0)
tab.glm2=table(pred.glm2, test.Y)
accurat.glm2 = confusionMatrix(tab.glm2, positive = "1")
c(accurat.glm2$overall["Accuracy"],
  accurat.glm2$byClass["Sensitivity"],
  accurat.glm2$byClass["Specificity"])

## LINEAR DISCRIMINANT ANALYSIS (LDA)
set.seed(2)
model.lda=lda(stroke~.,data = train)
pred.lda<-predict(model.lda, newdata=test)

plot(model.lda)
# with a threshold =0.5
tab.lda=table(pred.lda$class,test.Y)
Accurate.lda= confusionMatrix(tab.lda, positive = "1")
c(Accurate.lda$overall["Accuracy"],
  Accurate.lda$byClass["Sensitivity"],
  Accurate.lda$byClass["Specificity"])


# ROC curve and am optimal threshold for LDA.
roc.lda= roc(test.Y ~ pred.lda$posterior[, 2], plot = TRUE,
print.auc = TRUE,main = "LDA ROC curve")
c.lda=coords(roc.lda, "best", ret = "threshold")
pred.lda.adj <- ifelse(pred.lda$posterior[, 2] >c.lda$threshold,
1, 0)
pred.lda2=pred.lda.adj


#Accuracy, Sensitivity, Specificity table.
tab.lda2=table(pred.lda2,test.Y)
Accurate.lda2= confusionMatrix(tab.lda2, positive = "1")
c(Accurate.lda2$overall["Accuracy"],
  Accurate.lda2$byClass["Sensitivity"],
  Accurate.lda2$byClass["Specificity"])


## LASSO METHOD VS RIDGE REGRESSION.
set.seed(23)
# Find the best lambda using cross-validation
set.seed(123)
```

```
cv.lasso <- cv.glmnet(train.X, train.Y, alpha = 1, family =
"binomial")
cv.ridge <- cv.glmnet(train.X, train.Y, alpha = 0, family =
"binomial")

##Tuning parameter selection plot
par(mfrow=c(2,2))
plot(cv.lasso, main = "Lasso penalty\n")
plot(cv.ridge, main = "Ridge penalty\n")

# Fit the final model on the training data
model.lasso <- glmnet(train.X, train.Y, alpha = 1,   family =
"binomial",
                      lambda = cv.lasso$lambda.min)
model.ridge<- glmnet(train.X, train.Y, alpha = 0,   family =
"binomial",
                     lambda = cv.ridge$lambda.min)

# Make prediction in test data.
prob.lasso <-model.lasso %>% predict(newx = test.X, type =
"response" )
prob.ridge<- model.ridge %>% predict(newx = test.X, type =
"response" )
set.seed(24)
#Find the best threshold for both models
par(mfrow=c(2,2))
roc.lasso= roc(test$stroke ~ prob.lasso, plot = TRUE, print.auc
= TRUE,main = "Lasso ROC curve")
roc.ridge=roc(test$stroke ~ prob.ridge, plot = TRUE, print.auc =
TRUE,main = "Ridge Regression ROC curve")

# Comparison 2 thresholds.
c.lasso=coords(roc.lasso, "best", ret = "threshold")
c.ridge=coords(roc.ridge, "best", ret = "threshold")

# Make predictions as class.
pred.lasso<-ifelse(prob.lasso > c.lasso$threshold,1,0)
pred.ridge<-ifelse(prob.ridge > c.ridge$threshold,1,0)

##  Accurate rate, Sensitivity, Specificity table.
tab.lasso=table(pred.lasso, test$stroke)
tab.ridge=table(pred.ridge, test$stroke)

## For lasso method
Accurate.lasso= confusionMatrix(tab.lasso, positive = "1")
c(Accurate.lasso$overall["Accuracy"],
  Accurate.lasso$byClass["Sensitivity"],
  Accurate.lasso$byClass["Specificity"])
```

```
## For ridge regression method
Accurate.ridge= confusionMatrix(tab.ridge, positive = "1")
c(Accurate.ridge$overall["Accuracy"],
  Accurate.ridge$byClass["Sensitivity"],
  Accurate.ridge$byClass["Specificity"])


## KNN
#Using CV to find best K to get the maximum accuracy
k.vec=1:15
test.classer=rep(0,length(k.vec))
for (i in 1:length(k.vec))
{
  k=k.vec[i]
  knn.pred=knn(train.X,test.X,train.Y,k=k)
  test.classer[i]=mean(test.Y==knn.pred)
}
plot(1/k.vec,test.classer,type="o",main="Test     Classification
Accuracy")
best.k=which.max(test.classer)

#model with the best k.
knn.pred=knn(train.X,test.X,train.Y,k=best.k)
mean(test.Y==knn.pred) # accurate rate

#model with k=10
knn.pred1=knn(train.X,test.X,train.Y,k=10)
mean(test.Y==knn.pred1) # accurate rate

#model with k=15
knn.pred2=knn(train.X,test.X,train.Y,k=10)
mean(test.Y==knn.pred2) # accurate rate


#Accuracy, Sensitivity, Specificity table.
tab.knn=table(knn.pred,test.Y)
Accurate.knn= confusionMatrix(tab.knn, positive = "1")
c(Accurate.knn$overall["Accuracy"],
  Accurate.knn$byClass["Sensitivity"],
  Accurate.knn$byClass["Specificity"])

# Cost of Misclassification

acc <- c()
sen <- c()
spc <- c()
k.vec=1:50
for (i in 1:50) {
  k=k.vec[i]
```

```
  knn.pred=knn(train.X,test.X,train.Y,k=k)
  acc                                                              <-
c(acc,length(which(test.Y==knn.pred)==TRUE)/length(test.Y))
  sen <- c(sen,length(which((test.Y==knn.pred) & (test.Y==1))) /
length(which(test.Y==1)))
  spc <- c(spc,length(which((test.Y==knn.pred) & (test.Y==0))) /
length(which(test.Y==0)))
}
costdf                                                             <-
data.frame(k=1:50,Accuracy=acc,Sensitivity=sen,Specificity=spc)
cost=3*(1-costdf$Sensitivity)+(1-costdf$Specificity)
costdf <- cbind(costdf,"Cost"=cost)

kable(costdf[c(1:15,seq(20,50,by=5)),],row.names=FALSE)
plot(k.vec,sen,type="o",main="Sensitivity of Classification")
plot(k.vec,spc,type="o",main="Specificity of Classification")
plot(k.vec,spc,type="o",main="Accuracy of Classification")
```