Stat 419 Group Project

Molly Behrends, Payton Bokowy, Freeman Chen, Marvin Lim

12/15/2021

# Introduction

## Dataset Information

The dataset for this project was obtained data about strokes from kaggle.com which was collected from the World Health Organization. The dataset contains information about each person’s age, heart disease, hypertension, marital status, BMI, and their smoking status to list a few of the variables.

This dataset was chosen because strokes are the second leading cause of death in the world. It would be helpful to be able to know what features can possibly lead to a stroke, so we are aware and can reduce those factors.

## Reads in the data

data = read.csv("healthcare-dataset-stroke-data.csv") # reads data file  
#data = subset(data, select = -c(id)) # gets rid of the id  
  
# changes the predictors to factors  
data$gender = as.factor(data$gender)  
data$hypertension = as.factor(data$hypertension)  
data$heart\_disease = as.factor(data$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)  
data$bmi = as.numeric(data$bmi)

## Warning: NAs introduced by coercion

data = na.omit(data)  
str(data)

## 'data.frame': 4908 obs. of 11 variables:  
## $ gender : Factor w/ 2 levels "Female","Male": 2 2 1 1 2 2 1 1 1 1 ...  
## $ age : num 67 80 49 79 81 74 69 78 81 61 ...  
## $ hypertension : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 1 2 1 ...  
## $ heart\_disease : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 1 1 1 2 ...  
## $ ever\_married : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 1 2 2 2 ...  
## $ work\_type : Factor w/ 5 levels "children","Govt\_job",..: 4 4 4 5 4 4 4 4 4 2 ...  
## $ Residence\_type : Factor w/ 2 levels "Rural","Urban": 2 1 2 1 2 1 2 2 1 1 ...  
## $ avg\_glucose\_level: num 229 106 171 174 186 ...  
## $ bmi : num 36.6 32.5 34.4 24 29 27.4 22.8 24.2 29.7 36.8 ...  
## $ smoking\_status : Factor w/ 4 levels "formerly smoked",..: 1 2 3 2 1 2 2 4 2 3 ...  
## $ stroke : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:201] 2 9 14 20 28 30 44 47 51 52 ...  
## ..- attr(\*, "names")= chr [1:201] "2" "9" "14" "20" ...

prop.table(table(data$stroke))

##   
## 0 1   
## 0.95741646 0.04258354

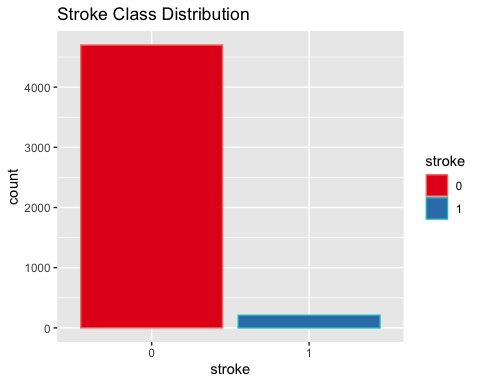
## Splits the data into a testing and training set

set.seed(1)  
n = dim(data)[1]  
  
dt = sort(sample(n, n\*.7))  
training = data[dt,]  
testing = data[-dt,]

## Processing Problems

When the data was loaded in, we noticed the data was highly unbalanced. Only about 5% of the data results in strokes, while the other 95% results in no stroke. This resulted in the models being fitted on unbalanced data. In order to combat this issue, we created synthetic data.

ggplot(data=data, aes(x=stroke, color = stroke)) + geom\_bar(aes(fill= stroke)) + ggtitle("Stroke Class Distribution") + scale\_fill\_brewer(palette = "Set1")

 From the graph we can see that the response variable are higher unbalance, which will lead to a problem that even model predict every sample as likely non-stroke, it will still likely to have a good accuracy, so in this case we are more looking for the error rates from the model instead just look at the accuracy.

# Synthetic Data

after research we found that ‘rose’ library might gave us a better chance to make the dataset look a bit ‘balance’, rose functions is design to deal with binary classification problems in the presence of imbalanced classes by generating synthetic balanced samples(<https://cran.r-project.org/web/packages/ROSE/ROSE.pdf>)

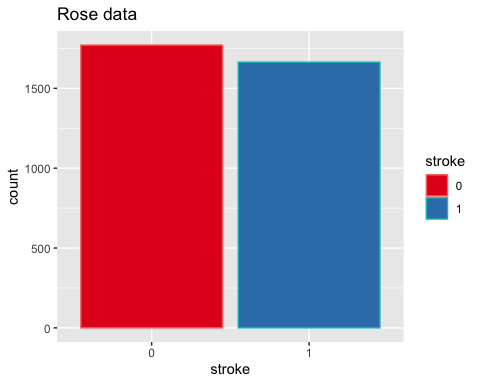
set.seed(1)  
  
syn\_train = ROSE(stroke~.,data=training)$data  
table(syn\_train$stroke)

##   
## 0 1   
## 1771 1664

syn\_test = ROSE(stroke~.,data=testing)$data  
table(syn\_test$stroke)

##   
## 0 1   
## 766 707

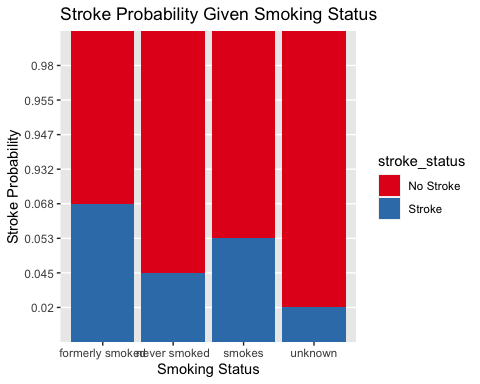
num\_train = syn\_train  
num\_test = syn\_test  
  
num\_train$gender = as.numeric(as.factor(num\_train$gender))  
num\_train$ever\_married = as.numeric(as.factor(num\_train$ever\_married))  
num\_train$work\_type = as.numeric(as.factor(num\_train$work\_type))  
num\_train$Residence\_type = as.numeric(as.factor(num\_train$Residence\_type))  
num\_train$smoking\_status = as.numeric(as.factor(num\_train$smoking\_status))  
  
num\_test$gender = as.numeric(as.factor(num\_test$gender))  
num\_test$ever\_married = as.numeric(as.factor(num\_test$ever\_married))  
num\_test$work\_type = as.numeric(as.factor(num\_test$work\_type))  
num\_test$Residence\_type = as.numeric(as.factor(num\_test$Residence\_type))  
num\_test$smoking\_status = as.numeric(as.factor(num\_test$smoking\_status))  
  
ggplot(data=syn\_train, aes(x=stroke, color = stroke)) + geom\_bar(aes(fill= stroke)) + ggtitle("Rose data") + scale\_fill\_brewer(palette = "Set1")



## Research Questions

### Will smoking status of an individual increase the chance of having a stroke?

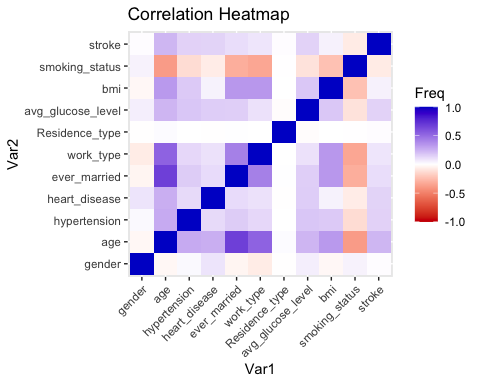
n = dim(data)[1]  
past\_smoked\_data = data[data$smoking\_status == "formerly smoked", ]  
past\_smoked\_prob = dim(past\_smoked\_data[past\_smoked\_data$stroke == 1, ])[1]/  
 dim(past\_smoked\_data)[1]  
past\_smoked\_no\_stroke = dim(past\_smoked\_data[past\_smoked\_data$stroke == 0, ])[1]/ dim(past\_smoked\_data)[1]  
  
  
never\_smoked\_data = data[data$smoking\_status == "never smoked", ]  
never\_smoked\_prob = dim(never\_smoked\_data[never\_smoked\_data$stroke == 1, ])[1]/  
 dim(never\_smoked\_data)[1]  
never\_smoked\_no\_stroke = dim(never\_smoked\_data[never\_smoked\_data$stroke == 0, ])[1]/ dim(never\_smoked\_data)[1]  
  
smokes\_data = data[data$smoking\_status == "smokes", ]  
smokes\_prob = dim(smokes\_data[smokes\_data$stroke == 1, ])[1]/  
 dim(smokes\_data)[1]  
smokes\_no\_stroke = dim(smokes\_data[smokes\_data$stroke == 0, ])[1]/  
 dim(smokes\_data)[1]  
  
u\_smoke\_data = data[data$smoking\_status == "Unknown", ]  
u\_smoke\_prob = dim(u\_smoke\_data[u\_smoke\_data$stroke == 1, ])[1]/  
 dim(u\_smoke\_data)[1]  
u\_smoke\_no\_stroke = dim(u\_smoke\_data[u\_smoke\_data$stroke == 0, ])[1]/  
 dim(u\_smoke\_data)[1]  
  
smoking\_status = c("formerly smoked", "never smoked", "smokes", "unknown",  
 "formerly smoked", "never smoked", "smokes", "unknown")  
stroke\_status = c("Stroke", "Stroke", "Stroke", "Stroke", "No Stroke",   
 "No Stroke", "No Stroke", "No Stroke")  
smoking\_prob = c(round(past\_smoked\_prob, 3), round(never\_smoked\_prob, 3),  
 round(smokes\_prob, 3), round(u\_smoke\_prob, 3),  
 round(past\_smoked\_no\_stroke, 3),   
 round(never\_smoked\_no\_stroke, 3), round(smokes\_no\_stroke, 3),  
 round(u\_smoke\_no\_stroke, 3))  
  
smoking\_data = as.data.frame(cbind(smoking\_status, stroke\_status,  
 smoking\_prob))  
  
ggplot(smoking\_data, aes(fill=stroke\_status, y=smoking\_prob,  
 x=smoking\_status)) +   
 geom\_bar(position = position\_stack(reverse = TRUE), stat="identity") +  
 xlab("Smoking Status") + ylab("Stroke Probability") +   
 ggtitle("Stroke Probability Given Smoking Status") +  
 scale\_fill\_brewer(palette = "Set1")



### What variables best classify if a patient is likely to have a stroke?

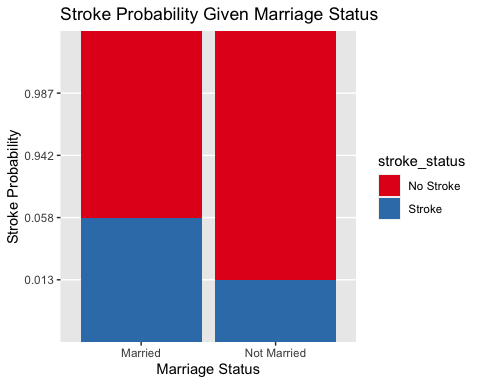
## Correlation Visualization

cor\_data = data  
  
cor\_data$gender = as.numeric(cor\_data$gender)  
cor\_data$hypertension = as.numeric(cor\_data$hypertension)  
cor\_data$heart\_disease = as.numeric(cor\_data$heart\_disease)  
cor\_data$ever\_married = as.numeric(cor\_data$ever\_married)  
cor\_data$work\_type = as.numeric(cor\_data$work\_type)  
cor\_data$Residence\_type = as.numeric(cor\_data$Residence\_type)  
cor\_data$smoking\_status = as.numeric(cor\_data$smoking\_status)  
cor\_data$stroke = as.numeric(data$stroke)  
cors = cor(cor\_data)  
cor\_df = data.frame(as.table(cors))  
  
ggplot(data=cor\_df, mapping=aes(Var1, Var2))+  
 geom\_tile(mapping = aes(fill = Freq)) +  
 scale\_fill\_gradient2(low = "red3", high = "blue3", mid = "white",   
 midpoint = 0, limit = c(-1,1)) + ggtitle("Correlation Heatmap") + scale\_x\_discrete(guide = guide\_axis(angle=45))



### Does marriage status affect the likelihood of having a stroke?

married\_data = data[data$ever\_married == "Yes", ]  
married\_stroke\_prob = dim(married\_data[married\_data$stroke == 1, ])[1]/ dim(married\_data)[1]  
married\_no\_stroke = dim(married\_data[married\_data$stroke == 0, ])[1]/ dim(married\_data)[1]  
  
not\_married\_data = data[data$ever\_married == "No", ]  
not\_married\_stroke\_prob = dim(not\_married\_data[not\_married\_data$stroke == 1, ])[1]/ dim(not\_married\_data)[1]  
not\_married\_no\_stroke = dim(not\_married\_data[not\_married\_data$stroke == 0, ])[1]/ dim(not\_married\_data)[1]  
  
marriage\_status = c("Married", "Not Married", "Married", "Not Married")  
stroke\_status = c("Stroke", "Stroke", "No Stroke", "No Stroke")  
marriage\_prob = c(round(married\_stroke\_prob, 3),   
 round(not\_married\_stroke\_prob, 3),   
 round(married\_no\_stroke, 3),   
 round(not\_married\_no\_stroke, 3))  
  
marriage\_data = as.data.frame(cbind(marriage\_status, stroke\_status,  
 marriage\_prob))  
  
ggplot(marriage\_data, aes(fill=stroke\_status, y=marriage\_prob,  
 x=marriage\_status)) +   
 geom\_bar(position = position\_stack(reverse = TRUE), stat="identity") +  
 xlab("Marriage Status") + ylab("Stroke Probability") +   
 ggtitle("Stroke Probability Given Marriage Status") +  
 scale\_fill\_brewer(palette = "Set1")

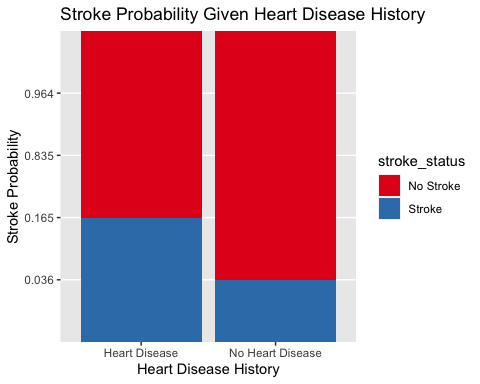


### Do people with heart disease have a higher risk of having a stroke?

no\_heart\_data = data[data$heart\_disease == 0, ]  
no\_heart\_prob = dim(no\_heart\_data[no\_heart\_data$stroke == 1, ])[1]/ dim(no\_heart\_data)[1]  
  
no\_heart\_no\_stroke = dim(no\_heart\_data[no\_heart\_data$stroke == 0, ])[1]/ dim(no\_heart\_data)[1]  
  
heart\_data = data[data$heart\_disease == 1, ]  
heart\_stroke\_prob = dim(heart\_data[heart\_data$stroke == 1, ])[1]/ dim(heart\_data)[1]  
  
heart\_no\_stroke\_prob = dim(heart\_data[heart\_data$stroke == 0, ])[1]/ dim(heart\_data)[1]  
  
heart\_status = c("No Heart Disease", "Heart Disease", "No Heart Disease", "Heart Disease")  
stroke\_status = c("Stroke", "Stroke", "No Stroke", "No Stroke")  
  
heart\_prob = c(round(no\_heart\_prob, 3), round(heart\_stroke\_prob, 3), round(no\_heart\_no\_stroke, 3), round(heart\_no\_stroke\_prob, 3))  
  
heart\_data = as.data.frame(cbind(heart\_status, stroke\_status, heart\_prob))  
heart\_data

## heart\_status stroke\_status heart\_prob  
## 1 No Heart Disease Stroke 0.036  
## 2 Heart Disease Stroke 0.165  
## 3 No Heart Disease No Stroke 0.964  
## 4 Heart Disease No Stroke 0.835

ggplot(heart\_data, aes(fill=stroke\_status, y=heart\_prob, x=heart\_status)) +   
 geom\_bar(position = position\_stack(reverse = TRUE), stat="identity") +   
 xlab("Heart Disease History") + ylab("Stroke Probability") +   
 ggtitle("Stroke Probability Given Heart Disease History") +  
 scale\_fill\_brewer(palette = "Set1")



### What classification model is best at predicting having stroke class?

## Logistic Regression with Unbalanced Data

# performs logistic regression on the stroke data  
full\_logistic = glm(stroke~., data = training, family = "binomial")  
summary(full\_logistic)

##   
## Call:  
## glm(formula = stroke ~ ., family = "binomial", data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.1853 -0.2798 -0.1350 -0.0701 3.3995   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -6.924004 1.096037 -6.317 2.66e-10 \*\*\*  
## genderMale -0.217819 0.192400 -1.132 0.257586   
## age 0.080518 0.008055 9.996 < 2e-16 \*\*\*  
## hypertension1 0.503586 0.212268 2.372 0.017672 \*   
## heart\_disease1 0.290192 0.265386 1.093 0.274185   
## ever\_marriedYes -0.224076 0.304718 -0.735 0.462122   
## work\_typeGovt\_job -1.219773 1.172320 -1.040 0.298118   
## work\_typeNever\_worked -10.268746 400.708503 -0.026 0.979555   
## work\_typePrivate -1.324364 1.158821 -1.143 0.253099   
## work\_typeSelf-employed -1.710292 1.184446 -1.444 0.148751   
## Residence\_typeUrban -0.036733 0.183037 -0.201 0.840943   
## avg\_glucose\_level 0.005245 0.001586 3.307 0.000943 \*\*\*  
## bmi 0.004218 0.014271 0.296 0.767556   
## smoking\_statusnever smoked -0.114802 0.228160 -0.503 0.614849   
## smoking\_statussmokes 0.248684 0.283563 0.877 0.380487   
## smoking\_statusUnknown -0.412705 0.316603 -1.304 0.192389   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1182.86 on 3434 degrees of freedom  
## Residual deviance: 912.51 on 3419 degrees of freedom  
## AIC: 944.51  
##   
## Number of Fisher Scoring iterations: 14

# predicts on the testing data  
probs = predict(full\_logistic, type = "response", testing)   
  
n = dim(testing)[1]  
t = 0.5  
pred.label = c()  
pred.label = rep(1, n)  
pred.label[probs>t] = 0  
  
table(predicted = pred.label, truth = testing$stroke) # confusion matrix

## truth  
## predicted 0 1  
## 0 0 2  
## 1 1406 65

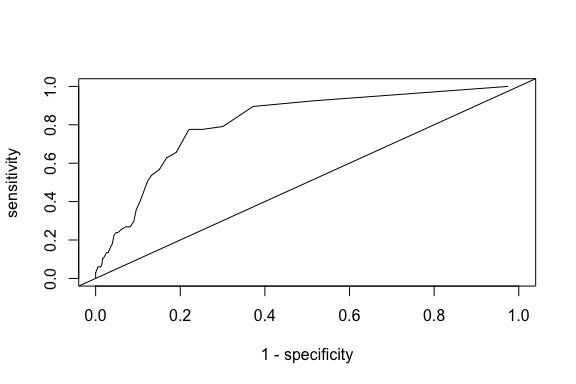
log1\_error = (1406 + 2)/ n  
log1\_error

## [1] 0.9558724

log1\_acc = (0+65) /n  
log1\_acc

## [1] 0.04412763

# plots the ROC Curve  
  
  
  
  
tseq = seq(0.001, 0.999, length.out = 100)  
sensitivity = c(); specificity = c()  
  
for (j in 1:length(tseq)){  
t = tseq[j]  
  
pred.label[probs>t] = 1  
pred.label[probs < t] = 0  
  
p.ind = which(testing$stroke == 1)  
sensitivity[j] = mean(pred.label[p.ind] == testing$stroke[p.ind])  
  
n.ind = which(testing$stroke == 0)   
specificity[j] = mean(pred.label[n.ind] == testing$stroke[n.ind])  
}  
  
plot(1 - specificity, sensitivity, type = "l", xlim = c(0, 1), ylim = c(0, 1))  
abline(a = 0, b = 1)



## Logistic Regression with Synthetic Data

full\_logistic = glm(stroke~., data = syn\_train, family = "binomial")  
  
probs = predict(full\_logistic, type = "response", syn\_test)  
n = dim(syn\_test)[1]  
t = 0.5  
pred.label = c()  
pred.label = rep(0, n)  
pred.label[probs>t] = 1  
table(predicted = pred.label, truth = syn\_test$stroke)

## truth  
## predicted 0 1  
## 0 557 178  
## 1 209 529

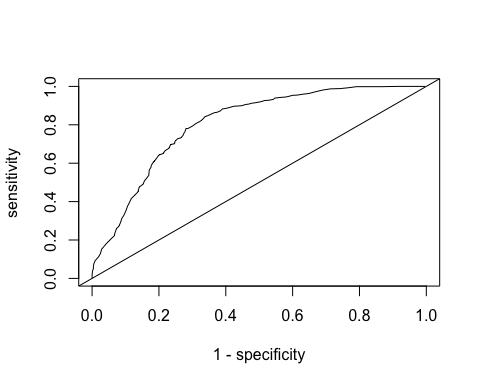
log\_TPM = (205 + 165) / n  
log\_TPM

## [1] 0.2511881

log\_acc = (562 + 541) / n  
log\_acc

## [1] 0.7488119

# plots the ROC Curve  
tseq = seq(0.001, 0.999, length.out = 100)  
sensitivity = c(); specificity = c()  
  
for (j in 1:length(tseq)){  
t = tseq[j]  
  
pred.label[probs>t] = 1  
pred.label[probs < t] = 0  
  
p.ind = which(syn\_test$stroke == 1)  
sensitivity[j] = mean(pred.label[p.ind] == syn\_test$stroke[p.ind])  
  
n.ind = which(syn\_test$stroke == 0)   
specificity[j] = mean(pred.label[n.ind] == syn\_test$stroke[n.ind])  
}  
  
plot(1 - specificity, sensitivity, type = "l", xlim = c(0, 1), ylim = c(0, 1))  
abline(a = 0, b = 1)



## Linear SVM

# performs linear support vector machine   
# svm.rad = tune(svm, stroke~., data = syn\_train, kernel = "linear",   
# ranges = list(cost = c(0.1, 1, 10, 1000)))  
  
# svm.best = svm.rad$best.model  
  
# best model: cost = 1  
library(ROCR)  
  
# predicts on the testing set  
svm.fit <- svm(stroke~., data = num\_train, kernel = "linear",   
 cost = 1, gamma = 0.01, degree = 1,probability=TRUE)  
  
pred=predict(svm.fit,num\_test[, !names(num\_test) %in% c("stroke")], probability=TRUE)  
table(prediction = pred, truth = num\_test$stroke) # confusion matrix

## truth  
## prediction 0 1  
## 0 565 180  
## 1 201 527

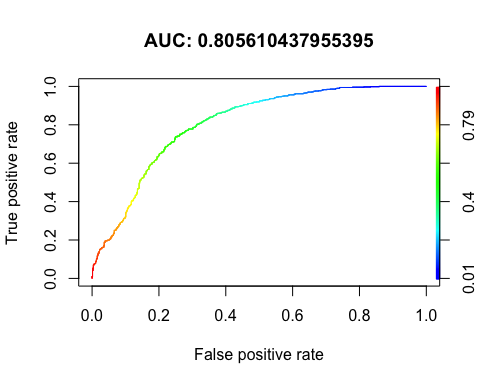
svm\_error = (201 + 181) / dim(num\_test)[1]  
svm\_error

## [1] 0.2593347

svm\_acc = (565 + 526) / dim(num\_test)[1]  
svm\_acc

## [1] 0.7406653

pred.prob = attr(pred, "probabilities")  
pred.to.roc = pred.prob[, 2]  
  
predct.rocr <- prediction(as.numeric(pred.to.roc), num\_test$stroke )  
perf.rocr<-performance(predct.rocr, measure = "auc", x.measure = "cutoff")  
perf.tpr.rocr<-performance(predct.rocr, "tpr","fpr")  
plot(perf.tpr.rocr, colorize=T,main=paste("AUC:",(perf.rocr@y.values)))



## Linear Discriminant Analysis

set.seed(123)  
lda.fit = lda(stroke~., data = num\_train, probability=TRUE) # performs LDA classification  
  
lda.pred = predict(lda.fit, num\_test) # predicts on the test set  
  
table(prediction = lda.pred$class, truth = num\_test$stroke) # confusion matrix

## truth  
## prediction 0 1  
## 0 547 165  
## 1 219 542

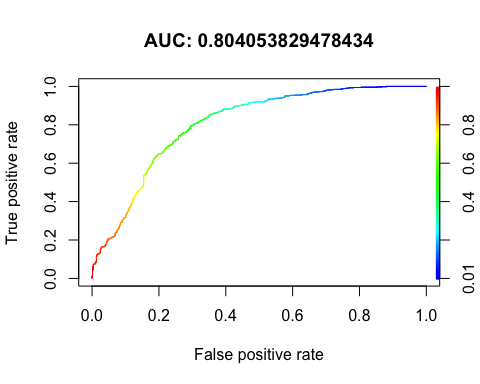
lda\_error = (219 + 165) / dim(syn\_test)[1]  
lda\_error

## [1] 0.2606925

lda\_acc = (547 + 542) / dim(syn\_test)[1]  
lda\_acc

## [1] 0.7393075

pred=predict(lda.fit,num\_test[, !names(num\_test) %in% c("stroke")], probability=TRUE)  
  
pred.prob = pred$posterior  
  
pred.to.roc = pred.prob[, 2]  
  
predct.rocr <- prediction(as.numeric(pred.to.roc), num\_test$stroke )  
perf.rocr<-performance(predct.rocr, measure = "auc", x.measure = "cutoff")  
perf.tpr.rocr<-performance(predct.rocr, "tpr","fpr")  
plot(perf.tpr.rocr, colorize=T,main=paste("AUC:",(perf.rocr@y.values)))



## KNN

set.seed(1)  
  
knn5 = knn(num\_train[, -11], num\_test[, -11], syn\_train$stroke, k=5)  
table(syn\_test$stroke, knn5)

## knn5  
## 0 1  
## 0 530 236  
## 1 180 527

TPM = (171+229) / dim(syn\_test)[1]  
TPM

## [1] 0.2715547

knn5\_acc = (538+535) / dim(syn\_test)[1]  
knn5\_acc

## [1] 0.7284453

knn10 = knn(num\_train[, -11], num\_test[, -11], syn\_train$stroke, k=10)  
table(syn\_test$stroke, knn10)

## knn10  
## 0 1  
## 0 522 244  
## 1 169 538

TPM = (165+239) / dim(syn\_test)[1]  
TPM

## [1] 0.2742702

knn10\_acc = (519+543) / dim(syn\_test)[1]  
knn10\_acc

## [1] 0.7209776

knn50 = knn(num\_train[, -11], num\_test[, -11], syn\_train$stroke, k=50)  
table(syn\_test$stroke, knn50)

## knn50  
## 0 1  
## 0 533 233  
## 1 148 559

TPM = (139+231) / dim(syn\_test)[1]  
TPM

## [1] 0.2511881

knn50\_acc = (538+565) / dim(syn\_test)[1]  
knn50\_acc

## [1] 0.7488119

## PCA&kmean

library(devtools)

## Loading required package: usethis

#install\_github("vqv/ggbiplot")  
require(tidyverse)

## Loading required package: tidyverse

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ tibble 3.1.4 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 2.0.1 ✓ forcats 0.5.1  
## ✓ purrr 0.3.4

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::select() masks MASS::select()

require(ggbiplot)

## Loading required package: ggbiplot

## Loading required package: plyr

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:purrr':  
##   
## compact

## Loading required package: scales

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

## Loading required package: grid

require(ggthemes)  
set.seed(123)  
data <- num\_train[, !names(syn\_train) %in% c("stroke")]  
str(data)

## 'data.frame': 3435 obs. of 10 variables:  
## $ gender : num 2 2 1 1 2 2 1 2 2 2 ...  
## $ age : num 0.515 44.085 36.951 28.701 49.877 ...  
## $ hypertension : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...  
## $ heart\_disease : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ ever\_married : num 1 2 2 1 2 1 1 2 2 2 ...  
## $ work\_type : num 1 5 4 1 5 4 1 5 5 4 ...  
## $ Residence\_type : num 1 2 2 1 1 2 2 1 1 2 ...  
## $ avg\_glucose\_level: num 77.7 121.1 99.7 111.2 88.1 ...  
## $ bmi : num 17.8 25.3 31.5 25 34.3 ...  
## $ smoking\_status : num 4 3 4 4 2 1 2 2 3 2 ...

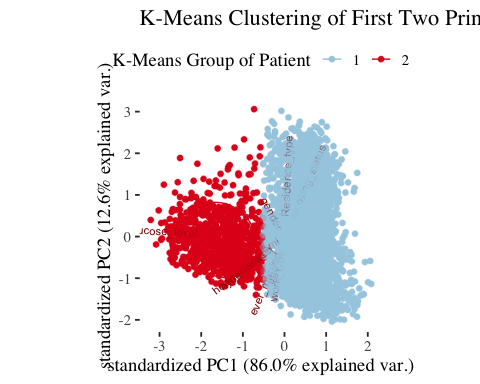
data$hypertension <- as.numeric(data$hypertension)  
data$heart\_disease <- as.numeric(data$heart\_disease)  
df.pca <- prcomp(data)  
pca\_2 <- df.pca$x %>%  
 as.tibble %>%  
 select(PC1, PC2)

## Warning: `as.tibble()` was deprecated in tibble 2.0.0.  
## Please use `as\_tibble()` instead.  
## The signature and semantics have changed, see `?as\_tibble`.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

pca\_kmeans <- pca\_2 %>%  
 kmeans(centers = 2)  
pca\_kmeans$betweenss/pca\_kmeans$totss##accuracy

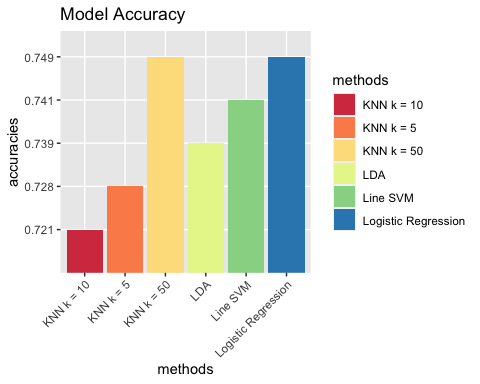
## [1] 0.6656663

ggbiplot(df.pca, groups = factor(pca\_kmeans$cluster),   
 ellipse = TRUE) +  
 theme\_tufte(base\_size = 14) +  
 geom\_point(aes(col = factor(pca\_kmeans$cluster)),   
 size = 2, alpha = 0.2) +  
 theme(legend.position = 'top') +  
 scale\_color\_manual(name = 'K-Means Group of Patient',  
 values = c('#a6cee3', '#e31a1c')) +  
 ggtitle('K-Means Clustering of First Two Principal Components')



## Results

accuracies = c(round(log\_acc, 3), round(svm\_acc, 3), round(lda\_acc, 3), round(knn5\_acc, 3), round(knn10\_acc, 3), round(knn50\_acc, 3))  
methods = c("Logistic Regression", "Line SVM", "LDA", "KNN k = 5", "KNN k = 10", "KNN k = 50")  
  
accuracy = as.data.frame(cbind(accuracies, methods))  
  
ggplot(data=accuracy, aes(x=methods, y=accuracies)) + geom\_bar(stat="identity", aes(fill = methods)) + scale\_x\_discrete(guide = guide\_axis(angle=45))+ ggtitle("Model Accuracy") + scale\_fill\_brewer(palette = "Spectral")



## Methods

Rose data method - As we see the minority class number “1” is only about 4.25% of the total cases. As we can see from the output, most observations are a 1, which means the data is highly unbalanced. This will result in the accuracy score to be nearly 1, but the model performance will be very poor. The only model affected by this will be the logistic model . We ran the logistic model without any synthetic data first, and then ran it after. This will improve the performance of the logistic regression model. but won’t effect for linear svm, lda,knn.

logistic regression - similar to linear regression,unlike linear regression the response variables can be categorical or continuous, as the model does not strictly require continuous data. To predict group membership, linear regression uses the log odds ratio rather than probabilities and an iterative maximum likelihood method rather than a least squares to fit the final model.

Linear SVM - a classification and regression model, it can solve linear and non-linear problems by creates a line or a hyperplane which separates the data into classes LDA - LDA is a linear classifier taht using bayesian statistic KNN - a supervised learning model taht used for both classification and regression, use feature similarity to predict the cluster that the new point will fall PCA+kmean - show case of the unsupervise learning model performance.

## Conclusion

After carrying out the logistic regression, non-linear support vector machine, linear discriminant analysis, and K-nearest neighbors, pca+kmean we calculated the error rate to determine which model best predicts if someone is likely to have a stroke. The logistic regression with synthetic data had the highest error rate between all the models we created. The model had a 25.11%. The next model we implemented was the non-linear support vector machine. Using a radial kernel with degree = 1 and cost = 1000 and gamma = 0.01. This model was same at predicting the stroke status of a person. The error rate of the support vector machine was 25.9% and accuracy 74% . This was one of the better models we implemented to predict the status of a person having a stroke. Linear Discriminant Analysis had a similar error rate. The linear discriminant analysis had a 26% error rate and 74% accuracy Finally, we attempted K-nearest neighbors algorithm on the data. This method gave us around the same error rate as others which is 25%, we think the best model at predicting a person’s stroke status are linear svm and logistic regression if we get acutal balanced data because we could use pca or reduction method to make it more efficient . PCA+kmean is a simple show case of how unsupervise model performance ## Additional Analyses If time allowed, it would have been interesting to test more models and compare the accuracy of those classification models to the ones we tested. We could have implemented decision trees and random forest models.

More Dimension reduction method could have been implemented to reduce the number of prediction features and should’ve try more combination of the dataset such as PCA + kmean.

To continue our exploration with stroke data, it would be beneficial to gather real world data that is balanced between having a stroke and not having a stroke.