**Signature Assignment: Final Project Complete Research Report**

**Group Beta**

**ALY 6015 Intermediate Analytics**

**CRN: 21495**

**Group Leader: Rambhipal Payyavula**

**Group Member: Cheng Chen, Minhang Lin, Sayantan Ganguly, Zilin Wang**

**Background**

Stroke, also known as "sudden onset of encephalopathy," is a common fatal disease, but it is even more deadly among the elderly. Each year, more than 795,000 people in the United States have a stroke. Approximately 610,000 of these are first-time or new strokes. According to the data from American Heart Association, in the United States, someone has a stroke every 40 seconds, and one person dies from a stroke every 4 minutes. In addition to those data, there are also some factors that influence individual’s chance of getting a stroke, such as level of obesity, diabetes, smoking, etc. Stroke has become a concerning topic, but we do not know the clinical features that can affect it. Therefore, studies exploring the clinical features that trigger or increase chance of stroke become interest of our group.

**Questions**

1. How can we more effectively predict the odds of getting stroke based on the clinical features given?
2. How can we more effectively reduce the odds of getting stroke?
3. Does living environment have an impact on the odds of getting stroke?
4. Do different genders have statistically different odds of getting stroke?
5. Does the average glucose level have an impact on the odds of getting stroke?

**Research Methods**

1. Chi-Square Tests

Since in our dataset, there are several categorical data like work type and resident type, for our final report, we want to perform more chi-square tests to see if there are significant differences between the categories. From these tests, we can find out whether different types of living environments or working environments will affect the chance of stroke.

1. Logistic Regression

Since the interest of our study is stroke, which is a binary variable, for predicting a binary variable, we need to perform a logistic regression. From the logistic regression model, we can see how each variable affect the chance of stroke.

1. LASSO Regression

We also want to perform a LASSO regression. By comparing the LASSO model with the logistic regression model, since LASSO regression may eliminate some variables to simplify the model, we can see which variables may have the higher factor on the chance of stroke.

**Dataset Introduction**

The dataset for our study is authored by fedesoriano and it is from the Kaggle website.

This dataset contains a total of 5110 records and 12 variables. The variables are specified as follows. 1) id: Unique patient identification number. 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: patient’s body mass index. 11) smoking\_status: "formerly smoked", "never smoked", "smokes" or "Unknown". 12) stroke: 1 is the patient had a stroke or 0 if not. (Fedesoriano)

**Data Analysis**

* **Data cleaning**

We first import the dataset and view the structure of it. We find variables of integer type (e.g. "hypertension", "heart\_disease", and "stroke"), factor type (e.g. "gender", "ever\_married", "work\_type", etc.), and numeric type (e.g. "age", "avg\_glucose\_level").

Text

Description automatically generated

We then view the summary of the dataset and find that there was only one "Other" value in the "Gender" variable. Meanwhile, we find that "bmi" variable has 201 N/A values. What’s more, the dataset contains some binary and continuous variables.

A screenshot of a computer

Description automatically generated with medium confidence

From the summary, we find “gender” variable only contains one “Other” value, so we decide to remove it for simplicity.

Text

Description automatically generated with medium confidence



Then, we convert integer variables (“hypertension”, “heart\_disease”) to factor for subsequent analysis purpose.

A picture containing calendar

Description automatically generated

Since “bmi” variable is obviously a continuous value, we convert it from factor to numeric.





In summary, we count there are 201 “N/A” values in “bmi” variables.



We decide to simply remove “N/A” values.



* **Exploratory Data Analysis**
  + Age distribution by stroke

Chart, histogram

Description automatically generated

The histogram above shows age distribution by stroke status. The two vertical dashed lines in the graphs are the means of age for stroke and non-stroke people in the dataset. We can clearly see that the mean age for people who have stroke is larger than that for people who don’t have it. From this initial analysis, we can guess there is a relationship between stroke and age.

* + Average glucose level by stroke

Chart, histogram

Description automatically generated

The histogram above shows the average glucose level by stroke status. We can see both distributions are bimodal. And we can’t see any difference in distribution between stroke and non-stroke. Therefore, we decided to make a boxplot to further examine if average glucose level will affect the odd of getting stroke.

Chart, scatter chart

Description automatically generated

The boxplot above shows the stroke status vs. average glucose level. From the plot, we can see people who don’t have stroke are mostly located inside the box, while most outliers are stretching on the higher end. While people who have stroke also shows a distribution of right skewed. From this plot, we can’t determine the relationship between stroke and average glucose level yet.

* + Age versus. glucose level by stroke

Chart, scatter chart

Description automatically generated

The scatterplot above shows the age vs. average glucose level by stroke status and gender. From the plot, we can see for both female and male, people who have stroke are mostly located on the high-age, and high-glucose-level area. Another interesting thing we find from this plot is that the slope of the linear regression line for male is way larger than that for female. Therefore, we guess gender may play a role in stroke chance.

* + BMI distribution by stroke

Chart, histogram

Description automatically generated

The histogram above shows the bmi distribution of people by stroke status. The two vertical dashed lines are the mean bmi for people who have stroke and people who don’t have stroke. We can see there is only a minimum difference between the two. And, both distributions are relatively normal. Therefore, we guess bmi doesn’t play a role in stroke chance.

* **Hypothesis testing**
  + T-test for bmi vs. average people

From initial EDA, we find bmi may not play a role in stroke chance, so we decide to go deeper to test if bmi data for people who have stroke in our dataset is consistent with the average data. According to CDC, BMI of a healthy person is between 18.5 to 24.9, so we decide to use 25 as our value for testing.

We randomly selected 25 data items in the dataset to meet the requirement of less than 30 t-test data. We default the significance level α to 0.05

Text

Description automatically generated

State Hypothesis:

H0: bmi of people who have stroke is the same as average bmi.

H1: bmi of people who have stroke is different than average bmi.

From the output below, the p-value is less than α, so we reject the null hypothesis. Therefore, the bmi of people who have stroke is significantly different than the average. From this result, we can guess that bmi can still be a factor for stroke chance.

Text, letter

Description automatically generated

* Chi-Squared test
  + Resident type vs. stroke

First, create a table to show the distribution of resident types vs. stroke status.

Table

Description automatically generated

State Hypothesis:

H0: Resident type is independent upon stroke.

H1: Resident type is dependent upon stroke.

We set the significance level to 0.05.



Then, create vectors and matrix for the chi-square test.

Graphical user interface, text, application

Description automatically generated

Next, run the test and save the results.

Text

Description automatically generated



Since the p-value is greater than the significance level, we fail to reject the null hypothesis. Therefore, we have enough evidence to conclude that the resident type is independent upon stroke.

* + Work type vs. stroke
* First, create a table to show the distribution of work types vs. stroke status.

Table

Description automatically generated

We can see the samples for people who have stroke have only one observation on children work type, and no observation on never-worked, so it is insignificant to run a chi-square test on work-type vs. stroke.

* Logistic Regression

First, split the dataset into training and test set.

Graphical user interface, text, application

Description automatically generated

Then, based on previous exploratory data analysis and hypothesis testing, fit a logistic model using predictor of gender, age, hypertension, heart disease, average glucose level, and BMI.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

By looking at the z-scores of the coefficients, we can see gender, heart disease, and BMI are very insignificant in the model.

Below are the log of coefficients and coefficients of our first model.

Text, letter

Description automatically generated

Therefore, we update the model by taking out these three predictors, and only using age, hypertension, and average glucose level as predictors.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

From the z-scores of the coefficients, we can see all predictors used are significant in the model. Also, by updating the model, the AIC of the model is improved from 979.98 to 976.83.

Below are the log of coefficients and coefficients of our second model.

A picture containing text

Description automatically generated

Then, we make a confusion matrix for the model.



Text

Description automatically generated

Below is the ROC curve of our model, and we calculate the area under the curve.

Chart

Description automatically generated

Text

Description automatically generated

* Lasso Regression

Split the dataset into training and test sets.

Text, letter

Description automatically generated

Then, find the best lambda value by using cross-validation

Graphical user interface, text, application

Description automatically generated

Then, plot the lambda.

Chart, histogram

Description automatically generated

From the graph above, the two dashed lines represent the minimum lambda value and the lambda value that is one standard error away from minimum. We can see by using lambda.1se instead of lambda.min, the number of predictors used reduces from 7 to 4.

Then, compare the RMSE for both lambda values on training and test sets.

Text

Description automatically generated

From the output above, we can see for both training and test sets, the RMSE is lower when using the lambda.1se. Therefore, we decide to use lambda.1se.

Then, we make a lasso regression model for lambda.1se and see its coefficients.



Table

Description automatically generated

From the output above, we should include age, hypertension, heart disease and BMI in the model.

Then we update our logistic regression model again.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Then, we make a confusion matrix for our final model.



Text

Description automatically generated

From the output above, we can see the accuracy of our model is 0.8618, which means our model prediction can make the right call 86% of the time, and sensitivity is 0.5986, which means that among the people who have stroke, the model can only predict correctly 59.86% of the time.

Then, we see the ROC curve and calculate the area under the curve.

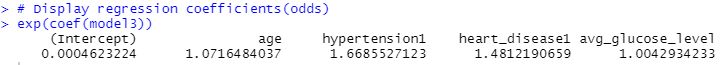
Chart

Description automatically generated

Text

Description automatically generated

Then, we see the coefficients for the model again.



Therefore, in the final model, we can see for each unit increase in age will increase the odds of getting stroke by 1.07, people who have had hypertension have about 67% more odds of having stroke than people who have not, people who have heart disease have about 48% more odds of having stroke than people who have not, and for each unit increase in average glucose level increases the odds of getting stroke by about 1.

Finally, we check if our model has multicollinearity by using VIF.



Each VIF score is about 1, which means that there are no multicollinearity issues within our model.

In our model, the predictors we used are age, hypertension, heart disease and average glucose level. Hypertension and heart disease are symptoms that are difficult to control, and age is impossible to moderate. Therefore, based our model, the only thing we can do to minimize the odds of getting stroke is to reduce the glucose level.

**Conclusion**

Starting from the exploratory data analysis, we observe that the mean age of patients having a stroke is greater than those not having one. When we segregated the dataset based on sex, we inferred that males are more prone to strokes than females. Next, we observe that the medians of BMI of both the classes of the population are almost similar which means BMI might not play an important role in our analysis. From our testing, we find out that the BMI is in fact a significant feature because its p value comes out to be less than our significance level. Next, we go on to test whether resident type is dependent upon stroke or not. From the p value statistics, we infer that the Resident level is independent of stroke. While model building, during the cross validation (using LASSO) we dig out the features which are statistically significant for our analysis. We see that the age, hypertension level, heart disease and average glucose level are significant for our research, so we go ahead for model building with them. After the model is built, we test the model and find out its accuracy to be 86.18%. However, the sensitivity of our model appears to be only around 0.6.

**Reference**

Centers for Disease Control and Prevention. (2017, September 6). *Preventing stroke deaths*. Centers for Disease Control and Prevention. Retrieved February 9, 2022, from https://www.cdc.gov/vitalsigns/stroke/

Centers for Disease Control and Prevention. (2021, August 27). *About adult BMI*. Centers for Disease Control and Prevention. Retrieved February 9, 2022, from https://www.cdc.gov/healthyweight/assessing/bmi/adult\_bmi/index.html

Fedesoriano. (2021, January 26). *Stroke prediction dataset*. Kaggle. Retrieved February 9, 2022, from https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

Virani, S., Salim, Alonso, A., Alvaro, Benjamin, Emelia J., Bittencourt, Marcio S., Callaway, Clifton W., Carson, April P., Chamberlain, Alanna M., Chang, Alexander R., Cheng, Susan. (2020, January 29). *Heart disease and stroke statistics-2020 update: A report from the American Heart Association*. Circulation. Retrieved February 9, 2022, from

https://www.ahajournals.org/doi/10.1161/CIR.0000000000000757

**Appendix: R code**

#Group Beta: Final Project: Initial Analysis Report

#install packages

install.packages**(**"dplyr"**)**

install.packages**(**"ggplot2"**)**

install.packages**(**"cowplot"**)**

install.packages**(**"plyr"**)**

install.packages**(**"gmodels"**)**

install.packages**(**"dummies"**)**

install.packages**(**"caret"**)**

install.packages**(**"glmnet"**)**

install.packages**(**"Metrics"**)**

install.packages**(**"pROC"**)**

install.packages**(**"regclass"**)**

#load library

library**(**dplyr**)**

library**(**ggplot2**)**

library**(**cowplot**)**

library**(**plyr**)**

library**(**gmodels**)**

library**(**dummies**)**

library**(**caret**)**

library**(**glmnet**)**

library**(**Metrics**)**

library**(**pROC**)**

library**(**regclass**)**

#load dataset

df **<-** read.csv**(**file.choose**()**, header **=** **TRUE**, stringsAsFactors **=** **TRUE)**

#view the structure and summary of dataset

str**(**df**)**

summary**(**df**)**

################################################################################

#data cleaning

################################################################################

#check gender summary

summary**(**df**$**gender**)**

#delete other from variable gender

df **<-** df %>%

filter**(**gender **!=** "Other"**)**

#turn variable stroke from int to factor and check

df**$**stroke **<-** as.factor**(**df**$**stroke**)**

str**(**df**$**stroke**)**

#turn variable hypertension from int to factor and check

df**$**hypertension **<-** as.factor**(**df**$**hypertension**)**

str**(**df**$**hypertension**)**

#turn variable heart disease from int to factor and check

df**$**heart\_disease**<-** as.factor**(**df**$**heart\_disease**)**

str**(**df**$**heart\_disease**)**

#turn variable bmi from factor to numeric and check

df**$**bmi **<-** as.numeric**(**as.character**(**df**$**bmi**))**

str**(**df**$**bmi**)**

#count NA in variable bmi

sum**(**is.na**(**df**$**bmi**))**

#remove NA value from variable bmi

df1 **<-** df %>%

na.omit**(**df**$**bmi**)**

################################################################################

# for numeric variable -- analyzing age, average glucose level, bmi vs. stroke

################################################################################

#calculate mean age by stroke status

mu **<-** ddply**(**df1, "stroke", summarise, grp.mean**=**mean**(**age**))**

#age distribution by stroke status with mean age

ggplot**(**df1, aes**(**x **=** age, color **=** stroke**))** **+**

geom\_histogram**(**fill **=** "darkgrey", position **=** "identity"**)** **+**

geom\_vline**(**data **=** mu, aes**(**xintercept **=** grp.mean, color **=** stroke**)**, linetype **=** "dashed"**)** **+**

theme\_bw**()**

#calculate mean average glucose level by stroke status

mu2 **<-** ddply**(**df1, "stroke", summarise, grp.mean**=**mean**(**avg\_glucose\_level**))**

#glucose distribution by stroke status

ggplot**(**df1, aes**(**x **=** avg\_glucose\_level, fill **=** stroke**))** **+**

geom\_histogram**(**position **=** "dodge"**)** **+**

theme\_bw**()**

#boxplot for average glucose level vs. stroke

ggplot**(**df1,aes**(**x**=**stroke,y**=**avg\_glucose\_level,color**=**stroke**))+**

geom\_boxplot**(**outlier.shape**=NA)+**

geom\_jitter**(**width**=**0.2,cex**=**0.6**)+**

ggtitle**(**"stroke and avg glucose level"**)+**

xlab**(**"stroke"**)+**

ylab**(**"avg glucose level" **)+**

theme\_bw**()**

#scatterplot of age vs. glucose level by stroke status

ggplot**(**df1,aes**(**x**=**age,avg\_glucose\_level,

color **=** stroke, shape **=** stroke**))+**

geom\_jitter**(**width**=**0.2, alpha **=** 0.4**)+**

geom\_smooth**(**method **=** "lm"**)+**

facet\_wrap**(~**gender**)+**

theme**(**panel.grid**=**element\_blank**()**,

panel.background**=**element\_rect**(**fill**=**'transparent', color**=**'black'**))**

#calculate mean bmi by stroke status

mu1 **<-** ddply**(**df1, "stroke", summarise, grp.mean**=**mean**(**bmi**))**

#bmi distribution by stroke status

ggplot**(**df1, aes**(**x **=** bmi, color **=** stroke**))** **+**

geom\_histogram**(**fill **=** "darkgrey", position **=** "identity"**)** **+**

geom\_vline**(**data **=** mu1, aes**(**xintercept **=** grp.mean, color **=** stroke**)**, linetype **=** "dashed"**)** **+**

theme\_bw**()**

#extract people who have stroke

df\_stroke1 **<-** df1 %>%

filter**(**stroke **==** "1"**)**

#initial randomizer

set.seed**(**123**)**

#randomly select a sample of 25 observations

sampleindex\_25 **<-** sample**(**nrow**(**df\_stroke1**)**, 25**)**

sample\_25 **<-** df\_stroke1**[**sampleindex\_25,**]**

#t test for bmi of people who have stroke vs. average (25)

#Ho: bmi of people who have stroke is the same as average bmi

#H1: bmi of people who have stroke is different than average bmi

test\_bmi\_stroke **<-** t.test**(**sample\_25**$**bmi, alternative **=** "two.side", mu **=** 25, conf.level **=** 0.99**)**

#view result of t test

test\_bmi\_stroke

################################################################################

# for factor variable -- hypothesis test of stroke vs. resident type and work type

################################################################################

#summarize resident type count by stroke status

CrossTable**(**df1**$**stroke,df1**$**Residence\_type,prop.r **=FALSE**,

prop.c **=** **FALSE**, prop.t **=** **FALSE**, prop.chisq **=** **FALSE**,

dnn **=**c**(**"stroke","resident type"**))**

# State the hypothesis

# H0: Residence type is independent upon Stroke

# H1: Residence type is dependent upon Stroke

# Set significant level

alpha\_rt**<-** 0.05

# Create a vector for each row

r1\_rt**<-** c**(**2318,2381**)**

r2\_rt**<-** c**(**100,109**)**

# State the number of rows for the matrix

rows\_rt**<-** 2

# Create a matrix from the rows

mtrx\_rt **=** matrix**(**c**(**r1\_rt,r2\_rt**)**,nrow **=** rows\_rt, byrow **=** **TRUE)**

# Name the rows and columns

rownames**(**mtrx\_rt**)** **=** c**(**"Rural","Urban"**)**

colnames**(**mtrx\_rt**)** **=** c**(**"Stroke\_0","Stroke\_1"**)**

# View the matrix

mtrx\_rt

# Run the test and save the result

result\_rt**<-** chisq.test**(**mtrx\_rt**)**

# View the test statistic and p-value

result\_rt**$**statistic #chi-square test value

result\_rt**$**p.value #chi-square p-value

result\_rt**$**parameter #degree of freedom

result\_rt

#Compare the p-value to the alpha and make decision

ifelse**(**result\_rt**$**p.value **>** alpha\_rt, "Fail to reject H0", "Reject H0"**)**

#summarize work type by stroke status in the table

CrossTable**(**df1**$**stroke,df1**$**work\_type,prop.r **=FALSE**,

prop.c **=** **FALSE**, prop.t **=** **FALSE**, prop.chisq **=** **FALSE**,

dnn **=**c**(**"stroke","work type"**))**

#Since there are columns with only 0 and 1 values, it is insignificant to perform the hypothesis test

#################################################################################

# Logistic Regression

#################################################################################

# Split the data into a train and test set

#initialize randomizer

set.seed**(**123**)**

#get 70% of random row numbers

trainIndex **<-** createDataPartition**(**df1**$**stroke,p**=**0.7,list**=FALSE)**

#get training set includes 70% of rows

train **<-** df1**[**trainIndex,**]**

#get test set excludes 70% of rows

test **<-** df1**[-**trainIndex,**]**

# Fit a logistic regression model

model1**<-**glm**(**stroke **~** gender **+** age **+** hypertension **+** heart\_disease **+** avg\_glucose\_level

**+** bmi,data**=**train,family**=**binomial**(**link**=**"logit"**))**

summary**(**model1**)**

# Interpret the GLS model

# Display regression coefficients(log-odds)

coef**(**model1**)**

# Display regression coefficients(odds)

exp**(**coef**(**model1**))**

# Update model2

model2**<-**glm**(**stroke **~** age **+** hypertension **+** avg\_glucose\_level,

data **=** train, family **=** binomial**(**link **=** "logit"**))**

summary**(**model2**)**

# Interpret the GLS model

# Display regression coefficients(log-odds)

coef**(**model2**)**

# Display regression coefficients(odds)

exp**(**coef**(**model2**))**

##################################################################

# Train set prediction

##################################################################

#prepare for making confusion matrix

pro\_train **<-** predict**(**model2, newdata**=**train, type**=**"response"**)**

pre\_class1**<-** as.factor**(**ifelse**(**pro\_train**>=**0.1, "1", "0"**))**

# Confusion matrix

confusionMatrix**(**pre\_class1, train**$**stroke**)**

# Roc curve

ROC **<-** roc**(**train**$**stroke,pro\_train**)**

plot**(**ROC, print.acu**=TRUE**, auc.polygon**=TRUE**,grid**=**c**(**0.1,0.2**)**,

grid.col**=**c**(**"Green","Red"**)**,max.auc.polygon**=TRUE**,

auc.polygon.col**=**"skyblue",print.thres**=TRUE**,

ylab**=**"Sensitivity - TP Rate", xlab**=** "Specificity - Fp Rate"**)**

# Calculate the area under the ROC curve

AUC**<-**ROC**$**auc

AUC

#################################################################################

#-Lasso Regression

#################################################################################

# Split dataset

# initialize randomizer

set.seed**(**123**)**

# get 80% of random row numbers

trainIndex1**<-**createDataPartition**(**df1**$**stroke,p**=**0.8,list **=** **FALSE)**

# get training set includes 80% of rows

train1 **<-** df1**[**trainIndex1,**]**

# get test set excludes 80% of rows

test1 **<-** df1**[-**trainIndex1,**]**

# Romove Grad.Rate from test\_x and train\_x

train1\_x **<-** model.matrix**(**stroke**~**.,train1**)[**,**-**1**]**

test1\_x **<-** model.matrix**(**stroke**~**.,test1**)[**,**-**1**]**

# Add Grad.Rate Column to test\_y and train\_y

train1\_y **<-** train1**$**stroke

test1\_y **<-** test1**$**stroke

# Find the best lamada with cross validation

set.seed**(**123**)**

# Find the best lamada with cross validation

cv.lasso **<-** cv.glmnet**(**train1\_x, train1\_y,alpha**=**1, nfolds**=**10, family**=**"binomial"**)**

#lambda min

cv.lasso**$**lambda.min

#lambda.1se

cv.lasso**$**lambda.1se

#plotting

plot**(**cv.lasso**)**

#Fitting a lasso regression model to the training set

# lasso regression coefficients for lambda min

model.lasso.min**<-**glmnet**(**x **=** train1\_x, y **=** train1\_y, alpha **=** 1,

lambda **=** cv.lasso**$**lambda.min,family**=**"binomial"**)**

# find coef of lasso model at lasso.min

coef**(**model.lasso.min**)**

# lasso regression coefficients for lambda 1se

model.lasso.1se**<-**glmnet**(**x **=** train1\_x, y **=** train1\_y, alpha **=** 1,

lambda **=** cv.lasso**$**lambda.1se,family**=**"binomial"**)**

# find coef of lasso model at lasso.1se

coef**(**model.lasso.1se**)**

#Train set prediction of lasso model by calculating RMSE

#lamda min

p1**<-** predict**(**model.lasso.min, newx **=** train1\_x**)**

train.lasso.rmse.min **<-** rmse**(**as.numeric**(**train1\_y**)**, p1**)**

train.lasso.rmse.min

#lamda 1se

p2 **<-** predict**(**model.lasso.1se, newx **=** train1\_x**)**

train.lasso.rmse.1se **<-** rmse**(**as.numeric**(**train1\_y**)**, p2**)**

train.lasso.rmse.1se

#Test set prediction of lasso model by calculating RMSE

#lamda min

p3**<-** predict**(**model.lasso.min, newx **=** test1\_x**)**

test.lasso.rmse.min **<-** rmse**(**as.numeric**(**test1\_y**)**, p3**)**

test.lasso.rmse.min

#lamda 1se

p4 **<-** predict**(**model.lasso.1se, newx **=** test1\_x**)**

test.lasso.rmse.1se **<-** rmse**(**as.numeric**(**test1\_y**)**, p4**)**

test.lasso.rmse.1se

# Update model3 - final model

model3**<-**glm**(**stroke **~** age **+** hypertension **+** heart\_disease **+** avg\_glucose\_level,

data **=** train, family **=** binomial**(**link **=** "logit"**))**

summary**(**model3**)**

# Confusion matrix

pro\_train3 **<-** predict**(**model3, newdata**=**train, type**=**"response"**)**

pre\_class3**<-** as.factor**(**ifelse**(**pro\_train3**>=**0.1, "1", "0"**))**

confusionMatrix**(**pre\_class3, train**$**stroke**)**

# Roc curve

ROC1 **<-** roc**(**train**$**stroke,pro\_train3**)**

plot**(**ROC1, print.acu**=TRUE**, auc.polygon**=TRUE**,grid**=**c**(**0.1,0.2**)**,

grid.col**=**c**(**"Green","Red"**)**,max.auc.polygon**=TRUE**,

auc.polygon.col**=**"skyblue",print.thres**=TRUE**,

ylab**=**"Sensitivity - TP Rate", xlab**=** "Specificity - Fp Rate"**)**

# Calculate the area under the ROC curve

AUC1**<-**ROC1**$**auc

AUC1

# Display regression coefficients(odds)

exp**(**coef**(**model3**))**

# Each unit increase in age increases the odds of getting stroke by 1.07

# People who have had hypertension have about 67% more odds of having stroke than people who have not

# People who have had heart disease have about 48% more odds of having stroke than people who have not

# Each unit increase in average glucose level increases the odds of getting stroke by about 1

# Check multicollinearity by using VIF

VIF**(**model3**)**