Group No: 282 HealthCare Data-Set- For Stroke Prediction

First Name	Last Name	Email (hawk.iit.edu)	Student ID
Vaibhavi	Kulkarni	vkulkarni3@hawk.iit.edu	A20452252
Ankit	Patil	apatil44@hawk.iit.edu	A20451742

Table of Contents

1. Introduction:

HealthCare is a broad term. It captures all the health, life related Care, precaution, prevention, detection, cure and all the other medical terms. We are working on one of the aspects of that healthcare, which will allow us/doctors to predict a medical condition happening in the first place. We have chosen the data to predict the heart Stroke, which will prevent the patient from suffering from heart strokes.

It will be very useful if we collect the healthcare data and predict the outcomes based on historical data and symptoms which are the commonly known for heart strokes. In order to intervene that from happening we will be predicting prior to the patient getting a stroke.

2. Data:

About our Data: We have taken our data from Kaggle:
https://www.kaggle.com/asaumya/healthcare-dataset-stroke-data

We are dealing with 2 excel files with a data size 4 MB with 11 columns and 18000 rows for test data and 43000 rows for training data.

```
> str(train_data)
'data.frame':   43400 obs. of  12 variables:
 $ id               : int  30669 30468 16523 56543 46136 32257 52800 41413 15266 28674 ...
 $ gender           : Factor w/ 3 levels "Female","Male",..: 2 2 1 1 2 1 1 1 1 1 ...
 $ age              : num  3 58 8 70 14 47 52 75 32 74 ...
 $ hypertension     : int  0 1 0 0 0 0 0 0 0 1 ...
 $ heart_disease    : int  0 0 0 0 0 0 0 1 0 0 ...
 $ ever_married     : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 2 2 2 2 2 ...
 $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 1 4 4 4 3 4 4 5 4 5 ...
 $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 1 2 2 1 1 2 2 1 1 2 ...
 $ avg_glucose_level: num  95.1 88 110.9 69 161.3 ...
 $ bmi              : num  18 39.2 17.6 35.9 19.1 50.1 17.7 27 32.3 54.6 ...
 $ smoking_status   : Factor w/ 4 levels "","formerly smoked",..: 1 3 1 2 1 1 2 3 4 3 ...
 $ stroke           : int  0 0 0 0 0 0 0 0 0 0 ...
```

<u>Our data includes</u>: Age, Gender, Marital Status, Work Type, BMI, Hypertension, Heart Diseases, Ever experienced stroke, Smoking Status, Glucose Level, Residence Type

We have numerical as well as some categorical data.

- **NUMERICAL**
- Id
- Age
- Hypertension
- Heart Diseases
- Avg Glucose Level
- BMI
- Stroke
- **CATEGORIAL**
- Gender
- Ever married
- Work Type
- Residence Type
- Smoking Status

3. Problems to be Solved

- Who can get heart Stroke (potential Patients)?
- What are symptoms of getting a heart stroke.
- To find how many people suffered from heart stroke.
- How many people may suffer from heart stroke?
- Is hypertension a main reason of heart stroke?
- Does smoking have any impact on getting a heart stroke.

4. Solutions

Our data had missing value: We solved that problem using the mean of the values

```
#Check if there are any missing values in the columns
na_count=sapply(train_data, function(y) sum(length(which(is.na(y)))))
na_count=data.frame(na_count)
na_count
```

Total Missing Value Count: **1462**

```
> na_count
                  na_count
id                       0
gender                   0
age                      0
hypertension             0
heart_disease            0
ever_married             0
work_type                0
Residence_type           0
avg_glucose_level        0
bmi                   1462
smoking_status           0
stroke                   0
```

```
#We find that there are 1462 missing values for the bmi column
#Replace those missing valyes with the mean value
train_data$bmi=ifelse(is.na(train_data$bmi),ave(train_data$bmi,FUN = function(x) mean(x,na.rm = TRUE)),train_data$bmi)
```

Solution: We can see below is the snap- shot of replaced missing value with mean values.

```
> na_count1
                   na_count1
id                         0
gender                     0
age                        0
hypertension               0
heart_disease              0
ever_married               0
work_type                  0
Residence_type             0
avg_glucose_level          0
bmi                        0
smoking_status             0
stroke                     0
```

Here, we can see that we have replaced all the missing values.

```
> str(train_data)
'data.frame':   43400 obs. of  12 variables:
 $ id               : int  30669 30468 16523 56543 46136 32257 52800 41413 15266 28674 ...
 $ gender           : Factor w/ 3 levels "Female","Male",..: 2 2 1 1 2 1 1 1 1 1 ...
 $ age              : num  3 58 8 70 14 47 52 75 32 74 ...
 $ hypertension     : int  0 1 0 0 0 0 0 0 0 1 ...
 $ heart_disease    : int  0 0 0 0 0 0 0 1 0 0 ...
 $ ever_married     : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 2 2 2 2 2 ...
 $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 1 4 4 4 3 4 4 5 4 5 ...
 $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 1 2 2 1 1 2 2 1 1 2 ...
 $ avg_glucose_level: num  95.1 88 110.9 69 161.3 ...
 $ bmi              : num  18 39.2 17.6 35.9 19.1 50.1 17.7 27 32.3 54.6 ...
 $ smoking_status   : Factor w/ 4 levels "","formerly smoked",..: 1 3 1 2 1 1 2 3 4 3 ...
 $ stroke           : int  0 0 0 0 0 0 0 0 0 0 ...
```

Now when building a good model, it is necessary to eliminate the extra field which have no impact on our data. For better designing of our models. If you are going to build predictive models, clearly indicate the dependent and independent variables.

We have eliminated a Column: ID which was of no use in prediction.

```
#Removing the ID column as we do not need the same for prediction of heart stroke
smoke_data=smoke_data[ , -which(names(smoke_data) %in% c("id"))]
head(smoke_data)
```

Eliminated:

```
> head(smoke_data)
   gender age hypertension heart_disease ever_married    work_type Residence_type avg_glucose_level  bmi  smoking_status stroke
2    Male  58           1            0          Yes      Private          Urban            87.96 39.2    never smoked      0
4  Female  70           0            0          Yes      Private          Rural            69.04 35.9 formerly smoked      0
7  Female  52           0            0          Yes      Private          Urban            77.59 17.7 formerly smoked      0
8  Female  75           0            1          Yes Self-employed          Rural           243.53 27.0    never smoked      0
9  Female  32           0            0          Yes      Private          Rural            77.67 32.3          smokes      0
10 Female  74           1            0          Yes Self-employed          Urban           205.84 54.6    never smoked      0
```

For Creating Visualization i.e For Understanding Purposes:

We have converted our stroke, hypertension, heart diseases Column from (1/0) to (Yes/No) type for better comparison.

```
#yes no conversion for visualisation
smoke_data$hypertension=as.factor(ifelse(smoke_data$hypertension==1, 'YES', 'NO'))
summary(smoke_data$hypertension)

smoke_data$heart_disease=as.factor(ifelse(smoke_data$heart_disease==1, 'YES', 'NO'))
summary(smoke_data$heart_disease)

smoke_data$stroke=as.factor(ifelse(smoke_data$stroke==1, 'YES', 'NO'))
summary(smoke_data$stroke)

head(smoke_data)

#Drawing bar plots
```

Here We can see the converted data:

```
> head(smoke_data)
   gender age hypertension heart_disease ever_married    work_type Residence_type avg_glucose_level  bmi  smoking_status stroke
2    Male  58          YES           NO          Yes      Private          Urban            87.96 39.2    never smoked     NO
4  Female  70           NO           NO          Yes      Private          Rural            69.04 35.9 formerly smoked     NO
7  Female  52           NO           NO          Yes      Private          Urban            77.59 17.7 formerly smoked     NO
8  Female  75           NO          YES          Yes Self-employed          Rural           243.53 27.0    never smoked     NO
9  Female  32           NO           NO          Yes      Private          Rural            77.67 32.3          smokes     NO
10 Female  74          YES           NO          Yes Self-employed          Urban           205.84 54.6    never smoked     NO
> str(smoke_test$pred)
```

We have Also Normalized our data to bring the data down to one equal scale:

- Dealing with Numerical Data

```
#Normalization function
norm=function(n) {
  return((n-min(n))/(max(n)-min(n)))
}

#normalize
smoke_num$age=norm(smoke_num$age)
smoke_num$avg_glucose_level = norm(smoke_num$avg_glucose_level)
smoke_num$bmi=norm(smoke_num$bmi)

head(smoke_num)
```

Here is the Output of Normalization: To scale our Data

```
> head(smoke_num)
          age avg_glucose_level        bmi
2   0.6666667         0.13959498 0.35531136
4   0.8333333         0.05943908 0.31501832
7   0.5833333         0.09566175 0.09279609
8   0.9027778         0.79867819 0.20634921
9   0.3055556         0.09600068 0.27106227
10  0.8888889         0.63900186 0.54334554
> str(smoke_test$pred)
```

- Dealing with Categorical Data:

Categorical data must be converted into n-1 dummy variables. We have thus created n-1 dummy variables for categorical data.

```
# categorial

smoke_cat = smoke_data[,-c(2,8,9)]
head(smoke_cat)

#dealing with categorical

smoke_cat=data.frame(sapply(smoke_cat,function(x) data.frame(model.matrix(~x-1,data =smoke_cat))[,-1]))
head(smoke_cat)

smoke_final = cbind(smoke_num,smoke_cat)
head(smoke_final)
```

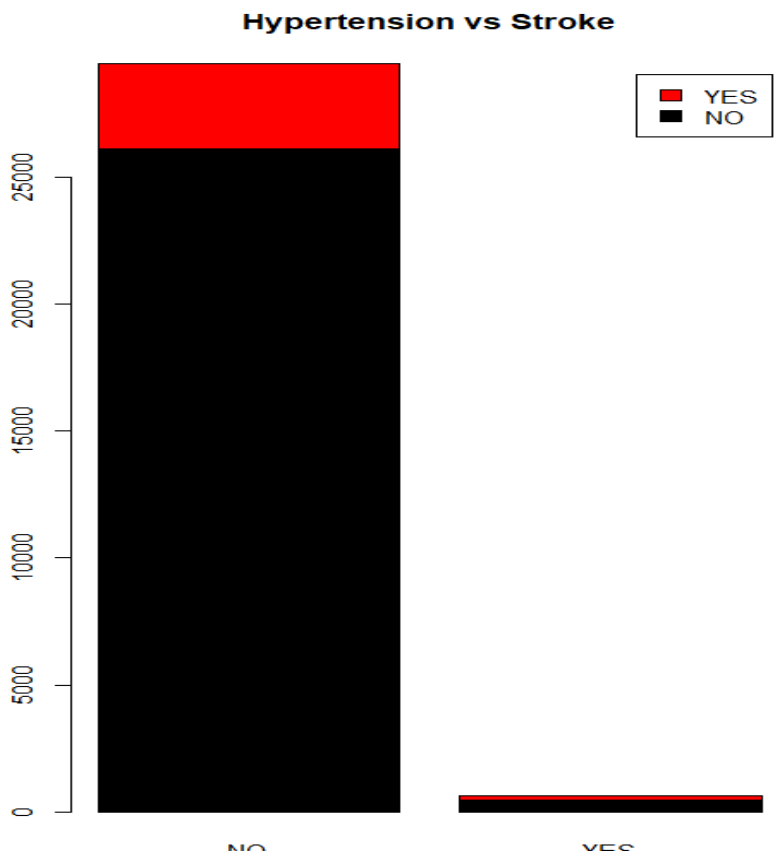
Output

```
> head(smoke_cat)
   gender.xMale gender.xOther hypertension heart_disease ever_married work_type.xGovt_job work_type.xNever_worked work_type.xPrivate
2             1            0            1             0           1                   0                       0                  1
4             0            0            0             0           1                   0                       0                  1
7             0            0            0             0           1                   0                       0                  1
8             0            0            0             1           1                   0                       0                  0
9             0            0            0             0           1                   0                       0                  1
10            0            0            1             0           1                   0                       0                  0
   work_type.xSelf.employed Residence_type smoking_status.xformerly.smoked smoking_status.xnever.smoked smoking_status.xsmokes stroke
2                         0             1                               0                           1                     0      0
4                         0             0                               1                           0                     0      0
7                         0             1                               1                           0                     0      0
8                         1             0                               0                           1                     0      0
9                         0             0                               0                           0                     1      0
10                        1             1                               0                           1                     0      0
```

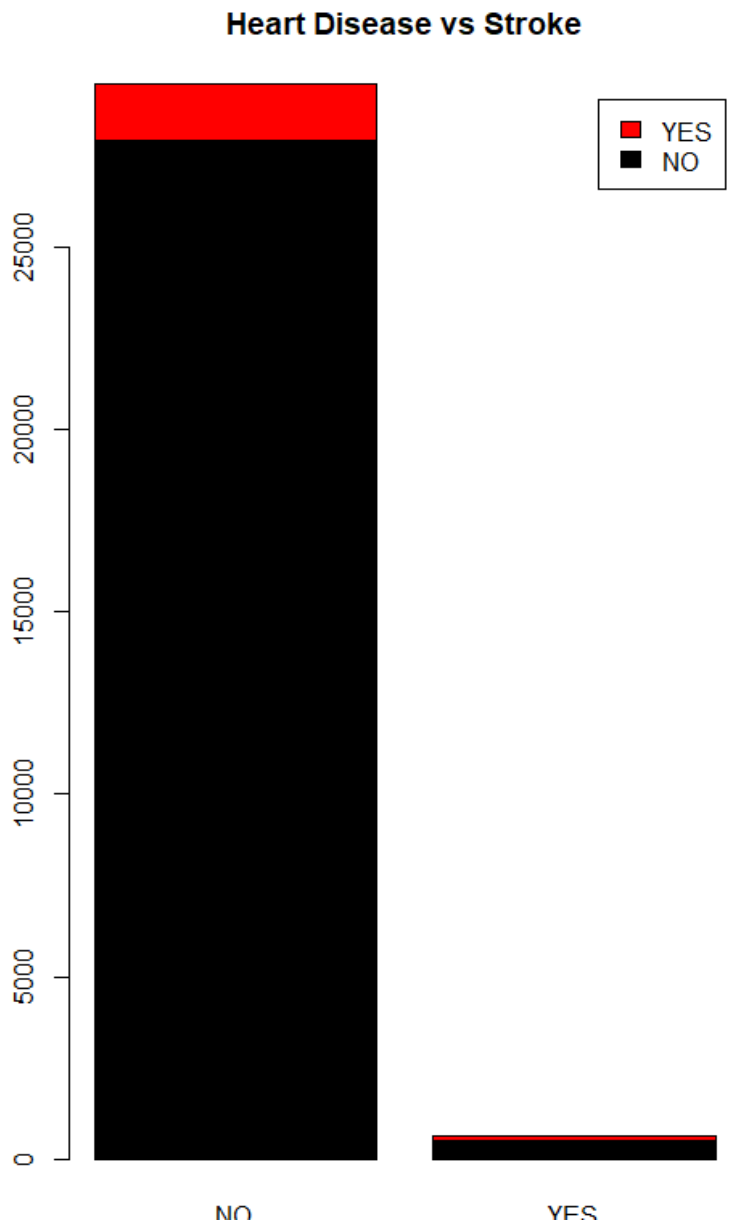
Here we have used cbind Function to bind our data again after conversion:

```
> head(smoke_final)
         age avg_glucose_level        bmi gender.xMale gender.xOther hypertension heart_disease ever_married work_type.xGovt_job work_type.xNever_worked
2  0.6666667         0.13959498 0.35531136           1            0           1            0           1                   0                       0
4  0.8333333         0.05943908 0.31501832           0            0           0            0           1                   0                       0
7  0.5833333         0.09566175 0.09279609           0            0           0            0           1                   0                       0
8  0.9027778         0.79867819 0.20634921           0            0           0            1           1                   0                       0
9  0.3055556         0.09600068 0.27106227           0            0           0            0           1                   0                       0
10 0.8888889         0.63900186 0.54334554           0            0           1            0           1                   0                       0
   work_type.xPrivate work_type.xSelf.employed Residence_type smoking_status.xformerly.smoked smoking_status.xnever.smoked smoking_status.xsmokes stroke
2                   1                       0             1                               0                           1                     0      0
4                   1                       0             0                               1                           0                     0      0
7                   1                       0             1                               1                           0                     0      0
8                   0                       1             0                               0                           1                     0      0
9                   1                       0             0                               0                           0                     1      0
10                  0                       1             1                               0                           1                     0      0
```

- We have plotted some bar graphs which tells us the impact of Hypertension on Stroke:

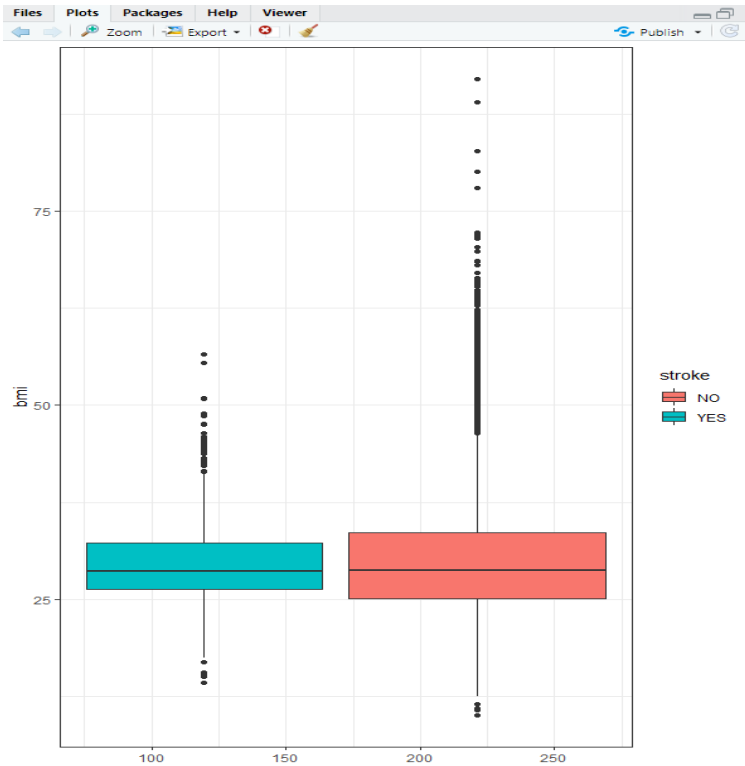


- The graph here helps us to predicts the number of patients who may suffer or may not suffer from hypertension may or may not get heart strokes.



- The graph here predicts that the number of patients who are suffering from heart problems which may result in heart stroke or may not get heart stroke even if they have heart problems.

We have performed ANOVA Hypothesis to solve our problem



- We draw box plot to show the difference between group means.
- H0: µ1 = µ2= µ3= … µk all the means are equal.
- H1: not all the means are equal.
- Hypothesis Result:

```
> summary(anovatt)

Call:
lm(formula = bmi ~ avg_glucose_level)

Residuals:
    Min      1Q  Median      3Q     Max 
-19.097  -4.774  -1.047   3.581  63.294 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      2.724e+01  1.020e-01  267.08   <2e-16 ***
avg_glucose_level 2.581e-02  8.744e-04   29.52   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.974 on 30106 degrees of freedom
Multiple R-squared:  0.02812,	Adjusted R-squared:  0.02809 
F-statistic: 871.2 on 1 and 30106 DF,  p-value: < 2.2e-16
```

Here we have consider two group mean i.e 1. BMI 2. Avg_glucose_level

We came to know that the p-value is smaller than alpha with 95% confidence level. Therefore, we reject null hypothesis and state that the means are not equal as the variance is very large and we cannot rely on the means.

Hence, we reject null hypothesis in Anova.

To Predict we are using Supervised Learning Models:

Supervised learning occurs when a system is given input and output variables with the intentions of learning how they are mapped together, or related. The goal is to produce an accurate enough mapping function that when new input is given, the algorithm can predict the output

Supervised Learning

Classification **Regression**

Both Classification and Regression uses the training data for prediction.

X= Input Variable **Y=Output Variable**

- The output Variable in classification is Categorial or discreate.
- In classification algorithm attempts to estimate the mapping function from the input variables to Categorical output variable(yes/no).
- We have used both classification Models to Build our model:

1.KNN- Classification. **2.Logistic Regression.**

When performing Classification for model we have split our data to predict the outcome. So, we have split our data into 2 type:

1.Traing Data **2. Test Data**
75% **25%**

To do this we have use Hold Out evaluation as our data was large enough.

We can use Hold-out when our data is large and so we split up our dataset into a 'train' and 'test' set. The training set is what the model is trained on, and the test set is used to see how well that model performs on unseen data. For our models we have split our data in 75% & 25% respectively.

```
set.seed(101)
sample = sample.split(smoke_final$stroke, SplitRatio = .75)
train = subset(smoke_final, sample == TRUE)
test  = subset(smoke_final, sample == FALSE)
```

Model-1 K- Nearest Neighbor Algorithm

- KNN is based on 'Feature Similarity' so we do classification using KNN Classifier.
- So, it classifies the based points on how the neighbors are classified.

How algorithm works?

- It stores all available cases and classifies new case based on similarity Measures.
- '**K**' in the KNN is parameter that refers to the number of Nearest Neighbor to include in the Majority voting Process.
- For our model we have choose **K=13**(For better accuracy).
- KNN uses Euclidean Distance: To find out the outcomes. So, it searches for result which is near to K value (in our case 13) and predicts the outcome.

```
m1<-knn(train=smoke_train, test=smoke_test, cl=smoke_train$stroke, k=13)
summary(m1)
```

Tells us the total no of data.

```
> summary(m1)
    0     1
 3795  3733
```

Confusion Matrix to tell us the accuracy of the model.

```
> confusionMatrix(table(m1 ,smoke_test$stroke))
Confusion Matrix and Statistics


m1     0    1
  0 8848   82
  1    0  103

               Accuracy : 0.9909
                 95% CI : (0.9887, 0.9928)
    No Information Rate : 0.9795
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.711

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 1.0000
            Specificity : 0.5568
         Pos Pred Value : 0.9908
         Neg Pred Value : 1.0000
             Prevalence : 0.9795
         Detection Rate : 0.9795
   Detection Prevalence : 0.9886
      Balanced Accuracy : 0.7784

       'Positive' Class : 0
```

ACCURACY: 99% with 95%CL

P value: Less than alpha also we can see the sensitivity, Specificity of the model.

- **Model-2: Logistic Regression**
  - Logistic Regression is a classification Technique
  - It is used to predict qualitative response. (to decide which category, it actually belongs to)
  - In our case, it predicts 'Strokes' depending upon the parameter given. Stroke is dependent variable/Response variable and so, independent variables such as age, gender, work type drives this dependent variable.
  - **Why are we using Logistic Regression?**
  - Because our Y variable (Stroke) is categorical variable(yes/no) type & it depends on X variables.
  - As logistic regression answers "will it happen or not"?
  - In our case will a person experience a stroke or not?
  - We have spilt our data into train and test.

```
sample = sample.split(smoke_final$stroke, SplitRatio = .75)
train = subset(smoke_final, sample == TRUE)
test  = subset(smoke_final, sample == FALSE)
```

Using glm function:

```
#logistic regression model
model <- glm (stroke ~ ., data = train, family = "binomial")

summary(model)
```

```
> model <- glm (stroke ~ ., data = train, family = "binomial")
> summary(model)

Call:
glm(formula = stroke ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6896  -0.8615  -0.2203   0.8773   2.5645

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -2.991444   0.086026 -34.774  < 2e-16 ***
age                              3.933301   0.070963  55.428  < 2e-16 ***
avg_glucose_level                0.762720   0.058328  13.076  < 2e-16 ***
bmi                             -0.300201   0.170396  -1.762 0.078105 .
gender.xMale                     0.016143   0.027503   0.587 0.557238
gender.xOther                   -1.946721   1.972916  -0.987 0.323779
hypertension                     0.464212   0.033558  13.833  < 2e-16 ***
heart_disease                    0.637136   0.040795  15.618  < 2e-16 ***
ever_married                     0.011332   0.038453   0.295 0.768224
work_type.xGovt_job             -0.076232   0.045978  -1.658 0.097313 .
work_type.xNever_worked         -0.672249   0.492428  -1.365 0.172199
work_type.xPrivate              -0.053000   0.037052  -1.430 0.152601
work_type.xSelf.employed         0.125869   0.039872   3.157 0.001595 **
Residence_type                   0.069158   0.026533   2.606 0.009149 **
smoking_status.xformerly.smoked  0.009883   0.037752   0.262 0.793497
smoking_status.xnever.smoked    -0.121908   0.036054  -3.381 0.000721 ***
smoking_status.xsmokes           0.165423   0.039928   4.143 3.43e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31302  on 22579  degrees of freedom
Residual deviance: 24354  on 22563  degrees of freedom
AIC: 24388

Number of Fisher Scoring iterations: 4
```

The ‘*’ in the model tells us which are the statistically significant variable in this data. I.e the variables that impacted and causes heart Stroke & they are as follows:

- AGE: Person’s age
- BMI
- Avg Glucose level present in the body
- Heart diseases
- Hypertension
- Work Type: Self employed
- Residence Type
- Smoking Status

- Predicting Stroke with 50%.

```
> table(test$stroke, predict1 > 0.5)

   FALSE TRUE
0   2602 1194
1    842 2890
```

- The value indicated by orange arrow is: **Actual Value**
- The value indicated by blue arrow is: **Predicted value**
- The values marked by Yellow are the values we are interested in: Correct Classification. They tell us number of patients who actually suffered, and model also says the same. And patient who did not suffered and model says the same.
- The value marks in Red are the misclassified values.
- So, as we are interested in only correct classification: 2602+2890=5492/entire data set

Confusion Matrix: To check is our prediction and actual value matching with each other or not.

```
> confusionMatrix(table(predict1 ,test$stroke))
Confusion Matrix and Statistics

predict1    0    1
        0 2602  842
        1 1194 2890

               Accuracy : 0.7295
                 95% CI : (0.7194, 0.7396)
    No Information Rate : 0.5043
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4595

 Mcnemar's Test P-Value : 7.316e-15

            Sensitivity : 0.6855
            Specificity : 0.7744
         Pos Pred Value : 0.7555
         Neg Pred Value : 0.7076
             Prevalence : 0.5043
         Detection Rate : 0.3456
   Detection Prevalence : 0.4575
      Balanced Accuracy : 0.7299

       'Positive' Class : 0
```

Accuracy: 72% when considered 50% with 95% CI

5. Experiments and Results

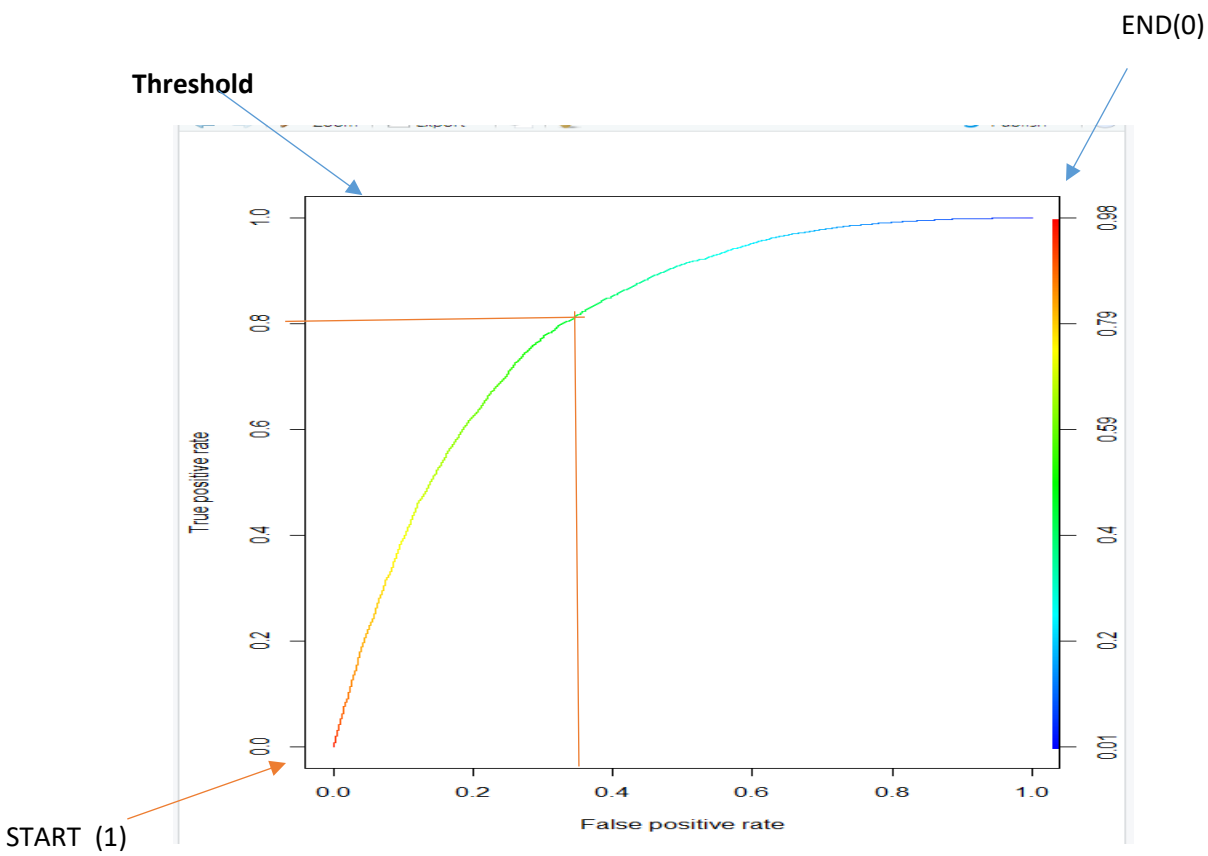
5.1. Evaluations and Results

- **ROC**: It is used to describe the sensitivity/specificity tradeoffs for a binary classifier.
- **ROC** curve plots true positive rate against false positive rate, giving a picture of the whole spectrum of such tradeoffs.

- **Why Roc?**

  ROC helps us to decide a good threshold value for our model.

```
#ROCR Curve
library(ROCR)
ROCRpred <- prediction(predict, train$stroke)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```



- Here True positive rate= Sensitivity(Portion of actual 1's caught)
- False positive Rate= 1- Specificity(Error Parameter/ Portion of 0's Predicted as 1).
- The curve gives us the threshold value.

- T=1 as threshold is 1 we are not predicting any 1's from our data.So, our sensitivity is 0.Also, our 1-specificity will be 0 for the same reason.
- When T=0: We have catched all 1's from our data.So, it will be 100%.i.e Sensitivity.Since, we have predicted everybody as 1.And so, false positive rate(error will also be 100%).
- If we consider 0.8 as our threshold that means we will have less error and we will get a good threshold for our model.

5.2. Findings

- KNN is better suited model for our data set.

6. Conclusions and Future Work

6.1. Conclusions

- We have used KNN and logistic regression model to predict heart strokes.
- The accuracy of KNN is 99% whereas, Logistic regression model shows 72% accuracy.
- Therefore, KNN model is better in predicting heart strokes.

6.2. Limitations

- There are other classifications model as well which we could have used.
- Prediction is based on only 2 models. We should try other and then decide which model is well suited to predict the heart stroke by the provided data.

6.3. Potential Improvements or Future Work

- We can analyze the data using different models such as Naïve Bayes and Trees to predict our data outcomes, also to test the accuracy of each model.
- We can consider more parameters like family history of strokes, alcohol consumption, etc. to predict heart strokes.

Code File

Project Name:

HealthCare Data Set to predict Heart Stroke.

Group No: 282

Vaibhavi Kulkarni	A20452252
Ankit Patil	A20451742

#Setting on directory path

```
setwd("D:/Data Analytics-527/project/healthcare-dataset-stroke-data/")

train_data = read.csv("train_2v.csv",header=T)
```

#Check if there are any missing values in the columns

```
na_count=sapply(train_data, function(y) sum(length(which(is.na(y)))))

na_count=data.frame(na_count)

na_count
```

#Data Preprocessing

```
#We find that there are 1462 missing values for the bmi column

#Replace those missing valyes with the mean value

train_data$bmi=ifelse(is.na(train_data$bmi),ave(train_data$bmi,FUN = function(x)
mean(x,na.rm = TRUE)),train_data$bmi)
```

#We check if all the bmi columns are replaced

```
sum(is.na(train_data))

na_count1=sapply(train_data, function(y) sum(length(which(is.na(y)))))

na_count1=data.frame(na_count1)

na_count1
```

#the count finally is 0

#Checking blank values and omitting them

```
smoke_data = train_data[ train_data$id %in% train_data$id & train_data$smoking_status != "",
]

str(smoke_data)


na.omit(smoke_data$smoking_status)
```

#Removing the ID column as we do not need the same for prediction of

```
smoke_data=smoke_data[ , -which(names(smoke_data) %in% c("id"))]

head(smoke_data)
```

#yes no conversion for data visualisation

```
smoke_data$hypertension=as.factor(ifelse(smoke_data$hypertension==1, 'YES', 'NO'))

summary(smoke_data$hypertension)


smoke_data$heart_disease=as.factor(ifelse(smoke_data$heart_disease==1, 'YES', 'NO'))

summary(smoke_data$heart_disease)
```

```
smoke_data$stroke=as.factor(ifelse(smoke_data$stroke==1, 'YES', 'NO'))

summary(smoke_data$stroke)


head(smoke_data)
```

#Drawing bar plots

```
countgender=table(smoke_data$gender,smoke_data$stroke)

barplot(countgender, main="Gender vs Stroke", xlab="Stroke Yes or No",
col=c("black","red"),legend=rownames(countgender))



counthypertension=table(smoke_data$hypertension,smoke_data$stroke)

barplot(counthypertension, main="Hypertension vs Stroke", xlab="Stroke Yes or No",
col=c("black","red"),legend=rownames(counthypertension))



countheart=table(smoke_data$heart_disease,smoke_data$stroke)

barplot(countheart, main="Heart Disease vs Stroke", xlab="Stroke Yes or No",
col=c("black","red"),legend=rownames(countheart))
```

#Plot for bmi

```
ggplot(smoke_data, aes(y= bmi, x = "", fill = stroke)) +

  geom_boxplot()+

  theme_bw()+

  xlab(" ")
```

```
ggplot(smoke_data, aes(y= avg_glucose_level, x = "", fill = stroke)) +
  geom_boxplot()+
  theme_bw()+
  xlab(" ")

ggplot(smoke_data, aes(y= age, x = "", fill = stroke)) +
  geom_boxplot()+
  theme_bw()+
  xlab(" ")
```

#Anova plot

```
ggplot(smoke_data, aes(y= bmi, x = avg_glucose_level, fill = stroke)) +
  geom_boxplot()+
  theme_bw()+
  xlab(" ")
```

#Performing analysis with Anova

```
bmi =smoke_data$bmi
stroke=smoke_data$stroke
anovatt=lm(bmi~stroke)
summary(anovatt)
```

#Normalising the data

#dealing with numerical variables

```
head(smoke_data)
smoke_num = smoke_data[,c(2,8,9)]
head(smoke_num)
```

#Normalization function

```
norm=function(n) {
  return((n-min(n))/(max(n)-min(n)))
}
```

#normalize

```
smoke_num$age=norm(smoke_num$age)
smoke_num$avg_glucose_level = norm(smoke_num$avg_glucose_level)
smoke_num$bmi=norm(smoke_num$bmi)

head(smoke_num)
```

categorial

```
smoke_cat = smoke_data[,-c(2,8,9)]
head(smoke_cat)
```

#dealing with categorical

```
smoke_cat=data.frame(sapply(smoke_cat,function(x) data.frame(model.matrix(~x-1,data
=smoke_cat))[,-1]))

head(smoke_cat)
```

#Binding the data using cbind function

```
smoke_final = cbind(smoke_num,smoke_cat)

head(smoke_final)


summary(smoke_final$stroke)
```

Performing Hold out evaluation to split the data into train and test.

```
#Splitting the data

require(caTools)

set.seed(101)

sample = sample.split(smoke_final$stroke, SplitRatio = .75)

train = subset(smoke_final, sample == TRUE)

test  = subset(smoke_final, sample == FALSE)
```

#WE have unbalanced data, so using rose function to balance the data and generate random data

```
install.packages("ROSE")

library(ROSE)
```

```
train <- ROSE(stroke ~ ., data = train, seed = 1)$data
test <- ROSE(stroke ~ ., data = test, seed = 1)$data
```

#Building KNN model

```
m1<-knn(train=train, test=test, cl=train$stroke, k=13)
m1
summary(m1)

test$stroke=factor(ifelse(predict==1, 'YES', 'NO'))

confusionMatrix(table(m1 ,test$stroke))
```

Building Logistic regression

```
model <- glm (stroke ~ ., data = train, family = "binomial")

summary(model)

predict1 <- predict(model, test, type = 'response')
table(test$stroke, predict1 > 0.5)
tapply(predict1, test$stroke, mean)

predict1=as.numeric(predict1>0.4, 'YES', 'NO')
str(predict1)
head(predict1)
```

```
confusionMatrix(table(predict1 ,test$stroke))
```

#ROC:To check which model is better

```
library(ROCR)
ROCRpred <- prediction(predict, test$stroke)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```

GROUP NO:282

HealthCare Data to Predict Heart Stroke

Our Data

```
> str(train_data)
'data.frame':   43400 obs. of  12 variables:
 $ id               : int  30669 30468 16523 56543 46136 32257 52800 41413 15266 28674 ...
 $ gender           : Factor w/ 3 levels "Female","Male",..: 2 2 1 1 2 1 1 1 1 1 ...
 $ age              : num  3 58 8 70 14 47 52 75 32 74 ...
 $ hypertension     : int  0 1 0 0 0 0 0 0 0 1 ...
 $ heart_disease    : int  0 0 0 0 0 0 0 1 0 0 ...
 $ ever_married     : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 2 2 2 2 2 ...
 $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 1 4 4 4 3 4 4 5 4 5 ...
 $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 1 2 2 1 1 2 2 1 1 2 ...
 $ avg_glucose_level: num  95.1 88 110.9 69 161.3 ...
 $ bmi              : num  18 39.2 17.6 35.9 19.1 50.1 17.7 27 32.3 54.6 ...
 $ smoking_status   : Factor w/ 4 levels "","formerly smoked",..: 1 3 1 2 1 1 2 3 4 3 ...
 $ stroke           : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
#Check if there are any missing values in the columns
na_count=sapply(train_data, function(y) sum(length(which(is.na(y)))))
na_count=data.frame(na_count)
na_count
```

Output:

```
> na_count
                   na_count
id                        0
gender                    0
age                       0
hypertension              0
heart_disease             0
ever_married              0
work_type                 0
Residence_type            0
avg_glucose_level         0
bmi                    1462
smoking_status            0
stroke                    0
```

#To find missing value:

```
#We find that there are 1462 missing values for the bmi column
#Replace those missing valyes with the mean value
train_data$bmi=ifelse(is.na(train_data$bmi),ave(train_data$bmi,FUN = function(x) mean(x,na.rm = TRUE)),train_data$bmi)
```

No missing values now:

```
> na_count1
                   na_count1
id                         0
gender                     0
age                        0
hypertension               0
heart_disease              0
ever_married               0
work_type                  0
Residence_type             0
avg_glucose_level          0
bmi                        0
smoking_status             0
stroke                     0
```

```
#Removing the ID column as we do not need the same for prediction of heart stroke
smoke_data=smoke_data[ , -which(names(smoke_data) %in% c("id"))]
head(smoke_data)
```

ID column is removed:

```
> head(smoke_data)
   gender age hypertension heart_disease ever_married    work_type Residence_type avg_glucose_level  bmi  smoking_status stroke
2    Male  58          1           0         Yes      Private         Urban           87.96 39.2    never smoked      0
4  Female  70          0           0         Yes      Private         Rural           69.04 35.9 formerly smoked      0
7  Female  52          0           0         Yes      Private         Urban           77.59 17.7 formerly smoked      0
8  Female  75          0           1         Yes Self-employed         Rural          243.53 27.0    never smoked      0
9  Female  32          0           0         Yes      Private         Rural           77.67 32.3         smokes      0
10 Female  74          1           0         Yes Self-employed         Urban          205.84 54.6    never smoked      0
```

Converting hypertension, heart_diseases, smoke_data into yes/no:

```
#yes no conversion for visualisation
smoke_data$hypertension=as.factor(ifelse(smoke_data$hypertension==1, 'YES', 'NO'))
summary(smoke_data$hypertension)

smoke_data$heart_disease=as.factor(ifelse(smoke_data$heart_disease==1, 'YES', 'NO'))
summary(smoke_data$heart_disease)

smoke_data$stroke=as.factor(ifelse(smoke_data$stroke==1, 'YES', 'NO'))
summary(smoke_data$stroke)

head(smoke_data)

#Drawing bar plots
```

Output:hypertension, heart_diseases, smoke_data got converted into yes/no type.

```
> head(smoke_data)
   gender age hypertension heart_disease ever_married     work_type Residence_type avg_glucose_level  bmi  smoking_status stroke
2    Male  58          YES           NO          Yes       Private          Urban             87.96 39.2    never smoked     NO
4  Female  70           NO           NO          Yes       Private          Rural             69.04 35.9 formerly smoked     NO
7  Female  52           NO           NO          Yes       Private          Urban             77.59 17.7 formerly smoked     NO
8  Female  75           NO          YES          Yes Self-employed          Rural            243.53 27.0    never smoked     NO
9  Female  32           NO           NO          Yes       Private          Rural             77.67 32.3          smokes     NO
10 Female  74          YES           NO          Yes Self-employed          Urban            205.84 54.6    never smoked     NO
> str(smoke_test$pred)
```

Normalizing the data to scale data in one format:

```
#Normalization function
norm=function(n) {
  return((n-min(n))/(max(n)-min(n)))
}

#normalize
smoke_num$age=norm(smoke_num$age)
smoke_num$avg_glucose_level = norm(smoke_num$avg_glucose_level)
smoke_num$bmi=norm(smoke_num$bmi)

head(smoke_num)
```

```
> head(smoke_num)
          age avg_glucose_level        bmi
2   0.6666667         0.13959498 0.35531136
4   0.8333333         0.05943908 0.31501832
7   0.5833333         0.09566175 0.09279609
8   0.9027778         0.79867819 0.20634921
9   0.3055556         0.09600068 0.27106227
10  0.8888889         0.63900186 0.54334554
> str(smoke_test$pred)
```

HealthCare Data to Predict Heart Stroke

Categorical:

```
# categorial

smoke_cat = smoke_data[,-c(2,8,9)]
head(smoke_cat)

#dealing with categorical

smoke_cat=data.frame(sapply(smoke_cat,function(x) data.frame(model.matrix(~x-1,data =smoke_cat))[,-1]))
head(smoke_cat)

smoke_final = cbind(smoke_num,smoke_cat)
head(smoke_final)
```

Output:

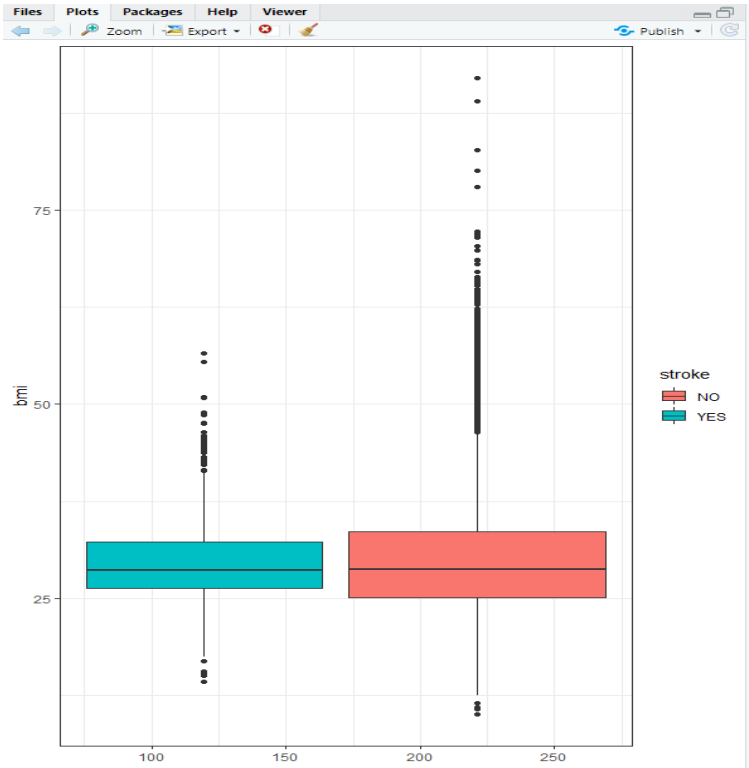
```
> head(smoke_cat)
   gender.xMale gender.xOther hypertension heart_disease ever_married work_type.xGovt_job work_type.xNever_worked work_type.xPrivate
2             1             0            1             0            1                  0                      0                  1
4             0             0            0             0            1                  0                      0                  1
7             0             0            0             0            1                  0                      0                  1
8             0             0            0             1            1                  0                      0                  0
9             0             0            0             0            1                  0                      0                  1
10            0             0            1             0            1                  0                      0                  0
   work_type.xSelf.employed Residence_type smoking_status.xformerly.smoked smoking_status.xnever.smoked smoking_status.xsmokes stroke
2                         0              1                               0                           1                     0      0
4                         0              0                               1                           0                     0      0
7                         0              1                               1                           0                     0      0
8                         1              0                               0                           1                     0      0
9                         0              0                               0                           0                     1      0
10                        1              1                               0                           1                     0      0
```

Output after model got combined:

```
> head(smoke_final)
         age avg_glucose_level        bmi gender.xMale gender.xOther hypertension heart_disease ever_married work_type.xGovt_job work_type.xNever_worked
2  0.6666667        0.13959498 0.35531136           1             0            1             0            1                  0                      0
4  0.8333333        0.05943908 0.31501832           0             0            0             0            1                  0                      0
7  0.5833333        0.09566175 0.09279609           0             0            0             0            1                  0                      0
8  0.9027778        0.79867819 0.20634921           0             0            0             1            1                  0                      0
9  0.3055556        0.09600068 0.27106227           0             0            0             0            1                  0                      0
10 0.8888889        0.63900186 0.54334554           0             0            1             0            1                  0                      0
   work_type.xPrivate work_type.xSelf.employed Residence_type smoking_status.xformerly.smoked smoking_status.xnever.smoked smoking_status.xsmokes stroke
2                   1                       0              1                               0                           1                     0      0
4                   1                       0              0                               1                           0                     0      0
7                   1                       0              1                               1                           0                     0      0
8                   0                       1              0                               0                           1                     0      0
9                   1                       0              0                               0                           0                     1      0
10                  0                       1              1                               0                           1                     0      0
```

ANOVA Plot:

Comparing bmi and stroke:



ANOVA Hypothesis:

```
> summary(anovatt)

Call:
lm(formula = bmi ~ avg_glucose_level)

Residuals:
    Min      1Q  Median      3Q     Max 
-19.097  -4.774  -1.047   3.581  63.294 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2.724e+01  1.020e-01  267.08   <2e-16 ***
avg_glucose_level 2.581e-02  8.744e-04   29.52   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.974 on 30106 degrees of freedom
Multiple R-squared:  0.02812,	Adjusted R-squared:  0.02809 
F-statistic: 871.2 on 1 and 30106 DF,  p-value: < 2.2e-16
```

#HOLD-OUT evaluation: For splitting the data

```
sample = sample.split(smoke_final$stroke, SplitRatio = .75)
train = subset(smoke_final, sample == TRUE)
test  = subset(smoke_final, sample == FALSE)
```

#KNN Classification Model:

```
m1<-knn(train=smoke_train, test=smoke_test, cl=smoke_train$stroke, k=13)
summary(m1)
```

Output:

```
> summary(m1)
    0    1
 3795 3733
```

Confusion Matrix For KNN:

```
> confusionMatrix(table(m1 ,smoke_test$stroke))
Confusion Matrix and Statistics


m1     0    1
  0 8848   82
  1    0  103

               Accuracy : 0.9909
                 95% CI : (0.9887, 0.9928)
    No Information Rate : 0.9795
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.711

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 1.0000
            Specificity : 0.5568
         Pos Pred Value : 0.9908
         Neg Pred Value : 1.0000
             Prevalence : 0.9795
         Detection Rate : 0.9795
   Detection Prevalence : 0.9886
      Balanced Accuracy : 0.7784

       'Positive' Class : 0
```

#LOGISTIC REGRESSION MODEL:

```
#logistic regression model
model <- glm (stroke ~ ., data = train, family = "binomial")

summary(model)
```

Output: We can see significant variable marked as *

```
> model <- glm (stroke ~ ., data = train, family = "binomial")
> summary(model)

Call:
glm(formula = stroke ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6896  -0.8615  -0.2203   0.8773   2.5645

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -2.991444   0.086026 -34.774  < 2e-16 ***
age                             3.933301   0.070963  55.428  < 2e-16 ***
avg_glucose_level               0.762720   0.058328  13.076  < 2e-16 ***
bmi                            -0.300201   0.170396  -1.762 0.078105 .
gender.xMale                    0.016143   0.027503   0.587 0.557238
gender.xOther                  -1.946721   1.972916  -0.987 0.323779
hypertension                    0.464212   0.033558  13.833  < 2e-16 ***
heart_disease                   0.637136   0.040795  15.618  < 2e-16 ***
ever_married                    0.011332   0.038453   0.295 0.768224
work_type.xGovt_job            -0.076232   0.045978  -1.658 0.097313 .
work_type.xNever_worked        -0.672249   0.492428  -1.365 0.172199
work_type.xPrivate             -0.053000   0.037052  -1.430 0.152601
work_type.xSelf.employed        0.125869   0.039872   3.157 0.001595 **
Residence_type                  0.069158   0.026533   2.606 0.009149 **
smoking_status.xformerly.smoked 0.009883   0.037752   0.262 0.793497
smoking_status.xnever.smoked   -0.121908   0.036054  -3.381 0.000721 ***
smoking_status.xsmokes          0.165423   0.039928   4.143 3.43e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31302  on 22579  degrees of freedom
Residual deviance: 24354  on 22563  degrees of freedom
AIC: 24388

Number of Fisher Scoring iterations: 4
```

```
> table(test$stroke, predict1 > 0.5)

    FALSE TRUE
  0  2602 1194
  1   842 2890
```

Confusion Matrix for logistic regression:

```
> confusionMatrix(table(predict1 ,test$stroke))
Confusion Matrix and Statistics

predict1    0    1
       0 2602  842
       1 1194 2890

               Accuracy : 0.7295
                 95% CI : (0.7194, 0.7396)
    No Information Rate : 0.5043
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4595

 Mcnemar's Test P-Value : 7.316e-15

            Sensitivity : 0.6855
            Specificity : 0.7744
         Pos Pred Value : 0.7555
         Neg Pred Value : 0.7076
             Prevalence : 0.5043
         Detection Rate : 0.3456
   Detection Prevalence : 0.4575
      Balanced Accuracy : 0.7299

       'Positive' Class : 0
```

#ROC CURVE TO CHECK WHICH MODEL IS BETTER: