R Notebook

Importing Libraries

```
pacman::p_load(tidyverse, data.table, reshape, rpart, rpart.plot, caret, e1071, forecast, leaps, readxl
```

Set WD

```
# setwd("C:/Users/tutej/Documents/UTD MSITM/SEM II Summer/Project")
```

Importing Dataset

```
df <- read.csv('healthcare-dataset-stroke-data.csv')
View(df)
head(df)
```

```
##      id gender age hypertension heart_disease ever_married     work_type
## 1  9046   Male  67            0             1          Yes       Private
## 2 51676 Female  61            0             0          Yes Self-employed
## 3 31112   Male  80            0             1          Yes       Private
## 4 60182 Female  49            0             0          Yes       Private
## 5  1665 Female  79            1             0          Yes Self-employed
## 6 56669   Male  81            0             0          Yes       Private
##   Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1          Urban            228.69 36.6 formerly smoked      1
## 2          Rural            202.21  N/A    never smoked      1
## 3          Rural            105.92 32.5    never smoked      1
## 4          Urban            171.23 34.4          smokes      1
## 5          Rural            174.12   24    never smoked      1
## 6          Urban            186.21   29 formerly smoked      1
```

```
dt <- setDT(df)
head(dt)
```

```
##       id gender age hypertension heart_disease ever_married     work_type
## 1:  9046   Male  67            0             1          Yes       Private
## 2: 51676 Female  61            0             0          Yes Self-employed
## 3: 31112   Male  80            0             1          Yes       Private
## 4: 60182 Female  49            0             0          Yes       Private
## 5:  1665 Female  79            1             0          Yes Self-employed
## 6: 56669   Male  81            0             0          Yes       Private
##    Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1:          Urban            228.69 36.6 formerly smoked      1
## 2:          Rural            202.21  N/A    never smoked      1
## 3:          Rural            105.92 32.5    never smoked      1
## 4:          Urban            171.23 34.4          smokes      1
## 5:          Rural            174.12   24    never smoked      1
## 6:          Urban            186.21   29 formerly smoked      1
```

Checking Null values

```
colSums(dt == 'N/A')
```

```
##                id            gender               age      hypertension 
##                 0                 0                 0                 0 
##     heart_disease      ever_married         work_type    Residence_type 
##                 0                 0                 0                 0 
## avg_glucose_level               bmi    smoking_status            stroke 
##                 0               201                 0                 0
```

```
# Converting N/A values in BMI to NA
dt[dt=='N/A'] <- NA
head(dt$bmi)
```

```
## [1] "36.6" NA     "32.5" "34.4" "24"   "29"
```

```
# Checking Null Values
colSums(is.na(dt))
```

```
##                id            gender               age      hypertension 
##                 0                 0                 0                 0 
##     heart_disease      ever_married         work_type    Residence_type 
##                 0                 0                 0                 0 
## avg_glucose_level               bmi    smoking_status            stroke 
##                 0               201                 0                 0
```

Handling Null Values

```
dt$bmi <- sapply(dt$bmi, as.numeric)
# We are gonna replace the value with the mean value
dt$bmi <- ifelse(is.na(dt$bmi),
                     ave(dt$bmi, FUN = function(x) mean(x, na.rm = TRUE)),
                     dt$bmi)

colSums(is.na(dt))
```

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

```
# No more N/A values in BMI column
```

Value Count in Categorical Column

```
cat("Gender")
```

```
## Gender
```

```r
table(dt$gender)
```

```
## 
## Female   Male  Other 
##   2994   2115      1
```

```r
cat("\nHypertension")
```

```
## 
## Hypertension
```

```r
table(dt$hypertension)
```

```
## 
##    0    1 
## 4612  498
```

```r
cat("\nEver Married")
```

```
## 
## Ever Married
```

```r
table(dt$ever_married)
```

```
## 
##   No  Yes 
## 1757 3353
```

```r
cat("\nWork Type")
```

```
## 
## Work Type
```

```r
table(dt$work_type)
```

```
## 
##      children      Govt_job  Never_worked       Private Self-employed 
##           687           657            22          2925           819
```

```r
cat("\nResidence Type")
```

```
## 
## Residence Type
```

```r
table(dt$Residence_type)
```

```
## 
## Rural Urban 
##  2514  2596
```

```r
cat("\nSmoking Status")
```

```
##
## Smoking Status
```

```r
table(dt$smoking_status)
```

```
##
## formerly smoked    never smoked          smokes         Unknown
##             885            1892             789            1544
```

```r
cat("\nHeart Disease")
```

```
##
## Heart Disease
```

```r
table(dt$heart_disease)
```

```
##
##    0    1
## 4834  276
```

Handling Gender Column Values

```r
# We will remove others from gender as there is only one row
dt <- subset(dt, gender!='Other')
table(dt$gender)
```

```
##
## Female   Male
##   2994   2115
```

Converting Categorical Columns to Factors

```r
summary(dt)
```

```
##        id           gender               age         hypertension
##  Min.   :   67   Length:5109        Min.   : 0.08   Min.   :0.00000
##  1st Qu.:17740   Class :character   1st Qu.:25.00   1st Qu.:0.00000
##  Median :36922   Mode  :character   Median :45.00   Median :0.00000
##  Mean   :36514                      Mean   :43.23   Mean   :0.09748
##  3rd Qu.:54643                      3rd Qu.:61.00   3rd Qu.:0.00000
##  Max.   :72940                      Max.   :82.00   Max.   :1.00000
##  heart_disease     ever_married        work_type          Residence_type
##  Min.   :0.00000   Length:5109        Length:5109        Length:5109
##  1st Qu.:0.00000   Class :character   Class :character   Class :character
##  Median :0.00000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.05402
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  avg_glucose_level      bmi        smoking_status        stroke
##  Min.   : 55.12    Min.   :10.30   Length:5109        Min.   :0.00000
##  1st Qu.: 77.24    1st Qu.:23.80   Class :character   1st Qu.:0.00000
##  Median : 91.88    Median :28.40   Mode  :character   Median :0.00000
##  Mean   :106.14    Mean   :28.89                      Mean   :0.04874
##  3rd Qu.:114.09    3rd Qu.:32.80                      3rd Qu.:0.00000
##  Max.   :271.74    Max.   :97.60                      Max.   :1.00000
```

```
str(dt)
```

```
## Classes ’data.table’ and ’data.frame’:   5109 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : num  36.6 28.9 32.5 34.4 24 ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
dt$gender <- as.factor(dt$gender)
dt$ever_married <- as.factor(dt$ever_married)
dt$work_type <- as.factor(dt$work_type)
dt$Residence_type <- as.factor(dt$Residence_type)
dt$smoking_status <- as.factor(dt$smoking_status)
dt$stroke <- as.factor(dt$stroke)

cat("\n Post Conversion Results \n")
```

```
##
##  Post Conversion Results
```

```
str(dt)
```

```
## Classes ’data.table’ and ’data.frame’:   5109 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : Factor w/ 2 levels "Female","Male": 2 1 2 1 1 2 2 1 1 1 ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 1 2 2 ...
##  $ work_type        : Factor w/ 5 levels "children","Govt_job",..: 4 5 4 4 5 4 4 4 4 4 ...
##  $ Residence_type   : Factor w/ 2 levels "Rural","Urban": 2 1 1 2 1 2 1 2 1 2 ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : num  36.6 28.9 32.5 34.4 24 ...
##  $ smoking_status   : Factor w/ 4 levels "formerly smoked",..: 1 2 2 3 2 1 2 2 4 4 ...
##  $ stroke           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

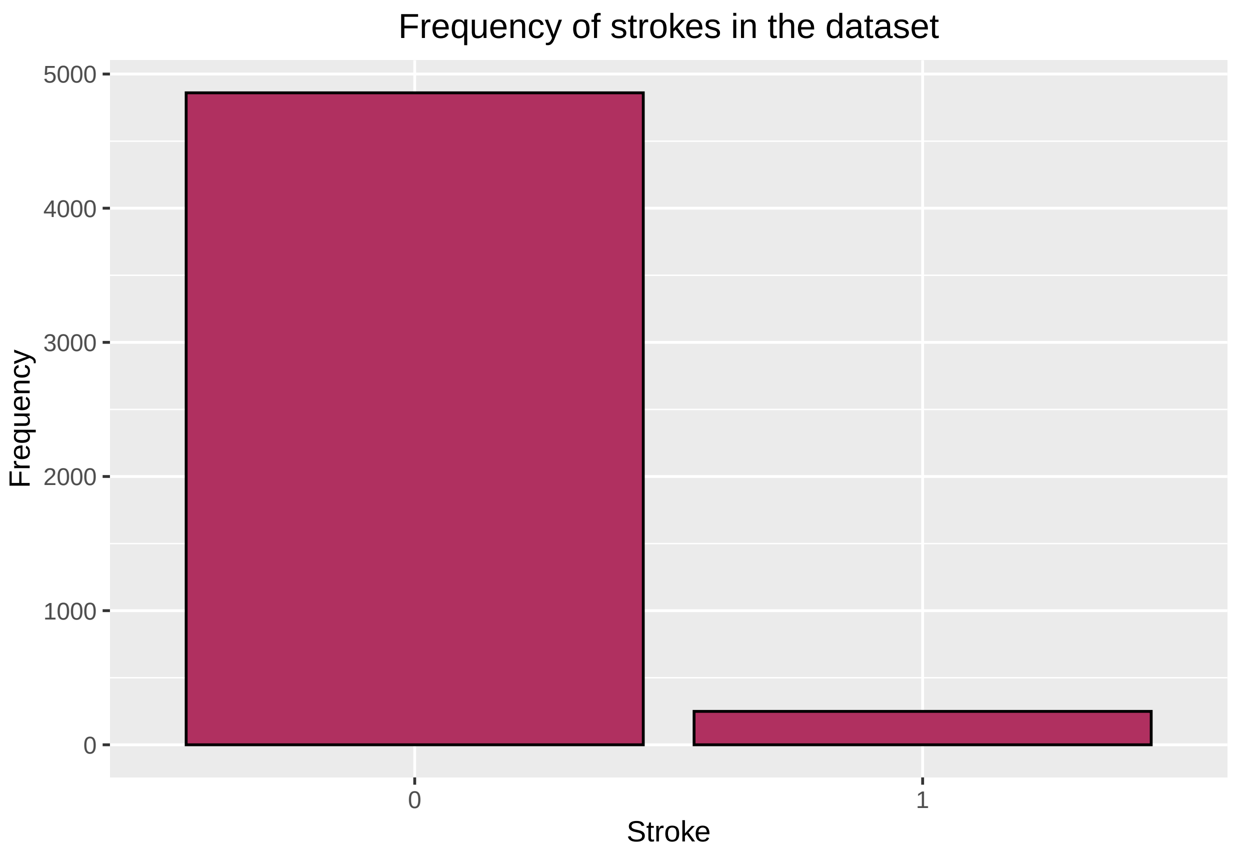
```r
# Removing ID Column as it is of no use to predict
dt <- dt %>% select(-id)
head(dt)
```

```
##    gender age hypertension heart_disease ever_married     work_type
## 1:   Male  67            0             1          Yes       Private
## 2: Female  61            0             0          Yes Self-employed
## 3:   Male  80            0             1          Yes       Private
## 4: Female  49            0             0          Yes       Private
## 5: Female  79            1             0          Yes Self-employed
## 6:   Male  81            0             0          Yes       Private
##    Residence_type avg_glucose_level      bmi  smoking_status stroke
## 1:          Urban            228.69 36.60000 formerly smoked      1
## 2:          Rural            202.21 28.89324    never smoked      1
## 3:          Rural            105.92 32.50000    never smoked      1
## 4:          Urban            171.23 34.40000          smokes      1
## 5:          Rural            174.12 24.00000    never smoked      1
## 6:          Urban            186.21 29.00000 formerly smoked      1
```

EDA

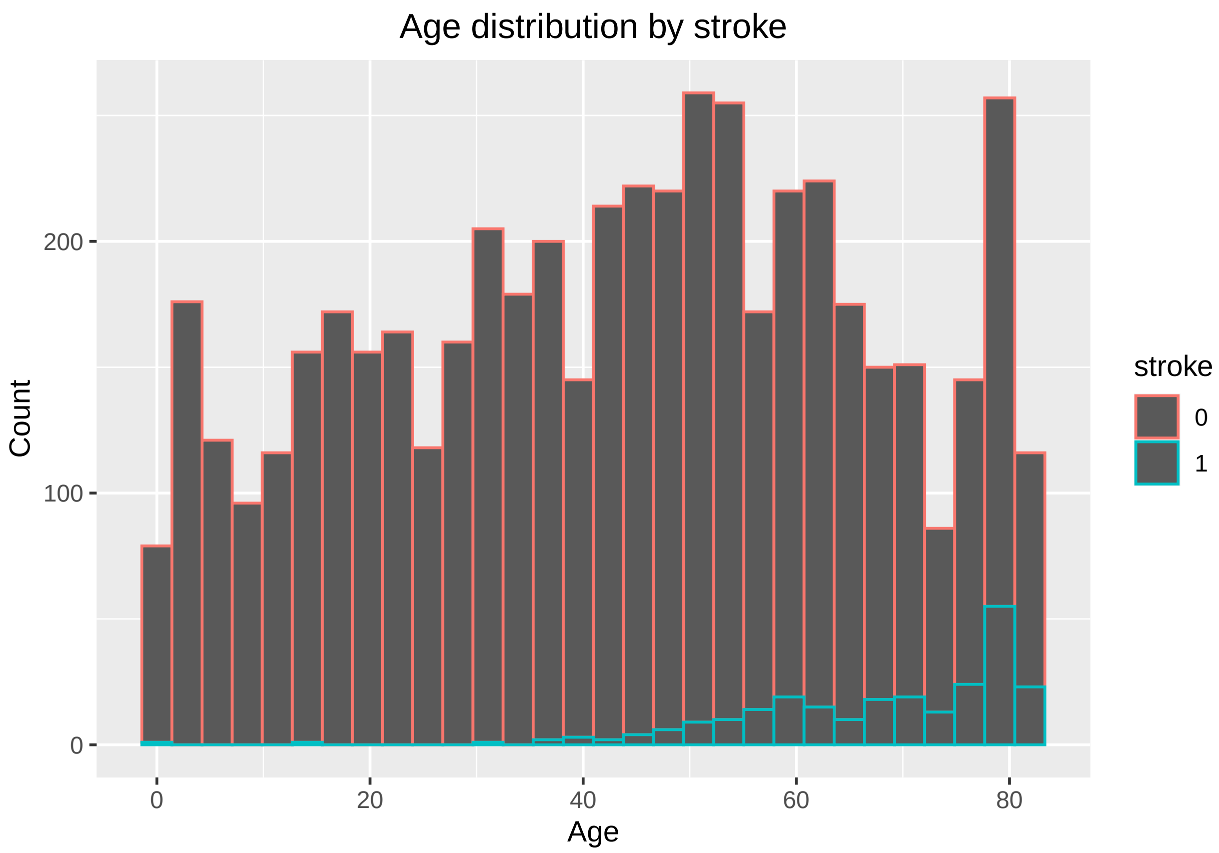
Stroke Count

```r
# Creating copy of dt
strokes.dt <- copy(dt)

#number of stroke cases count
ggplot(data = strokes.dt, aes(x = stroke)) +
  geom_bar(color = "black", fill = "maroon") +
  ggtitle("Frequency of strokes in the dataset") +
  xlab("Stroke") + ylab("Frequency") +
  theme(plot.title = element_text(hjust=0.5))
```



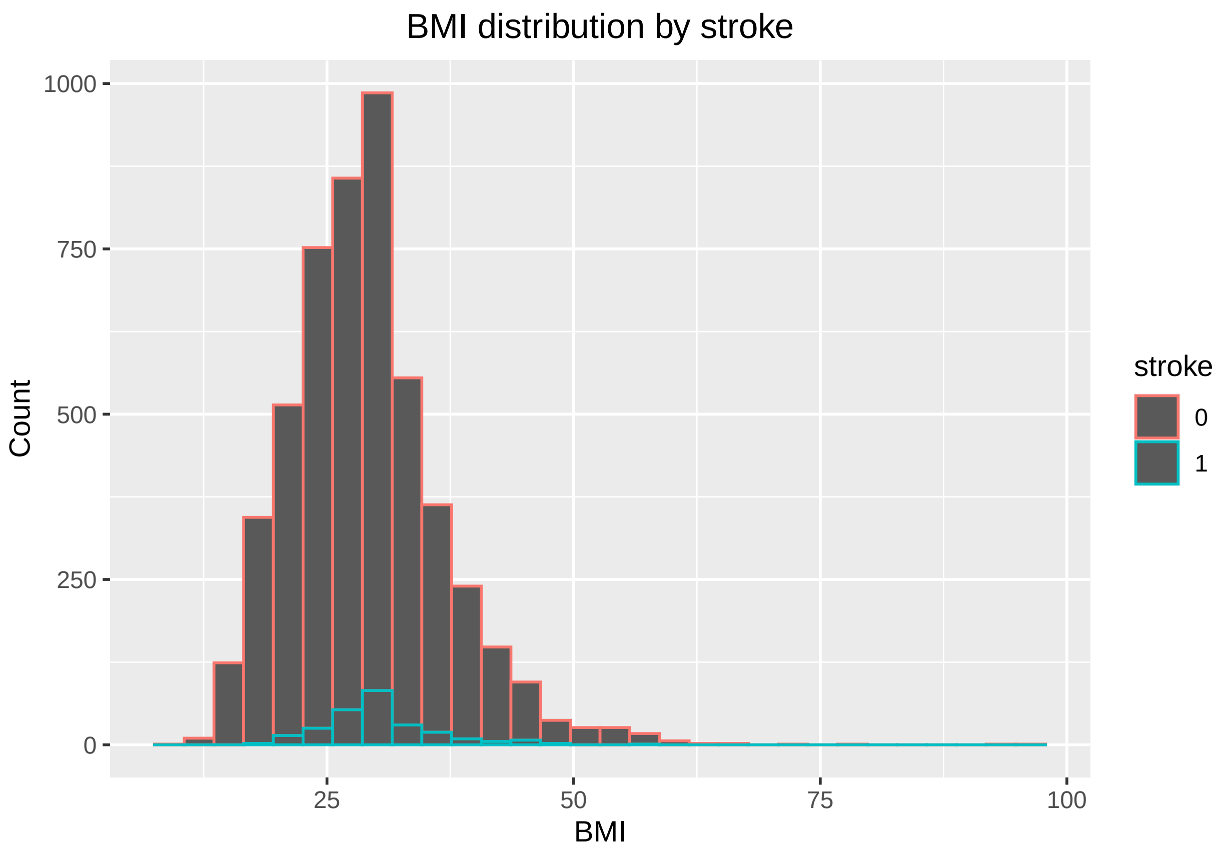
Age Plot

```
#age histogram
ggplot(strokes.dt) +
  geom_histogram(aes(x=age, color = stroke)) +
  ggtitle("Age distribution by stroke") + xlab("Age") + ylab("Count") +
  theme(plot.title = element_text(hjust=0.5))
```



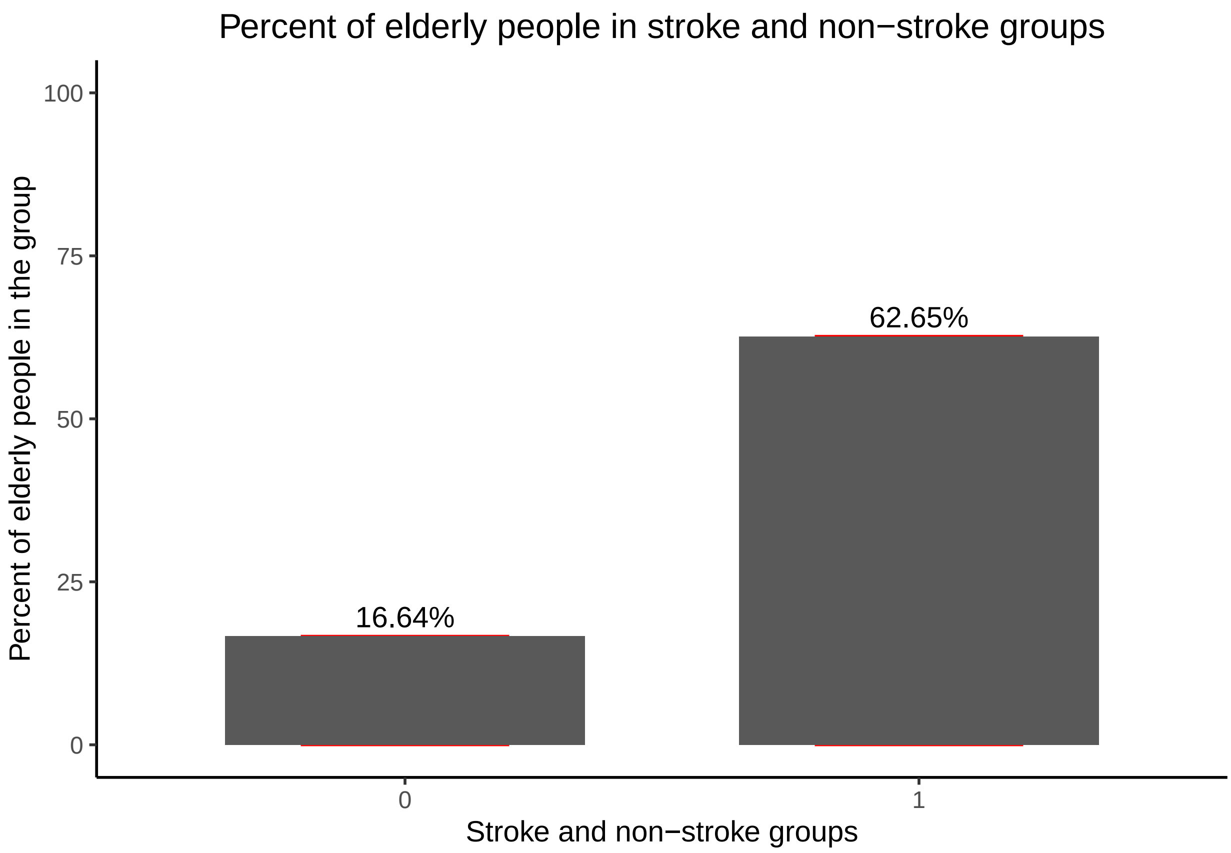
BMI Plot

```
#bmi histogram
ggplot(strokes.dt) +
  geom_histogram(aes(x=bmi, color = stroke)) +
  ggtitle("BMI distribution by stroke") + xlab("BMI") + ylab("Count") +
  theme(plot.title = element_text(hjust=0.5))
```



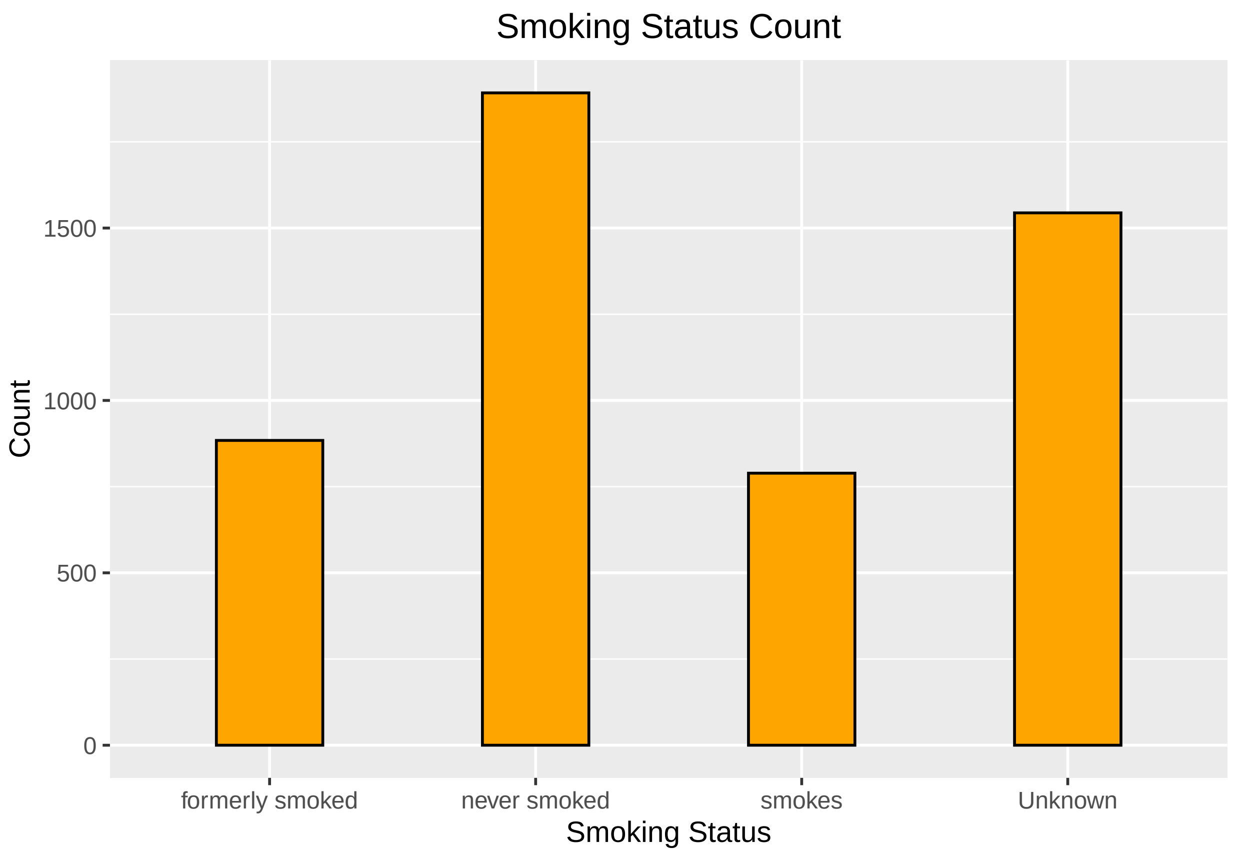
Percent of elderly people in stroke and non-stroke groups

```
strokes.dt[, oldAge := ifelse(age>65,1,0)]
data.for.this.plot <- strokes.dt[,.(percent_age = sum(oldAge)*100/.N), keyby = stroke]
ggplot(data.for.this.plot) +
  geom_bar(aes(x=stroke, y = percent_age), stat = 'identity', width = 0.4,
           fill = "blue", color = "red") +
  ylim(c(0,100)) +
  ylab("Percent of elderly people in the group") +
  xlab("Stroke and non-stroke groups") +
  ggtitle("Percent of elderly people in stroke and non-stroke groups") +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black")) +
  geom_text(aes(label=c("16.64%", "62.65%"), y = percent_age+3, x = stroke)) +
  geom_col(width = .7, aes(x=stroke, y = percent_age)) +
  theme(plot.title = element_text(hjust=0.5))
```



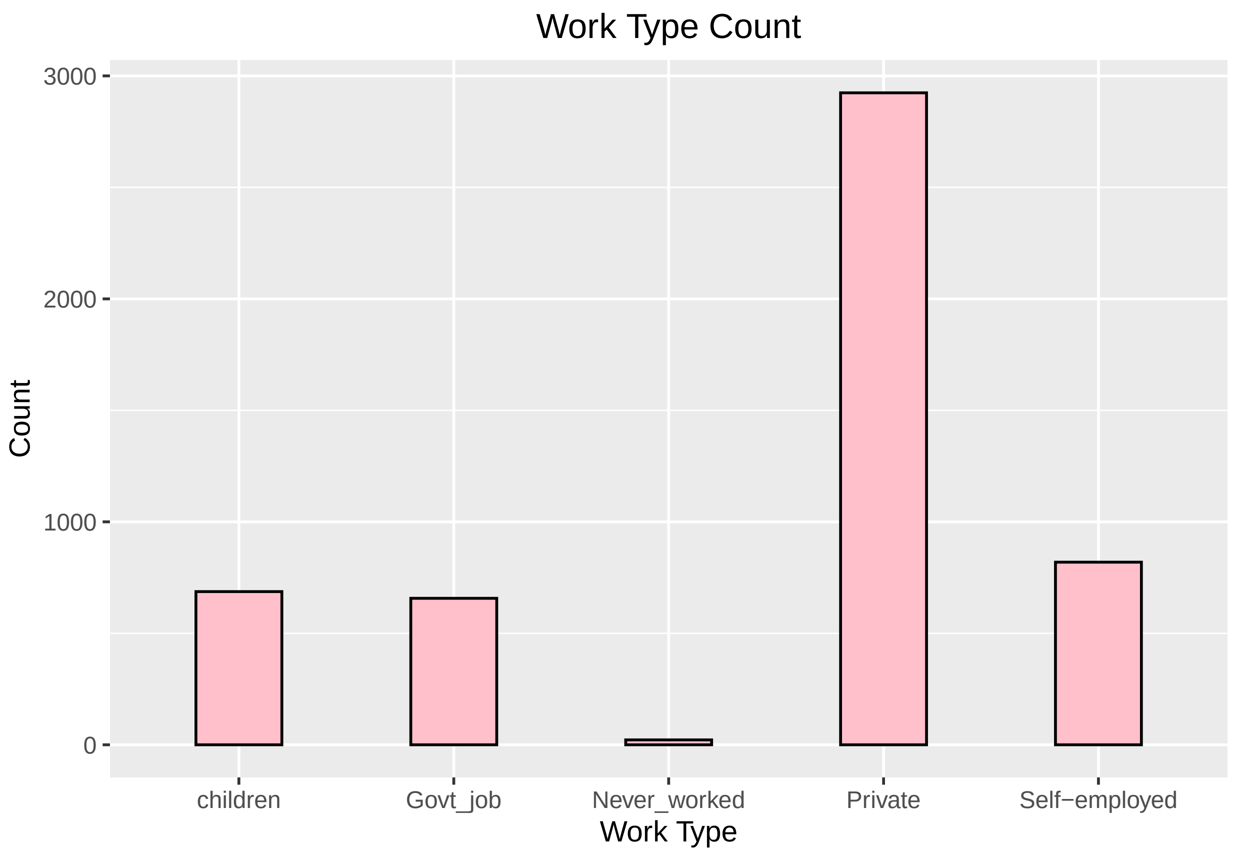
Smoking Status Count

```r
ggplot(data = strokes.dt, aes(x = smoking_status)) +
  geom_bar(color = "black", fill = "Orange", width = 0.4) +
  ggtitle("Smoking Status Count") + xlab("Smoking Status") + ylab("Count") +
  theme(plot.title = element_text(hjust=0.5))
```



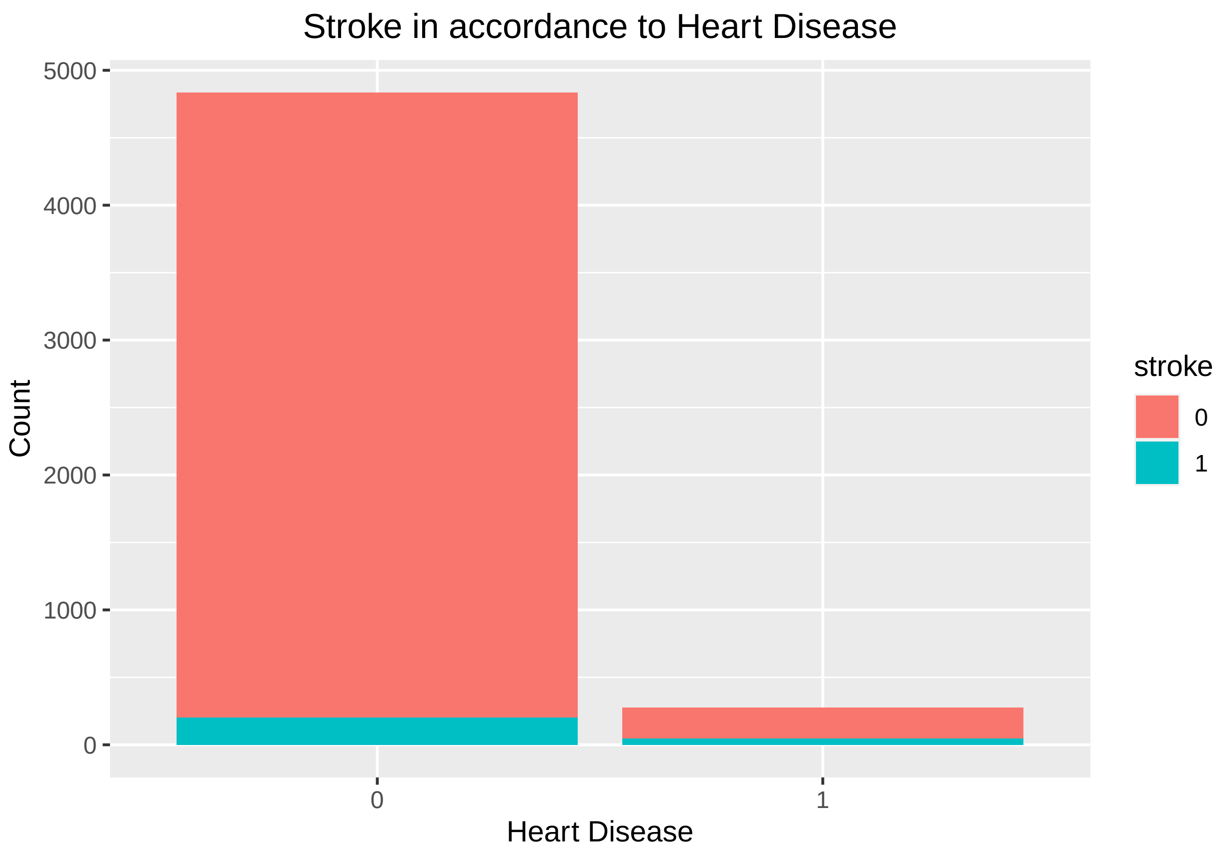
Work Type Count

```
#work type
ggplot(data = strokes.dt, aes(x = work_type)) +
  geom_bar(color = "black", fill = "Pink", width = 0.4) +
  ggtitle("Work Type Count") + xlab("Work Type") + ylab("Count") +
  theme(plot.title = element_text(hjust=0.5))
```



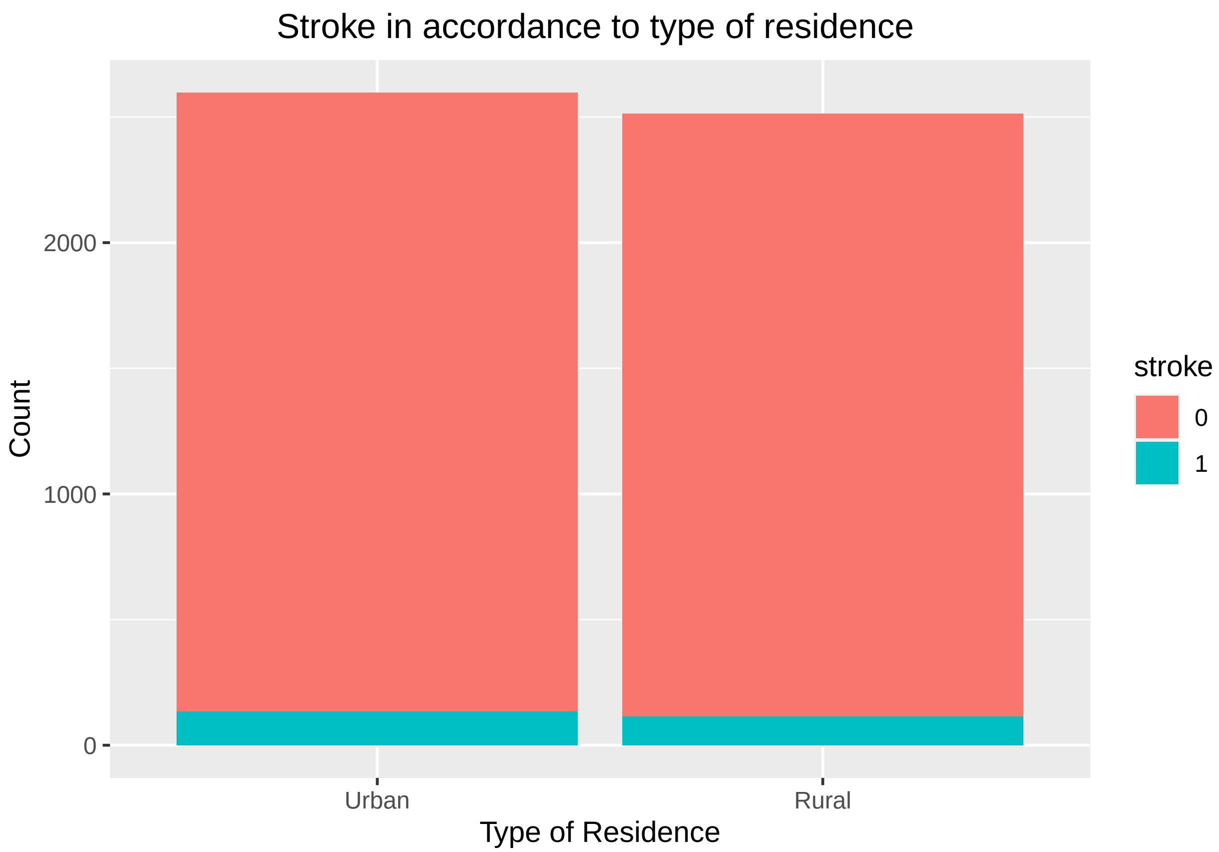
Stroke in accordance to Heart Disease

```r
strokes.dt$heart_disease <- as.factor(strokes.dt$heart_disease)
ggplot(mutate(strokes.dt,heart_disease=fct_infreq(heart_disease)))+
  geom_bar(aes(x=heart_disease,fill=stroke))+labs(x="Heart Disease",y="Count")+
  ggtitle(label="Stroke in accordance to Heart Disease") +
  theme(plot.title = element_text(hjust=0.5))
```



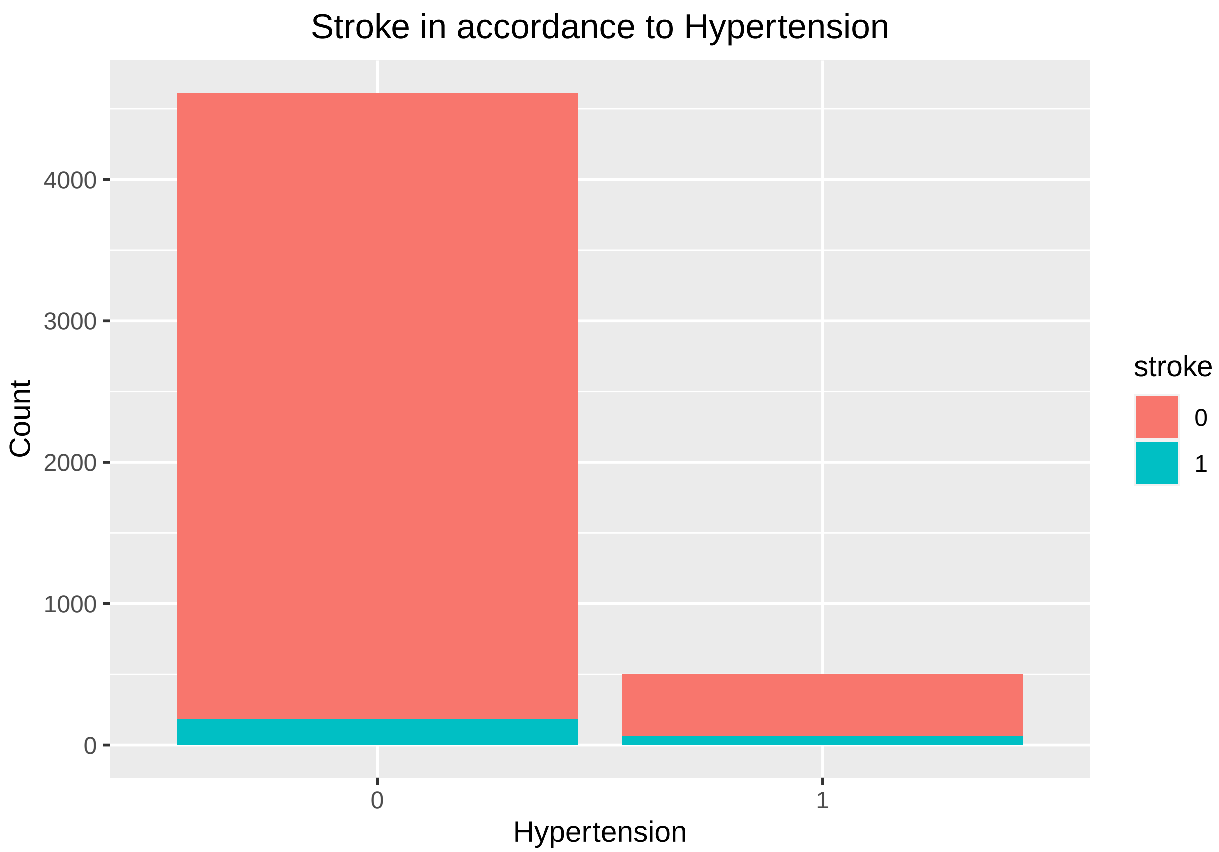
Stroke in accordance to type of Residence

```
ggplot(mutate(strokes.dt,Residence_type=fct_infreq(Residence_type)))+
  geom_bar(aes(x=Residence_type,fill=stroke))+labs(x="Type of Residence",y="Count")+
  ggtitle(label="Stroke in accordance to type of residence ") +
  theme(plot.title = element_text(hjust=0.5))
```



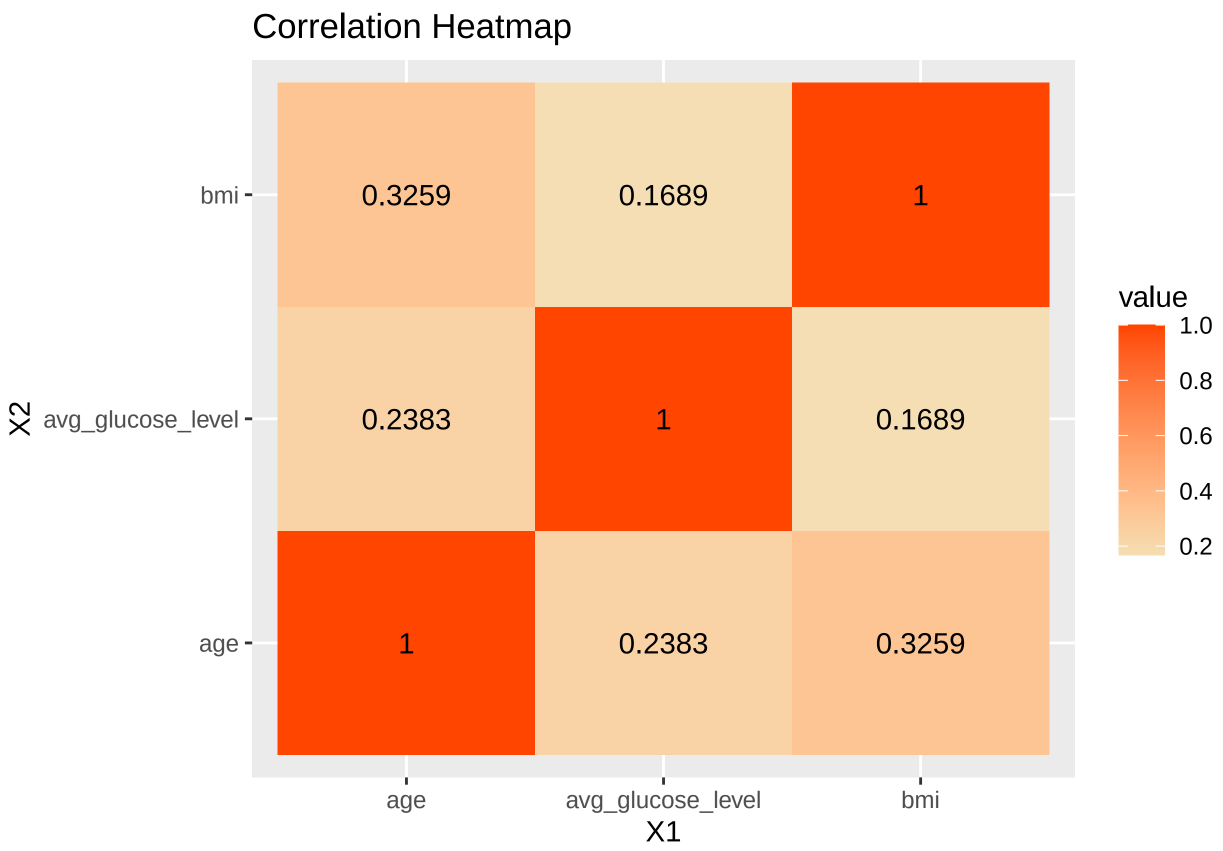
Stroke in accordance to Hypertension

```
strokes.dt$hypertension <- as.factor(strokes.dt$hypertension)
ggplot(mutate(strokes.dt,hypertension=fct_infreq(hypertension)))+
  geom_bar(aes(x=hypertension,fill=stroke))+labs(x="Hypertension",y="Count")+
  ggtitle(label="Stroke in accordance to Hypertension") +
  theme(plot.title = element_text(hjust=0.5))
```



Correlation Between Numerical Features

```
# Correlation between numerical variables
cor.mat <- round(cor(strokes.dt[,c("age", "avg_glucose_level", "bmi")]), 4)
melted.cor.mat <- melt(cor.mat)
ggplot(melted.cor.mat, aes(x = X1, y = X2, fill = value)) +
  scale_fill_gradient(low="wheat", high="orangered") +
  geom_tile() +
  geom_text(aes(x = X1, y = X2, label = value)) +
  ggtitle("Correlation Heatmap")
```



Splitting the dataset

```
set.seed(9)
smp_size <- ceiling(0.8 * nrow(dt))
train_ind <- sample(seq_len(nrow(dt)), size = smp_size)
train <- dt[train_ind,]
test <- dt[-train_ind,]
```

Balancing the dataset

```
cat("\n Stroke Count \n")
```

```
##
##  Stroke Count
```

```
table(train$stroke)
```

```
##
##    0    1
## 3887  201
```

```
# Oversampling
cat("\n Stroke Count Oversampling \n")
```

```
##
##  Stroke Count Oversampling
```

```
over_dt <- ovun.sample(stroke~., data = train, method = "over", N = 7764)$data
table(over_dt$stroke)
```

```
##
##    0    1
## 3887 3877
```

```
prop.table(table(over_dt$stroke))
```

```
##
##        0        1
## 0.500644 0.499356
```

```
summary(over_dt)
```

```
##     gender          age         hypertension    heart_disease    ever_married
##  Female:4531   Min.   : 0.08   Min.   :0.0000   Min.   :0.0000   No :1773
##  Male  :3233   1st Qu.:41.00   1st Qu.:0.0000   1st Qu.:0.0000   Yes:5991
##                Median :59.00   Median :0.0000   Median :0.0000
##                Mean   :55.07   Mean   :0.1777   Mean   :0.1104
##                3rd Qu.:75.00   3rd Qu.:0.0000   3rd Qu.:0.0000
##                Max.   :82.00   Max.   :1.0000   Max.   :1.0000
##          work_type    Residence_type avg_glucose_level      bmi
##  children     : 595   Rural:3730     Min.   : 55.12    Min.   :10.30
##  Govt_job     :1008   Urban:4034     1st Qu.: 77.82    1st Qu.:25.30
##  Never_worked :  19                  Median : 96.97    Median :28.89
##  Private      :4441                  Mean   :118.55    Mean   :29.36
##  Self-employed:1701                  3rd Qu.:144.90    3rd Qu.:32.60
##                                      Max.   :271.74    Max.   :97.60
##          smoking_status stroke
##  formerly smoked:1721   0:3887
##  never smoked   :2878   1:3877
##  smokes         :1276
##  Unknown        :1889
##
##
```

```
# Undersampling
cat("\n Stroke Count Undersampling \n")
```

```
##
##  Stroke Count Undersampling
```

```
under_dt <- ovun.sample(stroke~., data = train, method = "under", N = 412)$data
table(under_dt$stroke)
```

```
##
##   0   1
## 211 201
```

```r
prop.table(table(under_dt$stroke))
```

```
## 
##         0         1 
## 0.5121359 0.4878641
```

```r
summary(under_dt)
```

```
##     gender          age         hypertension    heart_disease    ever_married
##  Female:238   Min.   : 0.08   Min.   :0.0000   Min.   :0.0000   No : 96     
##  Male  :174   1st Qu.:42.00   1st Qu.:0.0000   1st Qu.:0.0000   Yes:316     
##               Median :59.00   Median :0.0000   Median :0.0000               
##               Mean   :55.02   Mean   :0.1845   Mean   :0.1262               
##               3rd Qu.:74.00   3rd Qu.:0.0000   3rd Qu.:0.0000               
##               Max.   :82.00   Max.   :1.0000   Max.   :1.0000               
##           work_type   Residence_type avg_glucose_level      bmi       
##  children     : 28   Rural:185      Min.   : 55.34    Min.   :16.20  
##  Govt_job     : 50   Urban:227      1st Qu.: 77.90    1st Qu.:25.88  
##  Never_worked :  1                  Median : 97.84    Median :28.89  
##  Private      :249                  Mean   :119.49    Mean   :29.37  
##  Self-employed: 84                  3rd Qu.:158.49    3rd Qu.:32.33  
##                                     Max.   :271.74    Max.   :57.20  
##           smoking_status stroke 
##  formerly smoked: 96    0:211  
##  never smoked   :152    1:201  
##  smokes         : 72           
##  Unknown        : 92           
##                                
## 
```

```r
# Both
cat("\n Stroke Count Both \n")
```

```
## 
##  Stroke Count Both
```

```r
both_dt <- ovun.sample(stroke~., data = train, method = "both", N = 3000)$data
table(both_dt$stroke)
```

```
## 
##    0    1 
## 1496 1504
```

```r
prop.table(table(both_dt$stroke))
```

```
## 
##         0         1 
## 0.4986667 0.5013333
```

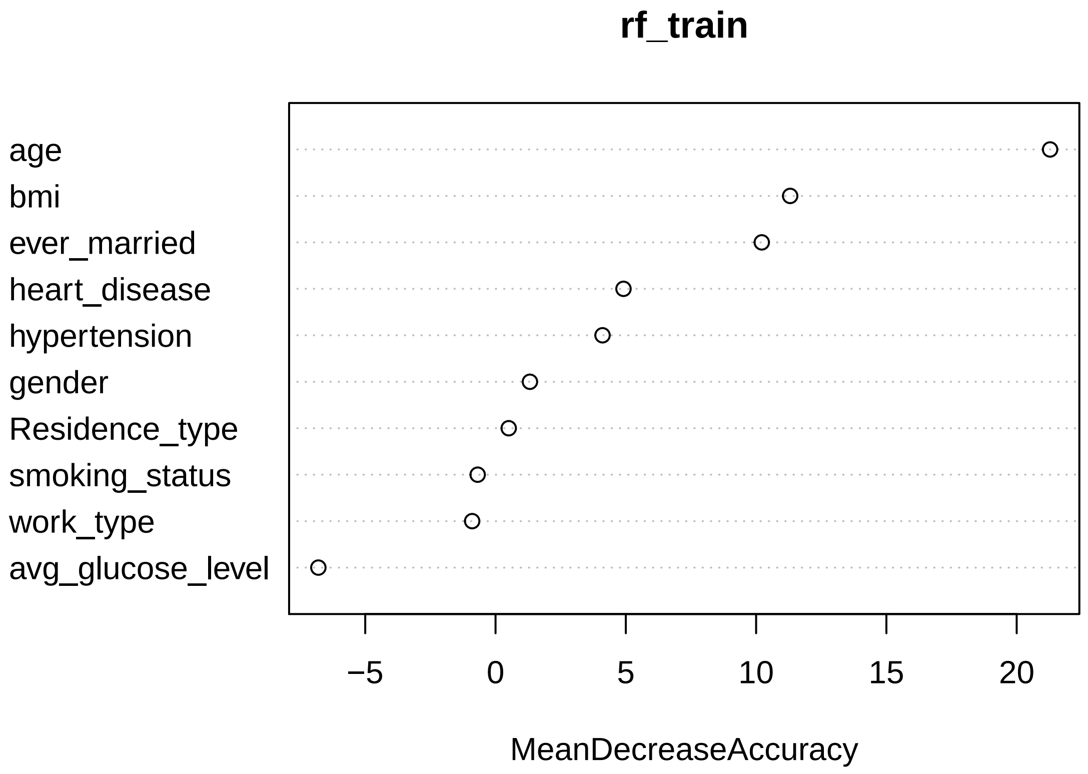
```r
summary(both_dt)
```

```
##     gender          age         hypertension    heart_disease    ever_married
##  Female:1749   Min.   : 0.16   Min.   :0.0000   Min.   :0.0000   No : 712
##  Male  :1251   1st Qu.:40.00   1st Qu.:0.0000   1st Qu.:0.0000   Yes:2288
##                Median :59.00   Median :0.0000   Median :0.0000
##                Mean   :55.02   Mean   :0.1777   Mean   :0.1153
##                3rd Qu.:75.00   3rd Qu.:0.0000   3rd Qu.:0.0000
##                Max.   :82.00   Max.   :1.0000   Max.   :1.0000
##          work_type    Residence_type avg_glucose_level      bmi
##  children     : 221   Rural:1461     Min.   : 55.27    Min.   :11.50
##  Govt_job     : 400   Urban:1539     1st Qu.: 78.18    1st Qu.:25.40
##  Never_worked :  11                  Median : 96.17    Median :28.89
##  Private      :1735                  Mean   :118.07    Mean   :29.60
##  Self-employed: 633                  3rd Qu.:144.90    3rd Qu.:32.73
##                                      Max.   :271.74    Max.   :78.00
##          smoking_status stroke
##  formerly smoked: 688   0:1496
##  never smoked   :1122   1:1504
##  smokes         : 493
##  Unknown        : 697
##
##
```

Random Forest Models

Using Normal Train Data

```r
rf_train <- randomForest(stroke~., importance=T, data = train)

#Variable Importance Plot
varImpPlot(rf_train, type=1)
```

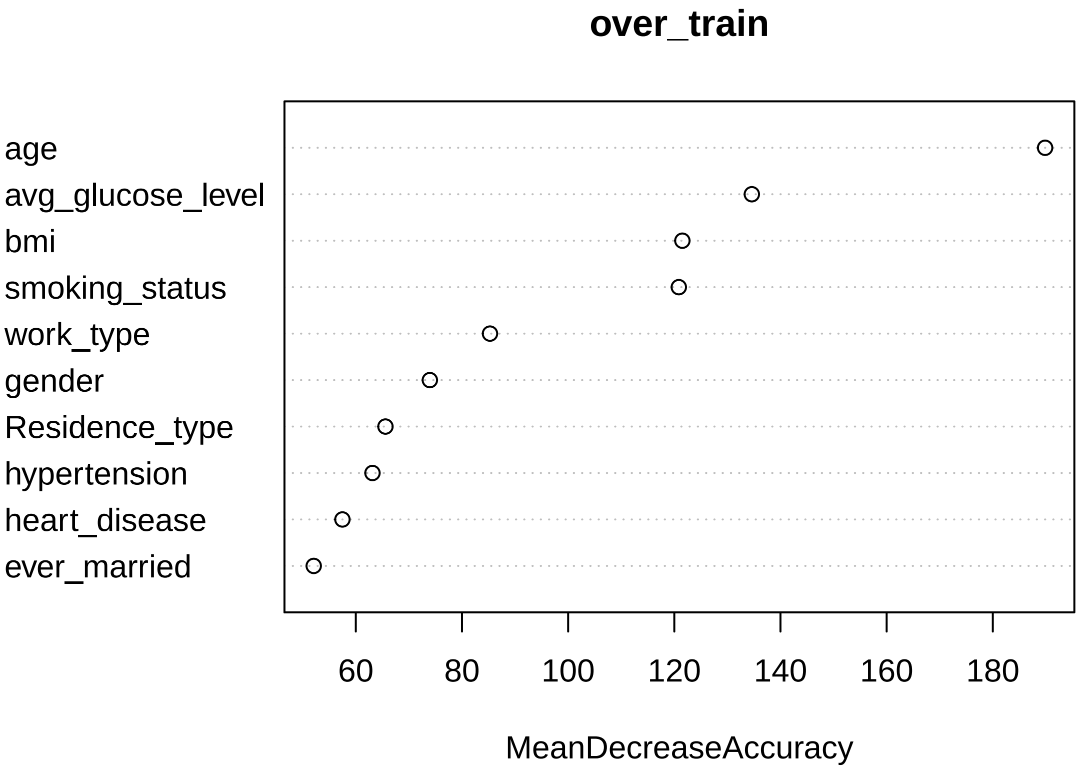


```r
confusionMatrix(predict(rf_train,test), test$stroke, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 971  48
##          1   2   0
##
##                Accuracy : 0.951
##                  95% CI : (0.9359, 0.9634)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 0.6513
##
##                   Kappa : -0.0038
##
##  Mcnemar's Test P-Value : 1.966e-10
##
##             Sensitivity : 0.000000
##             Specificity : 0.997945
##          Pos Pred Value : 0.000000
##          Neg Pred Value : 0.952895
##              Prevalence : 0.047013
##          Detection Rate : 0.000000
##    Detection Prevalence : 0.001959
##       Balanced Accuracy : 0.498972
##
##        'Positive' Class : 1
##
```

Using Oversampled Train Data

```
over_train <- randomForest(stroke~., importance=T, data = over_dt)

#Variable Importance Plot
varImpPlot(over_train, type=1)
```



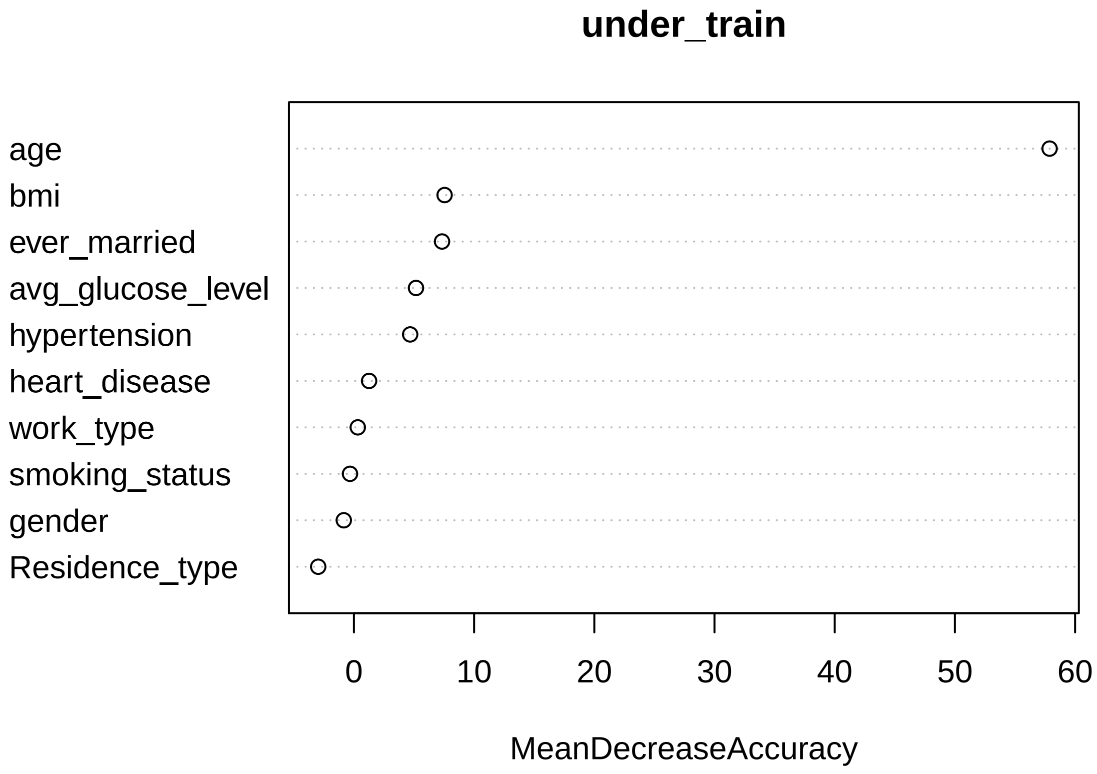
```
confusionMatrix(predict(over_train,test), test$stroke, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 959  42
##          1  14   6
##
##                Accuracy : 0.9452
##                  95% CI : (0.9294, 0.9583)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 0.8935445
```

```
##
##                    Kappa : 0.153
##
##  Mcnemar's Test P-Value : 0.0003085
##
##              Sensitivity : 0.125000
##              Specificity : 0.985612
##           Pos Pred Value : 0.300000
##           Neg Pred Value : 0.958042
##               Prevalence : 0.047013
##           Detection Rate : 0.005877
##     Detection Prevalence : 0.019589
##        Balanced Accuracy : 0.555306
##
##         'Positive' Class : 1
##
```

Using Undersampled Train Data

```r
under_train <- randomForest(stroke~., importance=T, data = under_dt)

#Variable Importance Plot
varImpPlot(under_train, type=1)
```

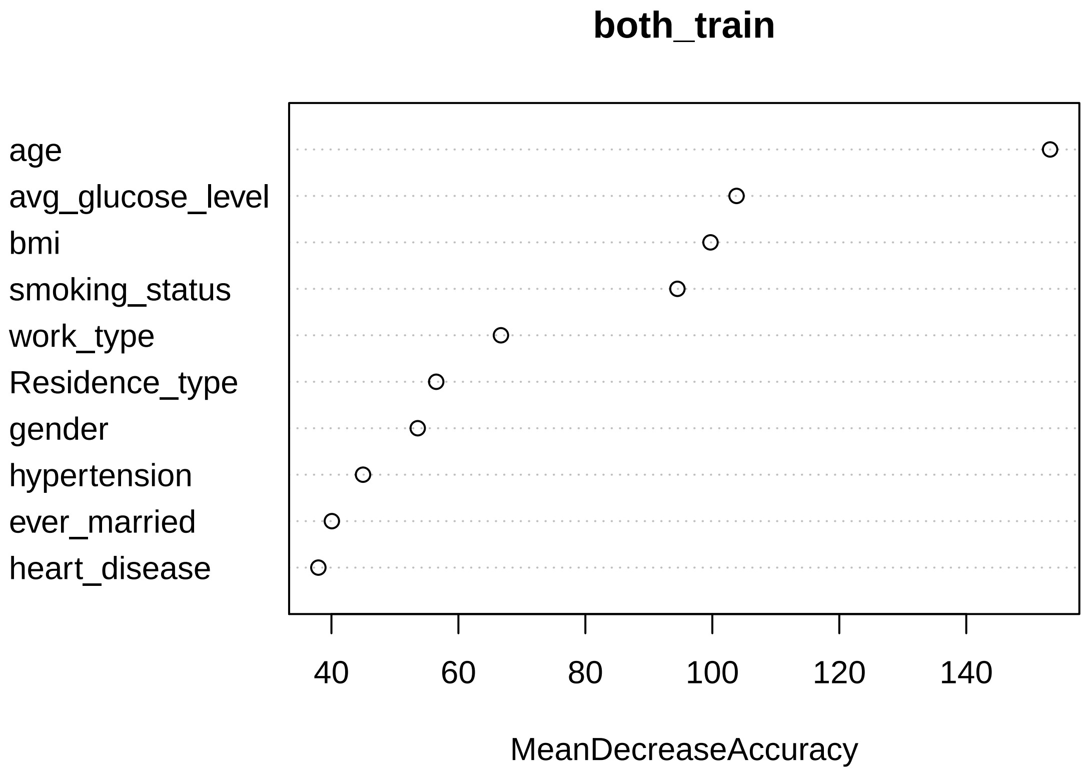


```r
confusionMatrix(predict(under_train,test), test$stroke, positive = ’1’)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 732  15
##          1 241  33
##
##                Accuracy : 0.7493
##                  95% CI : (0.7215, 0.7756)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1358
##
##  Mcnemar’s Test P-Value : <2e-16
##
##             Sensitivity : 0.68750
##             Specificity : 0.75231
##          Pos Pred Value : 0.12044
##          Neg Pred Value : 0.97992
##              Prevalence : 0.04701
##          Detection Rate : 0.03232
##    Detection Prevalence : 0.26836
##       Balanced Accuracy : 0.71991
##
##        ’Positive’ Class : 1
##
```

Using Both Oversampled & Undersampled Train Data

```r
both_train <- randomForest(stroke~., importance=T, data = both_dt)

#Variable Importance Plot
varImpPlot(both_train, type=1)
```



```
confusionMatrix(predict(both_train,test), test$stroke, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 892  30
##          1  81  18
##
##                Accuracy : 0.8913
##                  95% CI : (0.8706, 0.9097)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1938
##
##  Mcnemar's Test P-Value : 2.077e-06
##
##             Sensitivity : 0.37500
##             Specificity : 0.91675
##          Pos Pred Value : 0.18182
##          Neg Pred Value : 0.96746
##              Prevalence : 0.04701
##          Detection Rate : 0.01763
##    Detection Prevalence : 0.09696
##       Balanced Accuracy : 0.64588
##
##        'Positive' Class : 1
##
```

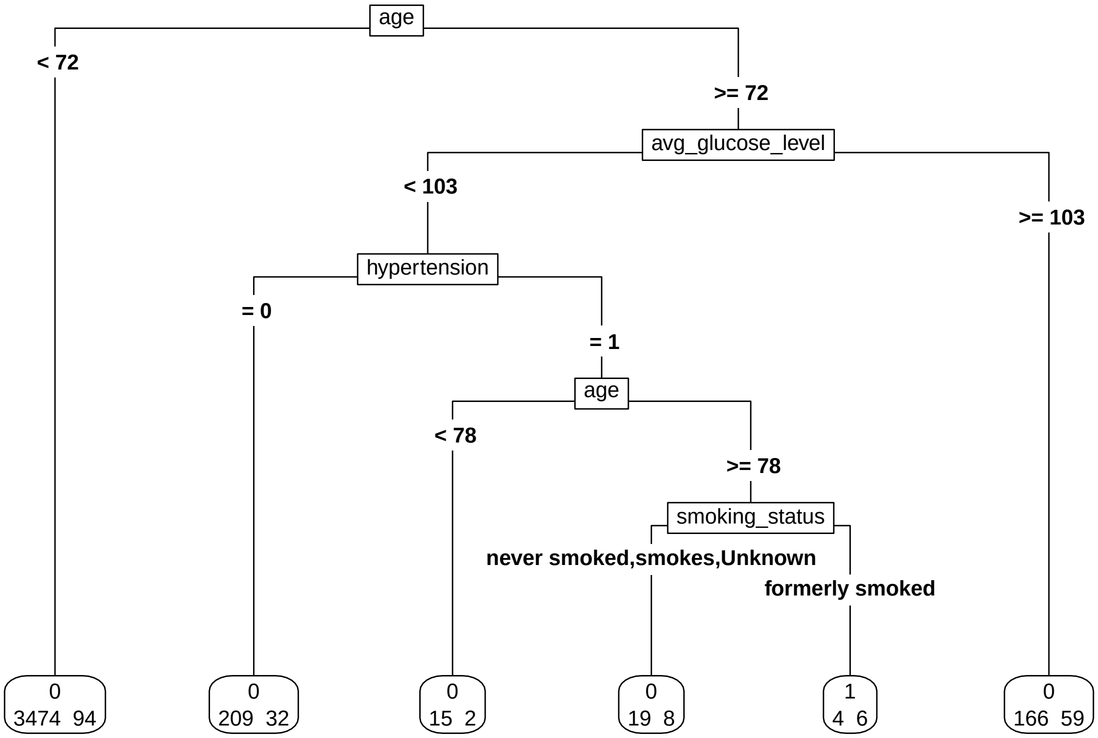
Decision Tree Models

Using Normal Train Data

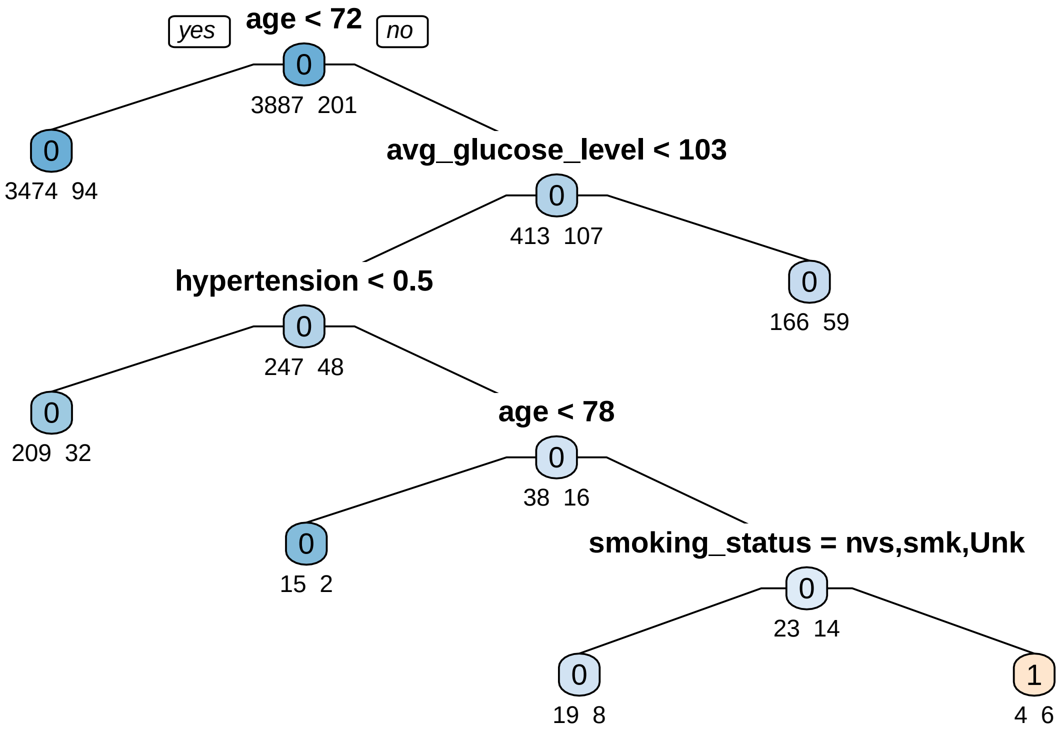
```
dt_train <- rpart(stroke ~ ., data = train, method = "class", cp = 0.00001, maxdepth = 5)

printcp(dt_train)
```

```
##
## Classification tree:
## rpart(formula = stroke ~ ., data = train, method = "class", cp = 1e-05,
##     maxdepth = 5)
##
## Variables actually used in tree construction:
## [1] age               avg_glucose_level hypertension      smoking_status
##
## Root node error: 201/4088 = 0.049168
##
## n= 4088
##
##        CP nsplit rel error xerror     xstd
## 1 0.00199      0   1.00000 1.0000 0.068779
## 2 0.00001      5   0.99005 1.0547 0.070536
```

```
# Graph 1
prp(dt_train, type = 5, extra = 1, under = FALSE, varlen = 0, fallen.leaves = T, faclen = 50)
```



```r
# Graph 2
prp(dt_train, type = 1, extra = 1, under = TRUE, varlen = 0, roundint = FALSE,  split.font = 2, box.pal
```



```r
# Predictions
dt_pred <- predict(dt_train, test, type = "class")
confusionMatrix(dt_pred, test$stroke, positive = "1")
```

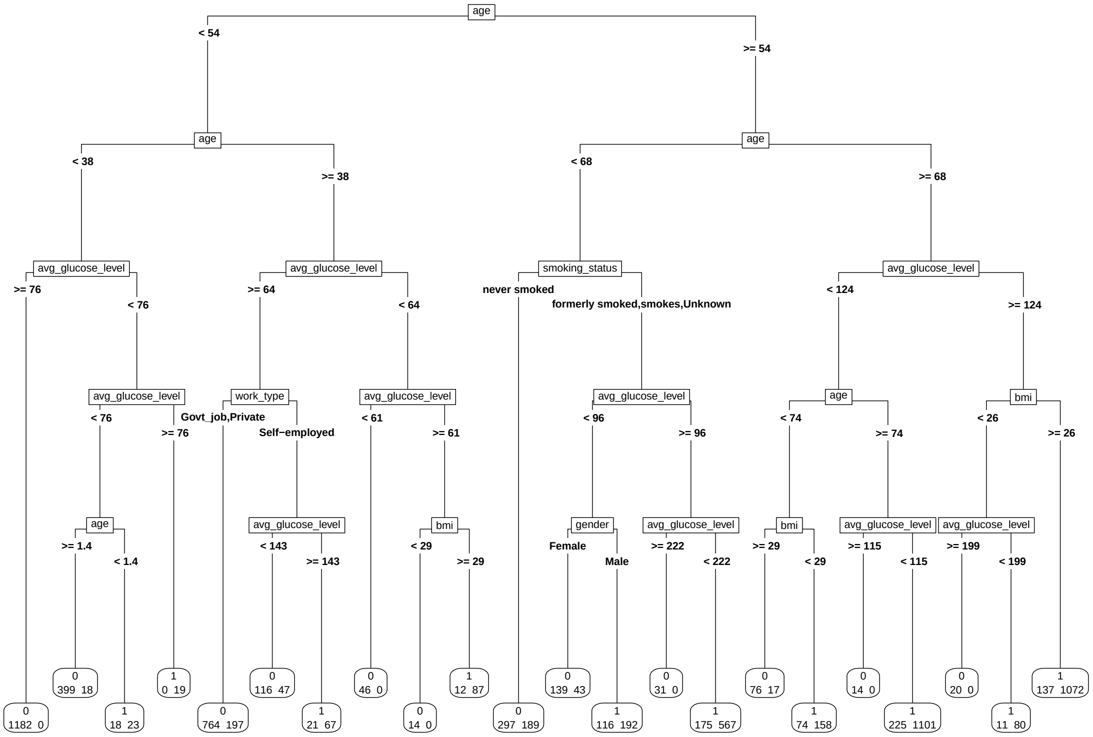
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   0   1
##          0 971  48
##          1   2   0
##
##                Accuracy : 0.951
##                  95% CI : (0.9359, 0.9634)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 0.6513
##
##                   Kappa : -0.0038
##
##  Mcnemar's Test P-Value : 1.966e-10
##
##             Sensitivity : 0.000000
##             Specificity : 0.997945
##          Pos Pred Value : 0.000000
##          Neg Pred Value : 0.952895
##              Prevalence : 0.047013
##          Detection Rate : 0.000000
##    Detection Prevalence : 0.001959
##       Balanced Accuracy : 0.498972
##
##        'Positive' Class : 1
##
```

Using Oversampled Train Data

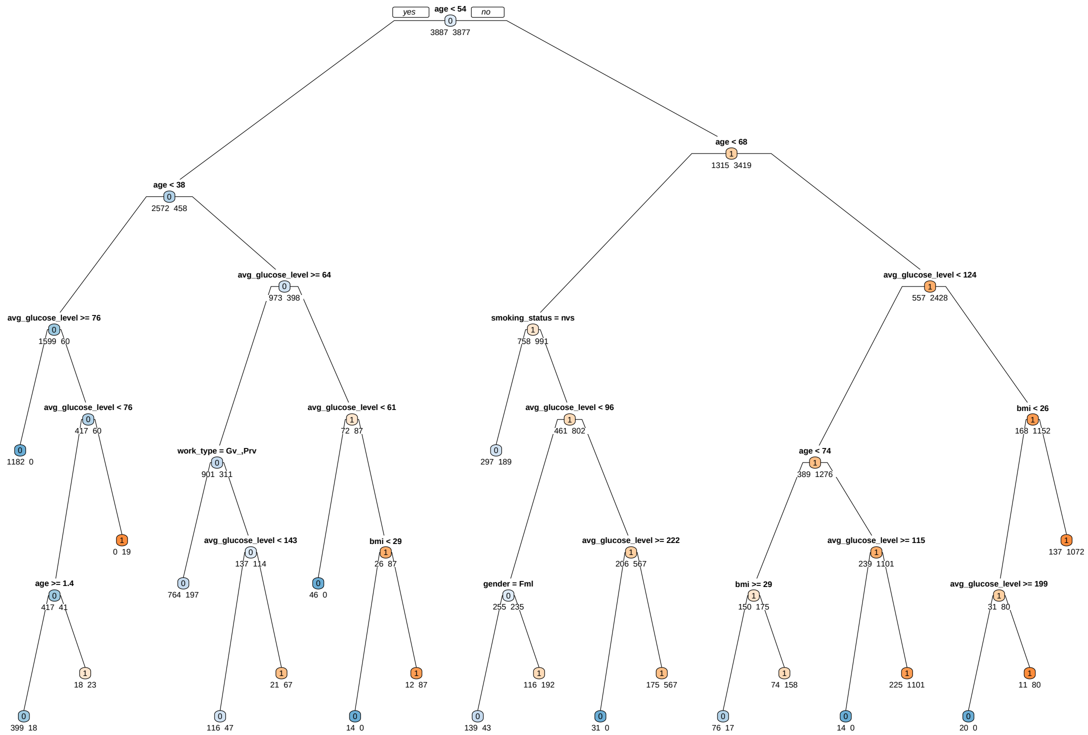
```
dt_over <- rpart(stroke ~ ., data = over_dt, method = "class", cp = 0.00001, maxdepth = 5)

printcp(dt_over)
```

```
##
## Classification tree:
## rpart(formula = stroke ~ ., data = over_dt, method = "class",
##     cp = 1e-05, maxdepth = 5)
##
## Variables actually used in tree construction:
## [1] age               avg_glucose_level bmi               gender
## [5] smoking_status    work_type
##
## Root node error: 3877/7764 = 0.49936
##
## n= 7764
##
##            CP nsplit rel error  xerror      xstd
## 1  0.5426876      0   1.00000 1.03069 0.0113587
## 2  0.0139283      1   0.45731 0.45680 0.0095366
## 3  0.0123807      3   0.42946 0.44828 0.0094733
## 4  0.0079959      5   0.40469 0.41965 0.0092498
## 5  0.0052446      6   0.39670 0.40083 0.0090935
## 6  0.0050727     11   0.36910 0.38200 0.0089294
## 7  0.0036110     14   0.35388 0.36858 0.0088075
## 8  0.0025793     16   0.34666 0.36110 0.0087376
## 9  0.0024503     18   0.34150 0.35440 0.0086737
## 10 0.0012897     20   0.33660 0.34692 0.0086011
## 11 0.0000100     21   0.33531 0.34331 0.0085656
```

```
# Graph 1
prp(dt_over, type = 5, extra = 1, under = FALSE, varlen = 0, fallen.leaves = T, faclen = 50)
```



```r
# Graph 2
prp(dt_over, type = 1, extra = 1, under = TRUE, varlen = 0, roundint = FALSE,  split.font = 2, box.palet
```



```
# Predictions
dt_pred <- predict(dt_over, test, type = "class")
confusionMatrix(dt_pred, test$stroke, positive = "1")
```

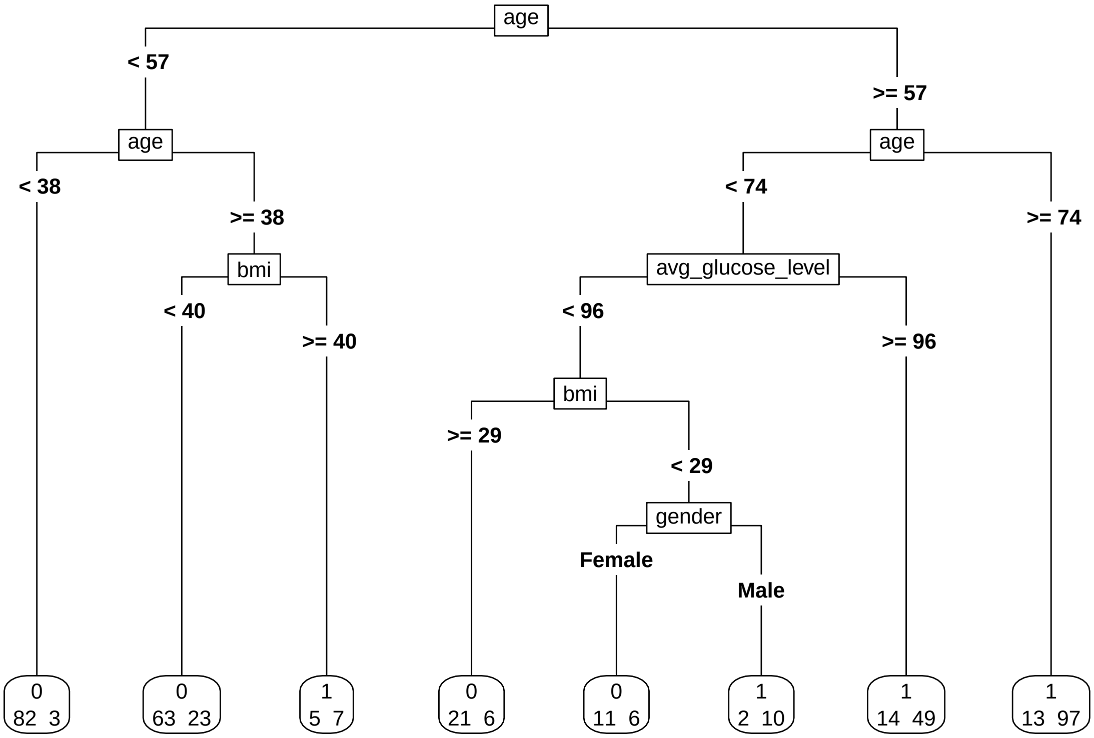
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 754  17
##          1 219  31
##
##                Accuracy : 0.7689
##                  95% CI : (0.7418, 0.7944)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1402
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.64583
##             Specificity : 0.77492
##          Pos Pred Value : 0.12400
##          Neg Pred Value : 0.97795
##              Prevalence : 0.04701
##          Detection Rate : 0.03036
##    Detection Prevalence : 0.24486
##       Balanced Accuracy : 0.71038
##
##        'Positive' Class : 1
##
```

Using Undersampled Train Data

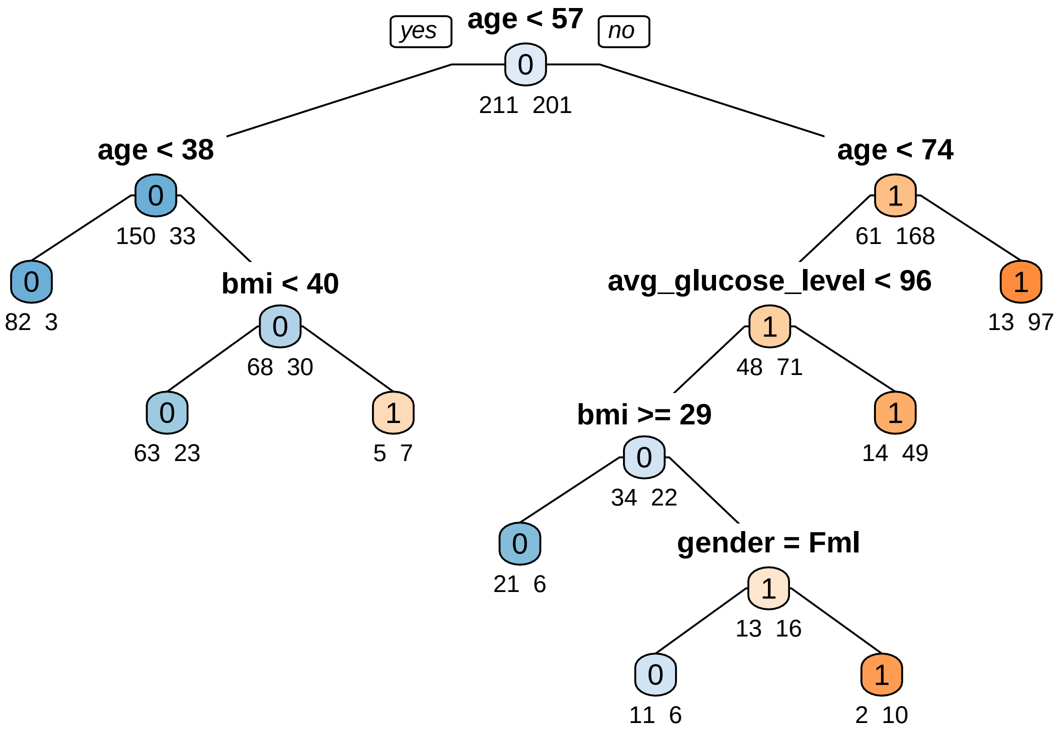
```r
dt_under <- rpart(stroke ~ ., data = under_dt, method = "class", cp = 0.00001, maxdepth = 5)

printcp(dt_under)
```

```
##
## Classification tree:
## rpart(formula = stroke ~ ., data = under_dt, method = "class",
##     cp = 1e-05, maxdepth = 5)
##
## Variables actually used in tree construction:
## [1] age               avg_glucose_level bmi               gender
##
## Root node error: 201/412 = 0.48786
##
## n= 412
##
##          CP nsplit rel error  xerror     xstd
## 1 0.5323383      0   1.00000 1.00000 0.050477
## 2 0.0298507      1   0.46766 0.49254 0.043146
## 3 0.0199005      3   0.40796 0.53234 0.044279
## 4 0.0049751      5   0.36816 0.49254 0.043146
## 5 0.0000100      7   0.35821 0.50249 0.043441
```

```r
# Graph 1
prp(dt_under, type = 5, extra = 1, under = FALSE, varlen = 0, fallen.leaves = T, faclen = 50)
```



```
# Graph 2
prp(dt_under, type = 1, extra = 1, under = TRUE, varlen = 0, roundint = FALSE,  split.font = 2, box.pal
```



```r
# Predictions
dt_pred <- predict(dt_under, test, type = "class")
confusionMatrix(dt_pred, test$stroke, positive = "1")
```

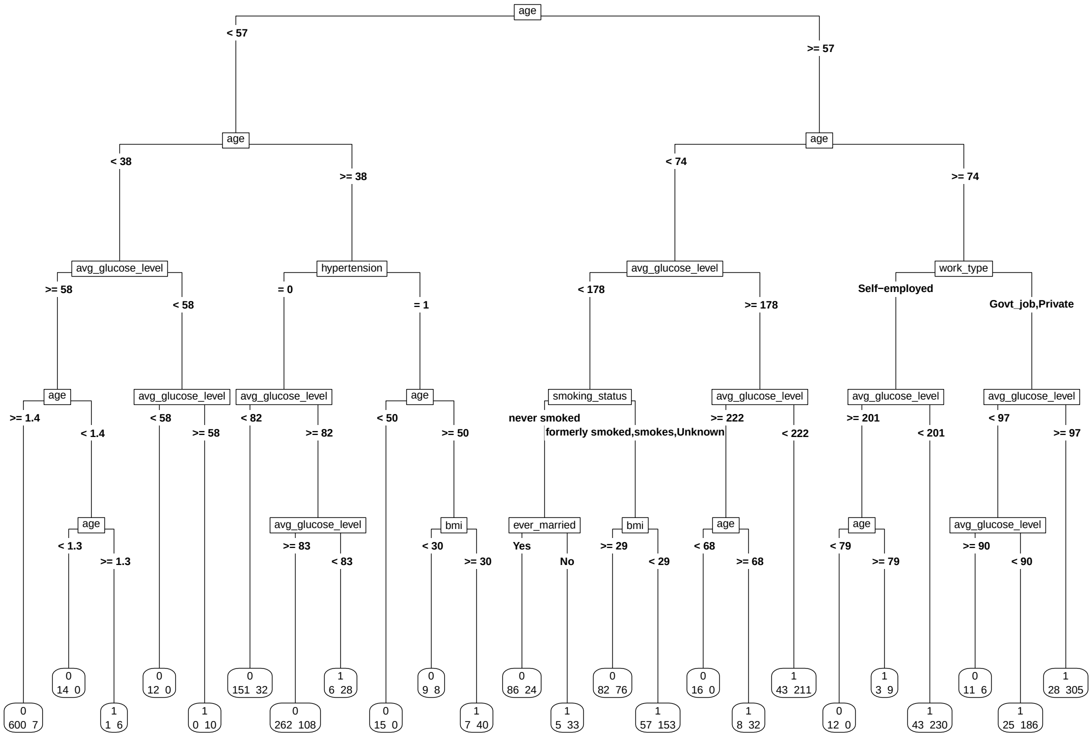
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 714  17
##          1 259  31
##
##                Accuracy : 0.7297
##                  95% CI : (0.7013, 0.7567)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1118
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.64583
##             Specificity : 0.73381
##          Pos Pred Value : 0.10690
##          Neg Pred Value : 0.97674
##              Prevalence : 0.04701
##          Detection Rate : 0.03036
##    Detection Prevalence : 0.28404
##       Balanced Accuracy : 0.68982
##
##        'Positive' Class : 1
##
```

Using Both Oversampled & Undersampled Train Data

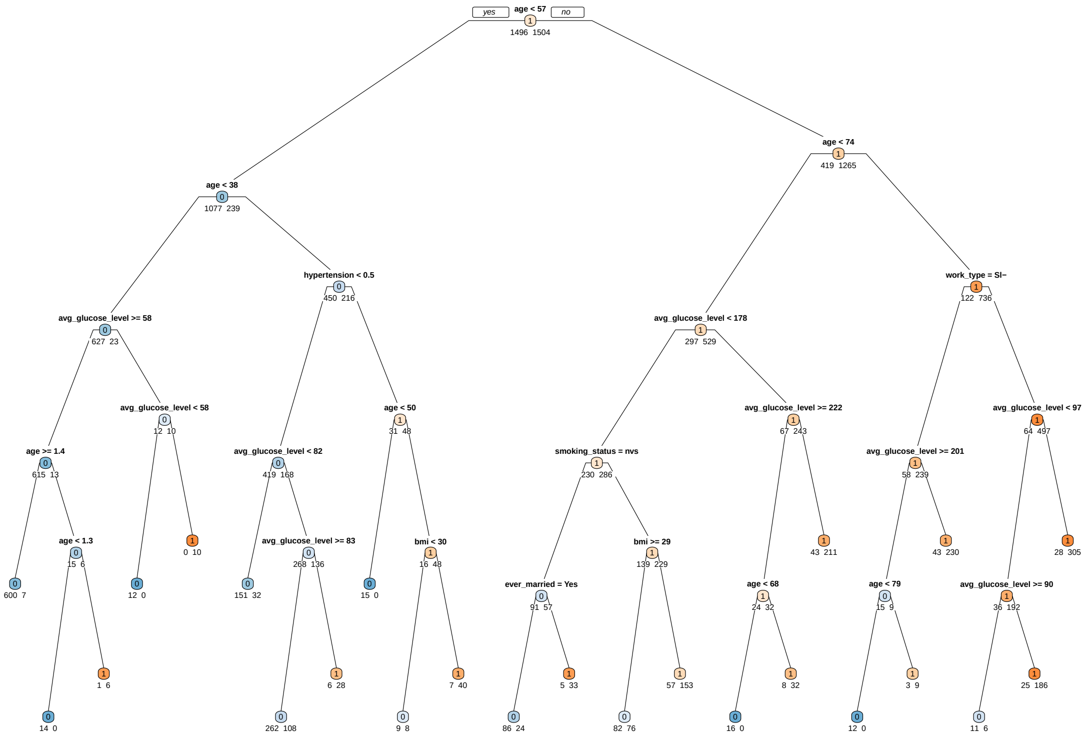
```r
dt_both <- rpart(stroke ~ ., data = both_dt, method = "class", cp = 0.00001, maxdepth = 5)

printcp(dt_both)
```

```
##
## Classification tree:
## rpart(formula = stroke ~ ., data = both_dt, method = "class",
##     cp = 1e-05, maxdepth = 5)
##
## Variables actually used in tree construction:
## [1] age               avg_glucose_level bmi               ever_married
## [5] hypertension      smoking_status    work_type
##
## Root node error: 1496/3000 = 0.49867
##
## n= 3000
##
##            CP nsplit rel error  xerror     xstd
## 1  0.56016043      0   1.00000 1.05615 0.018280
## 2  0.00757576      1   0.43984 0.44719 0.015240
## 3  0.00568182      5   0.39840 0.41845 0.014878
## 4  0.00534759     10   0.36230 0.40842 0.014745
## 5  0.00401070     12   0.35160 0.39706 0.014590
## 6  0.00334225     13   0.34759 0.39639 0.014581
## 7  0.00267380     15   0.34091 0.39572 0.014571
## 8  0.00167112     18   0.33289 0.39439 0.014553
## 9  0.00066845     22   0.32620 0.40174 0.014654
## 10 0.00001000     23   0.32553 0.40374 0.014681
```

```r
# Graph 1
prp(dt_both, type = 5, extra = 1, under = FALSE, varlen = 0, fallen.leaves = T, faclen = 50)
```



```
# Graph 2
prp(dt_both, type = 1, extra = 1, under = TRUE, varlen = 0, roundint = FALSE,  split.font = 2, box.pale
```



```
# Predictions
dt_pred <- predict(dt_both, test, type = "class")
confusionMatrix(dt_pred, test$stroke, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 787  18
##          1 186  30
##
##                Accuracy : 0.8002
##                  95% CI : (0.7743, 0.8243)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1629
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.62500
##             Specificity : 0.80884
##          Pos Pred Value : 0.13889
##          Neg Pred Value : 0.97764
##              Prevalence : 0.04701
##          Detection Rate : 0.02938
##    Detection Prevalence : 0.21156
##       Balanced Accuracy : 0.71692
##
##        'Positive' Class : 1
##
```

Logistic Regression Models

Using Normal Train Data

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

options(scipen = 999)
summary(log_train)
```

```
##
## Call:
## glm(formula = stroke ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1258  -0.3118  -0.1575  -0.0860   3.5099
##
## Coefficients:
##                              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)                 -6.493625   0.807018  -8.046 0.000000000000000852
## genderMale                  -0.086983   0.159418  -0.546              0.58532
## age                          0.078255   0.006685  11.707 < 0.0000000000000002
## hypertension                 0.452715   0.182733   2.477              0.01323
## heart_disease                0.225416   0.218199   1.033              0.30157
## ever_marriedYes             -0.022912   0.264810  -0.087              0.93105
## work_typeGovt_job           -1.505641   0.876466  -1.718              0.08582
## work_typeNever_worked      -10.621315 332.412742  -0.032              0.97451
## work_typePrivate            -1.415983   0.860866  -1.645              0.10000
## work_typeSelf-employed      -1.730302   0.885044  -1.955              0.05058
## Residence_typeUrban          0.076415   0.154529   0.495              0.62095
## avg_glucose_level            0.004162   0.001337   3.114              0.00185
## bmi                         -0.002151   0.013198  -0.163              0.87056
## smoking_statusnever smoked  -0.097092   0.197076  -0.493              0.62225
## smoking_statussmokes         0.205190   0.239324   0.857              0.39124
## smoking_statusUnknown       -0.074264   0.237082  -0.313              0.75410
##
## (Intercept)                ***
## genderMale
## age                        ***
## hypertension               *
## heart_disease
## ever_marriedYes
## work_typeGovt_job          .
## work_typeNever_worked
## work_typePrivate
## work_typeSelf-employed     .
## Residence_typeUrban
## avg_glucose_level          **
## bmi
## smoking_statusnever smoked
## smoking_statussmokes
## smoking_statusUnknown
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1603.0  on 4087  degrees of freedom
## Residual deviance: 1258.1  on 4072  degrees of freedom
## AIC: 1290.1
##
## Number of Fisher Scoring iterations: 14
```

```
log_pred <-predict(log_train, test, type = "response")
confusionMatrix(as.factor(ifelse(log_pred>0.5,1,0)), test$stroke, positive = "1")
```

```
## Warning in confusionMatrix.default(as.factor(ifelse(log_pred > 0.5, 1, 0)), :
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 973  48
##          1   0   0
##
##                Accuracy : 0.953
##                  95% CI : (0.9381, 0.9651)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 0.5383
##
##                   Kappa : 0
##
##  Mcnemar’s Test P-Value : 0.0000000000117
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##          Pos Pred Value :     NaN
##          Neg Pred Value : 0.95299
##              Prevalence : 0.04701
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
##
##        ’Positive’ Class : 1
##
```

Using Oversampled Train Data

```r
log_over <- glm(stroke ~ ., data = over_dt, family ="binomial")

options(scipen = 999)
summary(log_over)
```

```
##
## Call:
## glm(formula = stroke ~ ., family = "binomial", data = over_dt)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.46812  -0.68160  -0.00085   0.69682   2.46218
##
## Coefficients:
##                              Estimate  Std. Error z value             Pr(>|z|)
## (Intercept)                -3.6245815   0.2166875 -16.727 < 0.0000000000000002
## genderMale                 -0.2228696   0.0605619  -3.680             0.000233
## age                         0.0815609   0.0023324  34.968 < 0.0000000000000002
## hypertension                0.4907874   0.0788030   6.228    0.000000000472348
## heart_disease               0.1449883   0.1006799   1.440             0.149841
## ever_marriedYes             0.1723038   0.0990360   1.740             0.081892
## work_typeGovt_job          -2.1211647   0.2397338  -8.848 < 0.0000000000000002
## work_typeNever_worked     -12.6668171 200.8263975  -0.063             0.949708
## work_typePrivate           -1.9694930   0.2296699  -8.575 < 0.0000000000000002
## work_typeSelf-employed     -2.0424221   0.2423702  -8.427 < 0.0000000000000002
## Residence_typeUrban         0.1174146   0.0585871   2.004             0.045059
## avg_glucose_level           0.0040981   0.0005741   7.139    0.000000000000942
## bmi                         0.0072943   0.0047698   1.529             0.126200
## smoking_statusnever smoked -0.2013897   0.0773799  -2.603             0.009252
## smoking_statussmokes        0.2678116   0.0920220   2.910             0.003611
## smoking_statusUnknown      -0.0813441   0.0909810  -0.894             0.371280
##
## (Intercept)                ***
## genderMale                 ***
## age                        ***
## hypertension               ***
## heart_disease
## ever_marriedYes            .
## work_typeGovt_job          ***
## work_typeNever_worked
## work_typePrivate           ***
## work_typeSelf-employed     ***
## Residence_typeUrban        *
## avg_glucose_level          ***
## bmi
## smoking_statusnever smoked **
## smoking_statussmokes       **
## smoking_statusUnknown
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 10763.2  on 7763  degrees of freedom
## Residual deviance:  7319.2  on 7748  degrees of freedom
## AIC: 7351.2
##
## Number of Fisher Scoring iterations: 13
```

```
log_pred <-predict(log_over, test, type = "response")
confusionMatrix(as.factor(ifelse(log_pred>0.5,1,0)), test$stroke, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 713  15
##          1 260  33
##
##                Accuracy : 0.7307
##                  95% CI : (0.7023, 0.7577)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1227
##
##  Mcnemar’s Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.68750
##             Specificity : 0.73279
##          Pos Pred Value : 0.11263
##          Neg Pred Value : 0.97940
##              Prevalence : 0.04701
##          Detection Rate : 0.03232
##    Detection Prevalence : 0.28697
##       Balanced Accuracy : 0.71014
##
##        ’Positive’ Class : 1
##
```

Using Undersampled Train Data

```
log_under <- glm(stroke ~ ., data = under_dt, family ="binomial")

options(scipen = 999)
summary(log_under)
```

```
##
## Call:
## glm(formula = stroke ~ ., family = "binomial", data = under_dt)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1590  -0.7258  -0.1248   0.6732   2.4331
##
```

```
## Coefficients:
##                              Estimate Std. Error z value             Pr(>|z|)
## (Intercept)                 -4.043239   0.995215  -4.063            0.0000485
## genderMale                  -0.321229   0.264253  -1.216              0.22413
## age                          0.097097   0.011469   8.466 < 0.0000000000000002
## hypertension                 0.329700   0.342490   0.963              0.33572
## heart_disease               -0.423881   0.392323  -1.080              0.27995
## ever_marriedYes              0.435716   0.422534   1.031              0.30245
## work_typeGovt_job           -2.878440   1.109151  -2.595              0.00945
## work_typeNever_worked      -12.431341 882.743711  -0.014              0.98876
## work_typePrivate            -3.015887   1.067646  -2.825              0.00473
## work_typeSelf-employed      -2.934567   1.115892  -2.630              0.00854
## Residence_typeUrban          0.015065   0.259901   0.058              0.95378
## avg_glucose_level            0.003319   0.002510   1.322              0.18609
## bmi                          0.020048   0.022765   0.881              0.37851
## smoking_statusnever smoked  -0.306397   0.339969  -0.901              0.36745
## smoking_statussmokes         0.481279   0.394545   1.220              0.22253
## smoking_statusUnknown        0.180448   0.400463   0.451              0.65228
##
## (Intercept)                ***
## genderMale
## age                        ***
## hypertension
## heart_disease
## ever_marriedYes
## work_typeGovt_job          **
## work_typeNever_worked
## work_typePrivate           **
## work_typeSelf-employed     **
## Residence_typeUrban
## avg_glucose_level
## bmi
## smoking_statusnever smoked
## smoking_statussmokes
## smoking_statusUnknown
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 570.91  on 411  degrees of freedom
## Residual deviance: 379.57  on 396  degrees of freedom
## AIC: 411.57
##
## Number of Fisher Scoring iterations: 13
```

```
log_pred <-predict(log_under, test, type = "response")
confusionMatrix(as.factor(ifelse(log_pred>0.5,1,0)), test$stroke, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 707  13
##          1 266  35
##
##                Accuracy : 0.7267
##                  95% CI : (0.6983, 0.7539)
##     No Information Rate : 0.953
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.13
##
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.72917
##             Specificity : 0.72662
##          Pos Pred Value : 0.11628
##          Neg Pred Value : 0.98194
##              Prevalence : 0.04701
##          Detection Rate : 0.03428
##    Detection Prevalence : 0.29481
##       Balanced Accuracy : 0.72789
##
##        'Positive' Class : 1
##
```

Using Both Oversampled & Undersampled Train Data

```
log_both <- glm(stroke ~ ., data = both_dt, family ="binomial")

options(scipen = 999)
summary(log_both)
```

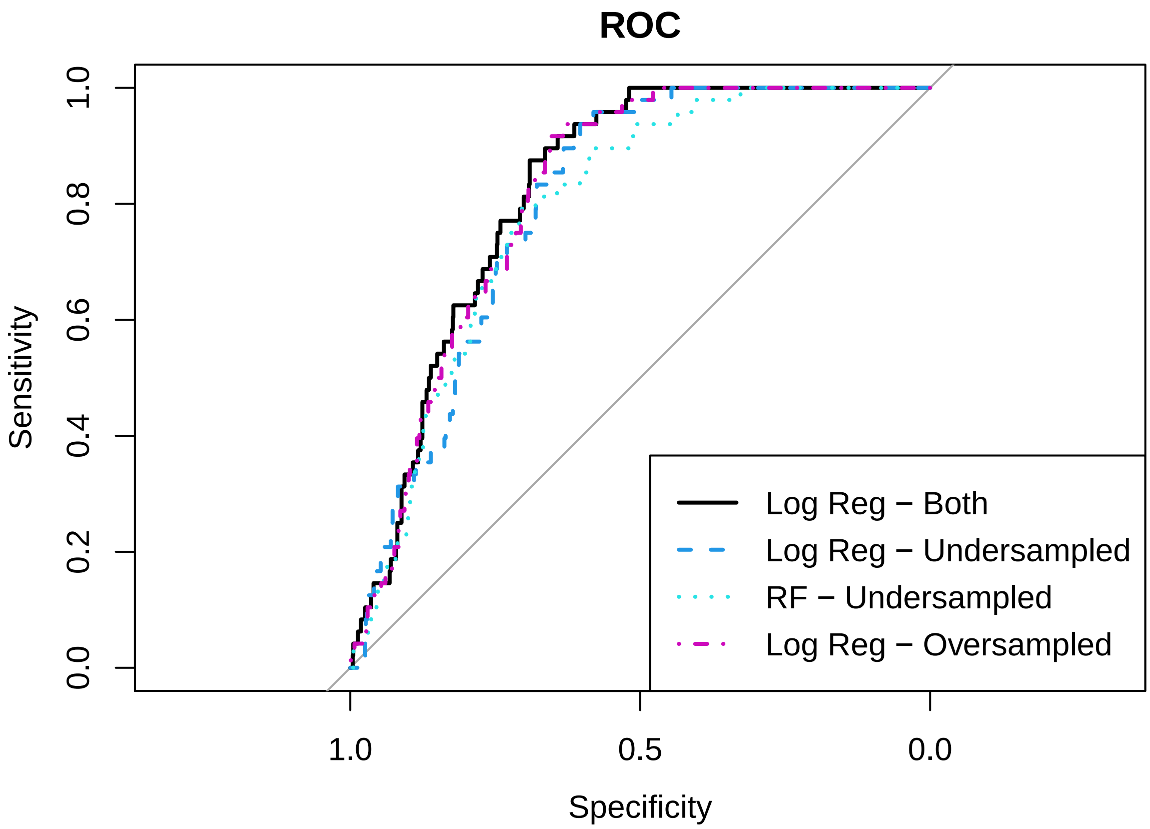
```
##
## Call:
## glm(formula = stroke ~ ., family = "binomial", data = both_dt)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4993  -0.6454   0.2666   0.6953   2.4638
##
## Coefficients:
##                              Estimate  Std. Error z value             Pr(>|z|)
## (Intercept)                -3.5216365   0.3488686 -10.094 < 0.0000000000000002
## genderMale                 -0.1107599   0.0978802  -1.132             0.257808
## age                         0.0838938   0.0038135  21.999 < 0.0000000000000002
## hypertension                0.5200088   0.1294455   4.017          0.000058893
## heart_disease               0.2052352   0.1644893   1.248             0.212137
## ever_marriedYes            -0.0702364   0.1557000  -0.451             0.651917
## work_typeGovt_job          -1.9050073   0.3805868  -5.005          0.000000557
## work_typeNever_worked     -12.6789519 263.9584946  -0.048             0.961689
## work_typePrivate           -1.8194150   0.3631380  -5.010          0.000000544
## work_typeSelf-employed     -1.9424909   0.3850586  -5.045          0.000000454
## Residence_typeUrban         0.1606458   0.0945412   1.699             0.089279
## avg_glucose_level           0.0031560   0.0009355   3.373             0.000743
## bmi                         0.0078643   0.0075075   1.048             0.294855
## smoking_statusnever smoked  -0.4502655   0.1245388  -3.615             0.000300
## smoking_statussmokes         0.1470708   0.1512481   0.972             0.330861
## smoking_statusUnknown       -0.1849266   0.1482655  -1.247             0.212300
##                                
## (Intercept)                ***
## genderMale                    
## age                        ***
## hypertension               ***
## heart_disease                 
## ever_marriedYes               
## work_typeGovt_job          ***
## work_typeNever_worked         
## work_typePrivate           ***
## work_typeSelf-employed     ***
## Residence_typeUrban        .  
## avg_glucose_level          ***
## bmi                           
## smoking_statusnever smoked ***
## smoking_statussmokes          
## smoking_statusUnknown         
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 4158.9  on 2999  degrees of freedom
## Residual deviance: 2794.9  on 2984  degrees of freedom
## AIC: 2826.9
## 
## Number of Fisher Scoring iterations: 13
```

```
log_pred <-predict(log_both, test, type = "response")
confusionMatrix(as.factor(ifelse(log_pred>0.5,1,0)), test$stroke, positive = "1")
```

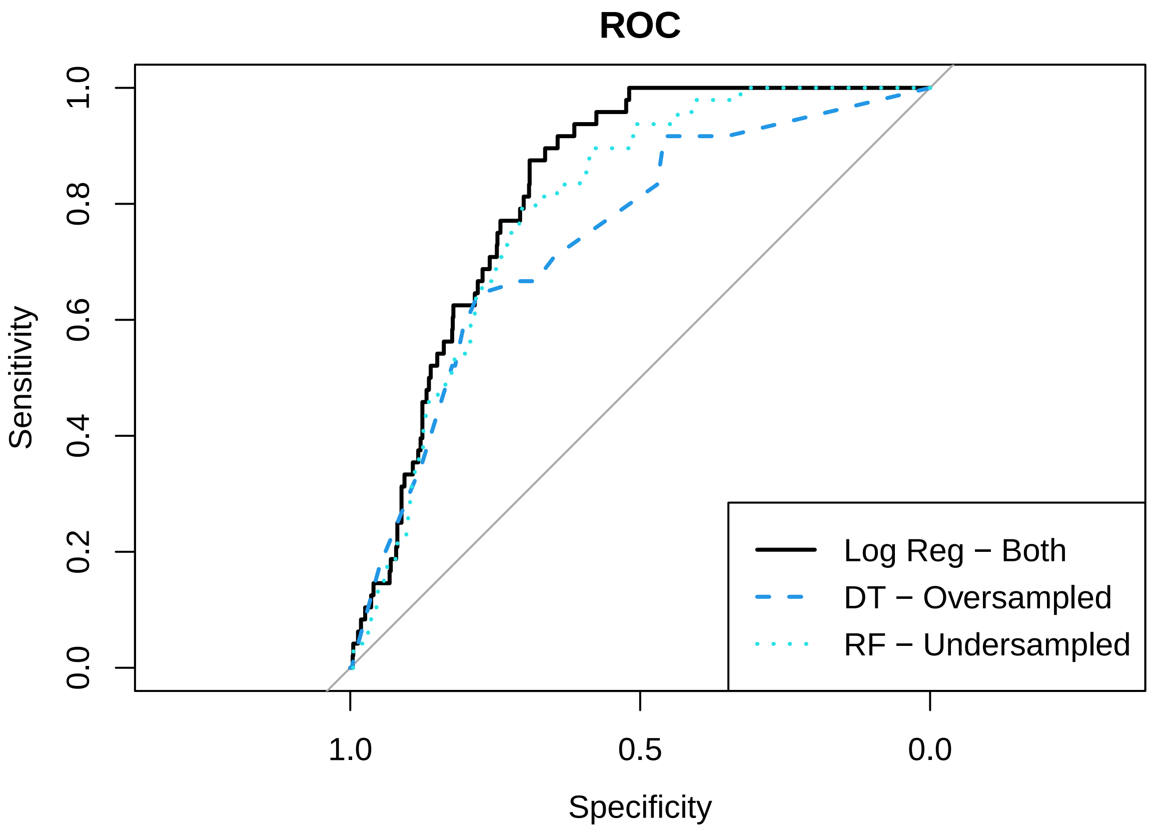
```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 708  11
##          1 265  37
##                                           
##                Accuracy : 0.7297          
##                  95% CI : (0.7013, 0.7567)
##     No Information Rate : 0.953           
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.1418          
##                                           
##  Mcnemar's Test P-Value : <0.0000000000000002
##                                           
##             Sensitivity : 0.77083         
##             Specificity : 0.72765         
##          Pos Pred Value : 0.12252         
##          Neg Pred Value : 0.98470         
##              Prevalence : 0.04701
##          Detection Rate : 0.03624
##    Detection Prevalence : 0.29579
##       Balanced Accuracy : 0.74924
##
##        'Positive' Class : 1
##
```

Comparison Between Models using ROC Curve

```
# Graph 1 (Comparing amongst all models)
# log_both, log_under, under_train, log_over
log_both_pred <- as.data.frame(predict(log_both, test, type = "response"))
log_under_pred <- as.data.frame(predict(log_under, test, type = "response"))
under_train_pred <- as.data.frame(predict(under_train, test, type = "prob"))
log_over_pred <- as.data.frame(predict(log_over, test, type = "response"))

roc_model_log_both <- roc(test$stroke, log_both_pred[,1])
roc_model_log_under <- roc(test$stroke, log_under_pred[,1])
roc_model_under_train <- roc(test$stroke, under_train_pred[,2])
roc_model_log_over <- roc(test$stroke, log_over_pred[,1])

plot(roc_model_log_both, col = 1, lty = 1, main = "ROC")
plot(roc_model_log_under, col = 4, lty = 2, add = TRUE)
plot(roc_model_under_train, col = 5, lty = 3, add = TRUE)
plot(roc_model_log_over, col = 6, lty = 4, add = TRUE)
legend(x = "bottomright",          # Position
       legend = c("Log Reg - Both", "Log Reg - Undersampled", "RF - Undersampled", "Log Reg - Oversample
       lty = c(1, 2, 3, 4),           # Line types
       col = c(1, 4, 5, 6),           # Line colors
       lwd = 2)
```



```
# Graph 2 (Comparing among different approaches)

dt_over_pred <- as.data.frame(predict(dt_over, test, type = "prob"))
roc_model_dt_over <- roc(test$stroke, dt_over_pred[,1])

plot(roc_model_log_both, col = 1, lty = 1, main = "ROC")
plot(roc_model_dt_over, col = 4, lty = 2, add = TRUE)
plot(roc_model_under_train, col = 5, lty = 3, add = TRUE)

legend(x = "bottomright",           # Position
       legend = c("Log Reg - Both", "DT - Oversampled", "RF - Undersampled"),  # Legend texts
       lty = c(1, 2, 3),            # Line types
       col = c(1, 4, 5),            # Line colors
       lwd = 2)
```



Testing the Model

```
test = c('Male', 67, 1, 1, 'Yes', 'Private', "Urban", 228.69, 36, "formerly smoked")
test = data.frame(gender = 'Male', age = 67, hypertension = 1, heart_disease = 1,
                    ever_married = "Yes", work_type = "Private", Residence_type = "Urban",
                    avg_glucose_level = 220.68, bmi = 32, smoking_status = "formerly smoked",
                   ncol =10)
predict(log_both, test, type = "response")
```

```
##         1
## 0.8735638
```

```
ifelse(predict(log_both, test, type = "response")>0.5, 1, 0)
```

```
## 1
## 1
```

“‘