Project Presentation, Analysis Appendix and Combined R-script (Group)
Student groups will have 10 minutes to present their project analysis and findings via a presentation to the class. Groups will present via recorded video in the final week of the quarter (week 10).

The group will submit three deliverables along with giving the presentation of their group's analysis work:

- Presentation slides
- Analysis appendix
- Single combined R-script

The analysis appendix may be an extension of the presentation (slides not presented) or as a separate presentation, pdf, or MS Word document. Details on the contents analysis appendix are found below and in the project rubric.

In the project presentation, groups should clearly explain analyses performed and conclusions reached in their analyses. A sample outline for the presentation along with example questions to address:

1. Purpose: **henry (2 min)**
    - What makes this an interesting setting to study?

Medical setting so it can truly change lives and let us know what things to worry about when it comes to the probability of having a stroke based on certain variables.

    - What is the key forecasting question?

What are the main factors responsible for causing a stroke?

    - Who benefits from the analysis?

Though patients who are at risk of having a stroke and/or those who have already had one are benefitting more directly, everyone of us can benefit from this forecast due to the fact that a stroke can affect anyone if the circumstances are right.

    - What is the anticipated benefit of this modeling exercise?

Letting the patients in the study/data set know what they should work on (from things that can be worked on, such as bmi and not things like age) in order to lower their risk of having a stroke.

2. Hypotheses/Background Research: **ethan terry (2 min)**
    - What did the group expect to find? More technically, what is the implicit status quo or a priori claim before performing any data analysis? What is the research (alternative) hypothesis?
        i. Group Expectations:
            1. Age and hypertension are positively correlated with the likelihood of a stroke
            2. Individuals with a higher BMI are more likely to experience a stroke
            3. Smoking and alcohol consumption increase the chances of a stroke
            4. Individuals with diabetes have a higher risk of stroke
            5. Potential gender differences in stroke rates
            6. Heart conditions are correlated to a higher likelihood of a stroke

ii. Implicit status quo/ priori claim:
    1. Risk factors for stroke such as age, hypertension, smoking, high BMI, and diabetes are known to be associated with an increased likelihood of a stroke
iii. Alternative hypothesis:
    1. Commonly identified risk factors such as age, hypertension, smoking, high BMI, and diabetes are not attributed to an increased likelihood of a stroke.

- What theories, secondary research, experiences, or other factors that inform this analytical setting?
    i. One of the main factors influencing our analytical setting was assumptions on the effects of various medical statuses on the likelihood of a stroke. Since the basis of this information comes from firsthand experience, our group decided to cross reference this information with articles from medical journals and databases in order to formulate an informed analytical setting.

3. Process and Descriptive Statistics: **helena (2 min)**
    - Where did the data set come from? What was the sample size?
        i. Website = Kaggle.com
        ii. Author = Fedesorian
        iii. sample size = 5110 observations
    - What is the dependent variable? What are the variables?
        i. Dependent variable = Stroke
        ii. Independent variables = id, gender, age, hypertension, heart_disease, ever_married, work_type, residence_type, avg_glucose_lvl, bmi
    - What does the data look like (descriptive analytics)?
        i. *data analysis slides from last presentation
    - What kind of cleaning/pre-processing did the group engage in?
        i. Chr → num
            1. Gender → “Female”
            2. Ever_married → “Married
            3. Residence_Type → “Urban”
        ii. Leveld variables → factors
        iii. <u>Remove rows with “N/A”</u>
    - Which statistical techniques did the group choose?
        i. Tables, histograms, box plots, scatterplot, Ggplots
    - Why were these modeling techniques employed?
        i. In order to better visualize the data and understand relationships of independent variables to the dependent variable.

4. Results from Analytical Analysis:(decision trees, logistic regression) **helena henry (2 min)**

- What is the most relevant/interesting model that the team built? Groups could show two predictive models, but the recommendation is against three or more as this will overwhelm the audience.
- What variables are significant? What discoveries or insights resulted from the statistical analysis?
- What do the results mean (in human terms)? Are the results intuitive or surprising? Are there general or larger conclusions from the research?
- How do the results relate to the hypotheses and/or prior research presented earlier? How did "science" move forward because of the work performed?

5. Forecasts/Predictions (using model from 4) **ethan (2 min)**
   - Provide at least one forecast/prediction using a model built in the project
   - What is the associated forecasting error?
     - CT: Prediction error is 0.3080685.
     - Log reg: Associated forecasting error is 0.3048085
   - How does the model perform, that is, how confident is the group in the forecasts?
     - CT: Unfortunately, accuracy is 69.2% using the decision tree model so our group is moderately confident in the performance of the model as a predictor.
     - Log reg: Accuracy slightly higher than CT, but not significant.
   - What considerations might explain the quality of forecasts?
     - CT: Due to high variance often associated with the classification tree method, we decided to reduce the amount of nodes for the classification tree in an effort to reduce variance. This could have led to an increase in error due to the lack of consideration for the other variables as pruning left our model with 4 variables.
     - Log reg: High ratio of error was due to type 1 errors (false positive). This could be due to a limited sample size or other factors such as exclusion of certain variables.

6. Conclusion: **everyone (2 min)**
   - What was the project's purpose? (**Henry**)
   - What were the key findings? What are the practical implications of the conclusions? (**Helena**)
     i.
   - Would additional data/variables have aided the research? (**Ethan**)
   - What are the lessons learned more generally from the work performed in the project? (**Terry**)

The analysis appendix is the project work and results that were not presented. It can be delivered as additional presentation slides or as another document. Items to include: (end of presentation)
APPENDIX DOC LINK
https://docs.google.com/document/d/15zcsT54IgS0wCuHQ7e2wdEmRODXKsFNT5Vxav6woLok/edit?usp=sharing

- Table of contents (**Helena**)
- Data source and references (if available) (**Everyone put one**) R script or outside reference
- Summary of all data manipulations and feature engineering applied to the data (**Henry**)
- Summary of statistical methods used in the research and the analyses performed (Ethan)
- Summary of key observations and results from the statistical models (**Helena**)
- Summary of the conclusions arising from the analysis (**Terry**)
- Lessons learned in the project (**Henry**)

*****APPENDIX GRADING RUBRIC FOR REFERENCE*****

Element	Exceeds Expectations (10)
Context, Question, and Hypothesis	- Clearly delineates project value for the "client" - Provides useful educational background - Provides pre-research hypotheses
Research Method and Descriptive Statistics	- Summarizes data clearly with scales/units, especially dependent variable - Describes data acquisition/manipulation - Describes and justifies research process used - Provides useful descriptive statistics
Analytical Analysis and Results	- Describes impactful statistical results succinctly and insightfully without excessive statistical jargon - Extends methods covered in the course
Forecasts and Implications	- Provides and evaluates one relevant forecast or prediction using models described earlier - Summarize project findings and the broader implications for "client" - Discusses lessons learned useful for class colleagues
Analysis Appendix and Combined R File	- Outline for appendix with summary overview - Summarizes data manipulations and analysis process - Includes additional analyses that are self-explanatory - Combined R file - easy to assimilate
Project Delivery and Media	- Engaging and cohesive learning experience - Improvises delivery with media that supports content well - Clear, relevant, stimulating, professional, and accessible media content

The combined R-script is a single file that incorporates the work of the group performed using R. It should be lightly commented for interpretability.

***** TO DO LIST *****

1. Conclusion
2. Finish slides
3. Paste R Script below
4. Record 2 min vid
    a. Send it to helena
5. Appendix

Replace code -> https://www.browserling.com/tools/text-replace
PASTE R-SCRIPT HERE

Henry

```
dat <- read.csv("healthcare-dataset-stroke-data.csv")
str(dat)
dat$gender <- as.factor(dat$gender)
str(dat)
dat$gender <- 2- as.numeric(dat$gender)
names(dat)[2] <- "Female"
str(dat)
dat$ever_married <- as.factor(dat$ever_married)
dat$ever_married <- 1- as.numeric(dat$ever_married)
dat$Residence_type <- as.factor(dat$Residence_type)
dat$Residence_type <- 2- as.numeric(dat$Residence_type)
dat$work_type <- as.factor(dat$work_type)
str(dat)
# For 'Female', 0 if Male and 1 if Female
# For 'Ever_married', 1 if Yes and o if No
# For 'Residence_type', 1 if Rural and 0 if Urban
dat$bmi <- as.numeric(dat$bmi)
dat$smoking_status <- as.factor(dat$smoking_status)
dat1 <- dat[,2:12]
dat1
str(dat1)
Attempting Regression
set.seed(112233)
s.dat <- subset(dat1, dat1$stroke == 1)
f.dat <- subset(dat1, dat1$stroke == 0)
train.s <- sample(1:nrow(s.dat),105)
```

```
train.f <- sample(1:nrow(f.dat),105)
train.s[1:20]
train.f[1:20]
dat.train <- rbind(s.dat[train.s,],f.dat[train.f,])
str(dat.train)
dim(dat.train)
loan.count.train <- table(dat.train$stroke)
loan.count.train
newfail <- f.dat[-train.f,]
test.f <- newfail[sample(1:nrow(newfail),2350),]
dat.test <- rbind(s.dat[-train.s,],test.f)
loan.count.test <- table(dat.test$stroke)
loan.count.test
mean(dat.test$stroke)
mean(dat1$stroke)
rm(s.dat, f.dat, test.f, newfail)
rm(loan.count.train, loan.count.test)
rm(train.f, train.s, loan.count, loan.perc)
logreg <- glm(stroke~. , data = dat.train,
              family = "binomial")
summary(logreg)
logreg0 <- glm(stroke ~ 1, data = dat.train,
               family = binomial)
summary(logreg0)
logreg.all <- glm(stroke ~ ., data = dat.train,
                  family = binomial)
summary(logreg.all)
anova(logreg, logreg.all, test = "Chisq")
```

Ethan

```
# Read in data
dat <- read.csv("healthcare-dataset-stroke-data.csv")
str(dat)

# Remove id variable (non-predictor)
dat <- dat[,2:12]
dat$bmi <- as.numeric(dat$bmi)
dat1 <- na.omit(dat)
str(dat1)

# Clean data

# Factor conversion
```

```
# work_type - levels: children(1), Govt_job(2), Never_worked(3),
#     Private(4), Self-employed(5)
dat1$work_type <- as.factor(dat1$work_type)

# smoking_status - levels: formerly smoked(1), never smoked(2)
#     smokes(3), unknown(4)
dat1$smoking_status <- as.factor(dat1$smoking_status)

# Numeric conversion
# gender - levels: male(0), female(1)
dat1$gender <- as.factor(dat1$gender)
dat1$gender <- 2- as.numeric(dat1$gender)
names(dat1)[1] <- "Female"

# ever_married - levels: not married(0), married(1)
dat1$ever_married <- as.factor(dat1$ever_married)
dat1$ever_married <- as.numeric(dat1$ever_married) -1
names(dat1)[5] <- "Married"

# Residence_type - levels: urban(0), rural(1)
dat1$Residence_type <- as.factor(dat1$Residence_type)
dat1$Residence_type <- 2- as.numeric(dat1$Residence_type)
names(dat1)[7] <- "Urban"

# Forecasting using logistic regression

# Create training set for prediction

yhat.train <- predict(logreg, dat.train,
                      type = "response")
yhat.train[1:20]

# Check accuracy using actual values
yhat.train.plus.act <- cbind(yhat.train,
                             dat.train$stroke)
yhat.train.plus.act[1:20,]


# Forecast results and test set creation
yhat.train.class <- ifelse(yhat.train > 0.5, 1, 0)
yhat.train.class[1:20]
mean(yhat.train.class != dat.train$stroke)
yhat.test <- predict(logreg, dat.test,
                     type = "response")
```

```
yhat.test[1:20]
yhat.test.class <- ifelse(yhat.test > 0.5, 1, 0)
yhat.test.class[1:20]

# Confusion matrix using test set
tab.lr1.test <- table(dat.test$stroke,
                      yhat.test.class,
                      dnn = c("Actual","Predicted"))
tab.lr1.test
mean(yhat.test.class != dat.test$stroke)

# Compute class errors
class1.test.err <- tab.lr1.test[2,1]/2350
class1.test.err
class0.test.err <- tab.lr1.test[1,2]/2350
class0.test.err
```

Terry

```
# Obs plot
library(ggplot2)
# Create scatter plots without lines
ggplot(dat, aes(x = work_type, y = smoking_status)) +
  geom_jitter(width = 0.2, height = 0) +
  labs(x = "Work Type", y = "Smoke Status") +
  ggtitle("Observation Plot: Work Type vs Smoke Status")

# Create scatter plots for subsets of data
subset1 <- dat[1:149, ]
subset2 <- dat[150:300, ]

ggplot() +
  geom_jitter(data = subset1, aes(x = work_type, y = smoking_status), color = "blue", width =
0.2, height = 0) +
  geom_jitter(data = subset2, aes(x = work_type, y = smoking_status), color = "red", width = 0.2,
height = 0) +
  labs(x = "Work Type", y = "Smoke Status") +
  ggtitle("Observation Plot: Work Type vs Smoke Status (Subsets)")


#Scatterplot with regression line
ggplot(dat, aes(x = avg_glucose_level, y = bmi)) +
  geom_point(color = "steelblue") +
```

```
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(x = "Average Glucose Level", y = "BMI") +
  ggtitle("Scatter Plot: Average Glucose Level vs BMI with Regression Line") +
  theme_minimal()

#Boxplot
dat$ever_married <- as.factor(dat$ever_married)
levels(dat$ever_married) <- c("Married", "Not Married")
ggplot(dat, aes(ever_married, age)) +
  geom_boxplot(color = "black", fill = 'steelblue') +
  labs(x = "Ever_married", y = "Age", title ="Marital Status vs Age") +
  theme_minimal()
```

Helena

```
# Read in data
dat <- read.csv("healthcare-dataset-stroke-data.csv")
str(dat)

# Remove id variable (non-predictor)
dat <- dat[,2:12]
str(dat)

# Factor conversion
# work_type - levels: children(1), Govt_job(2), Never_worked(3),
#     Private(4), Self-employed(5)
dat$work_type <- as.factor(dat$work_type)

# smoking_status - levels: formerly smoked(1), never smoked(2)
#     smokes(3), unknown(4)
dat$smoking_status <- as.factor(dat$smoking_status)

# Numeric conversion
# gender - levels: male(0), female(1)
dat$gender <- as.factor(dat$gender)
dat$gender <- 2- as.numeric(dat$gender)
names(dat)[1] <- "female"

# ever_married - levels: not married(0), married(1)
dat$ever_married <- as.factor(dat$ever_married)
dat$ever_married <- as.numeric(dat$ever_married) -1
names(dat)[5] <- "married"
```

```
# Residence_type - levels: urban(0), rural(1)
dat$Residence_type <- as.factor(dat$Residence_type)
dat$Residence_type <- 2- as.numeric(dat$Residence_type)
names(dat)[7] <- "rural_residence"

# bmi - gives a warning due to NA values (non-numeric) but can be ignored
dat$bmi <- as.numeric(dat$bmi)

# Clean data
colSums(is.na(dat))
dat <- na.omit(dat)
colSums(is.na(dat))
dat <- dat[,c(11,1:10)]
str(dat)

# Linear regression (Correlation - predictor/response)
#       (omits 1 variable from factor) (obs deleted due to NA values)
reg <- lm(stroke ~ . , data = dat)
summary(reg)

# Graphs - observations between variables

# Table - Stroke vs No Stroke
tab1 <- table(dat$stroke)
tab1
# Bar Graph - Stroke vs No Stroke
barplot(tab1, main = "Likelihood of Stroke",
        xlab = "Number of Persons", ylab = "Stroke Status",
        horiz = T, col = c("lightblue", "orange"))

# Table - Heart Disease vs Stroke
tab2 <- table(dat$heart_disease, dat$stroke)
tab2
# Bar Graph - Heart Disease vs Stroke Status
barplot(tab2, main = "Occurance of Heart Disease in Stroke Patients",
        xlab = "Heart Disease", ylab = "Stroke",
        col = c("lightblue", "orange"),
        horiz = T, legend = T)

# Table - Marriage vs Stroke
tab3 <- table(dat$married, dat$stroke)
tab3
# Bar Graph - Marriage vs Stroke
barplot(tab3, main = "Occurance of Stroke in Marriages",
```

```
        xlab = "Marital Status", ylab = "Stroke",
        col = c("lightblue", "orange"),
        horiz = T, legend = T)

# Table - Hypertension vs Stroke
tab4 <- table(dat$hypertension, dat$stroke)
tab4
# Bar Graph - Hypertension vs Stroke
barplot(tab4, main = "Likelihood of Hypertension in Stroke Patients",
        xlab = "Hypertension", ylab = "Stroke",
        col = c("lightblue", "orange"),
        horiz = T, legend = T)

# Conditional Box Plot - Age vs Stroke
boxplot(age~stroke, data = dat, main = "Stroke Status Dependent on Age",
        col = c("lightblue", "orange"), horizontal=T)

# Conditional Box Plot - Avg Glucose Level vs Stroke
boxplot(avg_glucose_level~stroke, data = dat, main = "Avg Glucose Level in Stroke Patients",
        col = c("lightblue", "orange"), horizontal=T,
        xlab = "Average Glucose Level (mg/dL)")

# 5x2 Table - Work Type vs Stroke Status
tab5 <- table(dat$work_type, dat$stroke)
tab5

# 4x2 Table - Smoking Status vs Stroke Status
tab6 <- table(dat$smoking_status, dat$stroke)
tab6

library(ggplot2)

# Correlation - Age vs Stroke
ggplot(dat, aes(x = age, y = stroke)) +
  xlab("Age") +
  ylab("Stroke") +
  ggtitle("Stroke Rate as Age Increases") +
  geom_point() +
  geom_smooth(method = "gam", se = TRUE) +
  geom_point(aes(col=stroke))

# Correlation - Age vs Stroke for various Work Types
ggplot(dat, aes(x = stroke, y = age)) +
  xlab("Stroke") +
```

```
  ylab("Age") +
  ggtitle("Stroke Rate as Age Increases for various Work Types") +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  facet_grid(.~work_type) +
  geom_point(aes(col=stroke))

# Set up training and test data

loan.count <- table(dat$stroke)
loan.count

# 209 observations of 4700 are positive for having a stroke

set.seed(112233)

datarows.1 <- subset(dat, dat$stroke == 1)
datarows.0 <- subset(dat, dat$stroke == 0)
train.1 <- sample(1:nrow(datarows.1),105)
train.0 <- sample(1:nrow(datarows.0),105)

dat.train <- rbind(datarows.1[train.1,],datarows.0[train.0,])
leftoverrows.0 <- datarows.0[-train.0,]
rowsfortest.0 <- leftoverrows.0[sample(1:nrow(leftoverrows.0),2349),]
dat.test <- rbind(datarows.1[-train.1,],rowsfortest.0)

#  Perform some checks

table(dat.train$stroke)
table(dat.test$stroke)
table(dat$stroke)
mean(dat.train$stroke)
mean(dat.test$stroke)
#
#  Train set is 50/50
#  Test set is same ratio as original data set
#
rm(datarows.0, datarows.1, leftoverrows.0)
rm(train.0, train.1, rowsfortest.0)
rm(loan.count)

# Convert the Y variable to a factor

dat.train[,1] <- as.factor(dat.train[,1])
```

```
dat.test[,1] <- as.factor(dat.test[,1])

#  Build the first classification tree

library(tree)
tree1 <- tree(stroke ~ ., data = dat.train)
summary(tree1)

# Plot the tree

plot(tree1)
text(tree1, pretty = 0)

tree1

#  Use the tree to make predictions on the training
#    and test data

tree.pred.tr <- predict(tree1, dat.train, type = "class")
table(dat.train$stroke, tree.pred.tr,
     dnn = c("Actual", "Predicted"))
err.tr1.train <- mean(dat.train$stroke != tree.pred.tr)
err.tr1.train

#Let's try the "trained" tree
#    on the test data

tree.pred.tst <- predict(tree1, dat.test, type = "class")
table(dat.test$stroke, tree.pred.tst,
     dnn = c("Actual", "Predicted"))
err.tr1 <- mean(dat.test$stroke != tree.pred.tst)
err.tr1

# The testing data has greater error than the training set, the model is
# over-fit to the training data leading to high variance

# prune the tree

prune1 <- prune.misclass(tree1)
names(prune1)

plot(prune1)
plot(prune1$size, prune1$dev, xlab = "Size of Tree",
    ylab = "Number Misclassified")
```

```
# the best tree seems to be of size 2

prune.tree1 <- prune.misclass(tree1, best = 2)
summary(prune.tree1)
prune.tree1
plot(prune.tree1)
text(prune.tree1, pretty = 0)

# compare pruned tee too original tree

pt1.pred <- predict(prune.tree1, dat.test, type = "class")
table(dat.test$stroke, pt1.pred,
     dnn = c("Actual", "Predicted"))
err.pt1 <- mean(dat.test$stroke != pt1.pred)
err.pt1

err.tr1

#  The pruned tree is slightly better




#  Cross-validation of Classification Trees
#
#  Run cross-validation the first tree fit to
#    see if there is a lower variance model

#    This uses k = 10-fold cross-validation
#
set.seed(445566)
#
tree1 <- tree(stroke ~ ., data = dat.train)
cv.tree <- cv.tree(tree1, FUN = prune.misclass, K = 10)
cv.tree
#
#  Plot the results
#
?cv.tree
plot(cv.tree)
plot(cv.tree$size, cv.tree$dev, main = "Size vs Deviance",
    xlab = "Tree Size", ylab = "# Misclassified")
#
```

```
#  From the plots, choose the minimum number of nodes
#    Here it appears to be 2

prune.tree2 <- prune.tree(tree1, best = 2)
plot(prune.tree2)

text(prune.tree2, pretty = 0)
#
#  Now using this new pruned tree, we predict on
#    the test data
#
pt2.pred <- predict(prune.tree2, dat.test, type = "class")
table(dat.test$stroke, pt2.pred,
     dnn = c("Actual", "Predicted"))
err.pt2 <- mean(dat.test$stroke != pt2.pred)
err.pt2
err.pt1

# The error from pruning is the same as the error from cross validation pruning




# Bootstrap aggregation

library(randomForest)
names(dat.train)
set.seed(223344)
bag.10 <- randomForest(stroke ~ .,
                       data = dat.train,
                       mtry = 10, ntree = 10,
                       importance = TRUE)
bag.10
#
#  Now we make predictions on the test data and compute
#    error
#
pred.bag.10 <- predict(bag.10, dat.test)
tab.bag.10 <- table(dat.test$stroke, pred.bag.10)
tab.bag.10
err.bag10 <- mean(dat.test$stroke != pred.bag.10)
err.bag10
#
#  How did we do?
```

```
#
#  Now we compute a bagged model with 25 trees
#
bag.25 <- randomForest(stroke ~ ., data = dat.train,
                       mtry = 10, ntree = 25,
                       importance = TRUE)
bag.25
pred.bag.25 <- predict(bag.25, dat.test)
tab.bag.25 <- table(dat.test$stroke, pred.bag.25)
tab.bag.25
err.bag25 <- mean(dat.test$stroke != pred.bag.25)
err.bag25
#
#  Again, how did the classifier perform on the test data
#
#  Now we compute a bagged model based on 100 trees
#
bag.100 <- randomForest(stroke ~ ., data = dat.train,
                        mtry = 10, ntree = 100,
                        importance = TRUE)
bag.100
pred.bag.100 <- predict(bag.100, dat.test)
tab.bag.100 <- table(dat.test$stroke, pred.bag.100)
tab.bag.100
err.bag100 <- mean(dat.test$stroke != pred.bag.100)
err.bag100
#
#  Maybe a bit better...
#
#  Now compute a bagged model based on 1000 trees
#
bag.1000 <- randomForest(stroke ~ ., data = dat.train,
                         mtry = 10, ntree = 1000,
                         importance = TRUE)
bag.1000

pred.bag.1000 <- predict(bag.1000, dat.test)
tab.bag.1000 <- table(dat.test$stroke, pred.bag.1000)
tab.bag.1000
err.bag1000 <- mean(dat.test$stroke != pred.bag.1000)
err.bag1000
#
#  Compare the classification errors from the bagged models
#
```

```
err.bag10
err.bag25
err.bag100
err.bag1000

# the smallest error was in the model bagging 10 trees




# Random Forests
library(randomForest)

rf.1 <- randomForest(stroke ~ ., data = dat.train,
                     mtry = 4, ntree = 1000,
                     importance = TRUE)
rf.1

#
#  Now make predictions on the test data and compute the MSE
#
pred.rf <- predict(rf.1, dat.test)
tab.rf <- table(dat.test$stroke, pred.rf)
tab.rf
err.rf <- mean(dat.test$stroke != pred.rf)
err.rf
#
#  Take a look at the important variables
#
importance(rf.1)
varImpPlot(rf.1, main = "Variable Importance Plot")
#
#  The top variables in terms of improving accuracy can be
#    provided
#
imp.rf.1 <- importance(rf.1)
imp.rf.1[order(imp.rf.1[,4], decreasing = TRUE),]
sort(imp.rf.1[,3], decreasing = TRUE)
sort(imp.rf.1[,4], decreasing = TRUE)
#



str(dat.train$stroke)
```

```
#
#  Change the Y variable back to quantitative 0/1
#
dat.train$stroke <- as.numeric(dat.train$stroke) - 1
dat.test$stroke <- as.numeric(dat.test$stroke) - 1
#
#  Run Gradient Boosting
#  Build a final tree in sequence, by fitting a shallow
#    tree and then examining the errors.  The next tree
#    is fitted on the errors and "averaged" in with the
#    first (current) tree.  This process is repeated
#    The averages are computed using the "learning"
#    parameter, "shrinkage"
library(gbm)
#
boost.1 <- gbm(stroke ~ ., data = dat.train,
             distribution = "bernoulli", n.trees = 1000,
             interaction.depth = 2, shrinkage = 0.005)
summary(boost.1)
#
#  The top 10 variables are the usual suspects
#
#  Make predictions on the test set
#
pred.bst1 <- predict(boost.1, dat.test,
                  n.trees= 1000, type = "response")
pred.bst1
pred.bst1.cl <- ifelse(pred.bst1 > 0.5, 1, 0)
tab.bst1 <- table(dat.test$stroke, pred.bst1.cl)
tab.bst1
err.bst1 <- mean(dat.test$stroke != pred.bst1.cl)
err.bst1
#
#  Boosting did not work as well as some of the earlier
#    ensemble methods
#



#
#  Compare errors on the test data on all models
#
#  Compute logistic regression model on all variables for a
#    benchmark
#
```

```
lreg <- glm(stroke ~ ., data = dat.train,
            family = binomial)
summary(lreg)
pred.lr <- predict(lreg, dat.test, type = "response")
pred.lr.cl <- ifelse(pred.lr > 0.5, 1, 0)
err.lr <- mean(dat.test$stroke != pred.lr.cl)
err.lr
#
#  Compare errors
#
#  First the straight-up tree building routines
#
err.tr1
err.pt1

err.pt2
#
#  Now Bagging
#
err.bag10
err.bag25
err.bag100
err.bag1000
#
#  Now Random Forests and Boosting
#
err.rf
err.bst1
#
#  Finally logisitic regression
#
err.lr
#
#  Which classifier worked the best? bag 10
#
```