CSIS734-01 Data Mining & Predictive Analytics
Garth Mortensen, mort0052@stthomas.edu

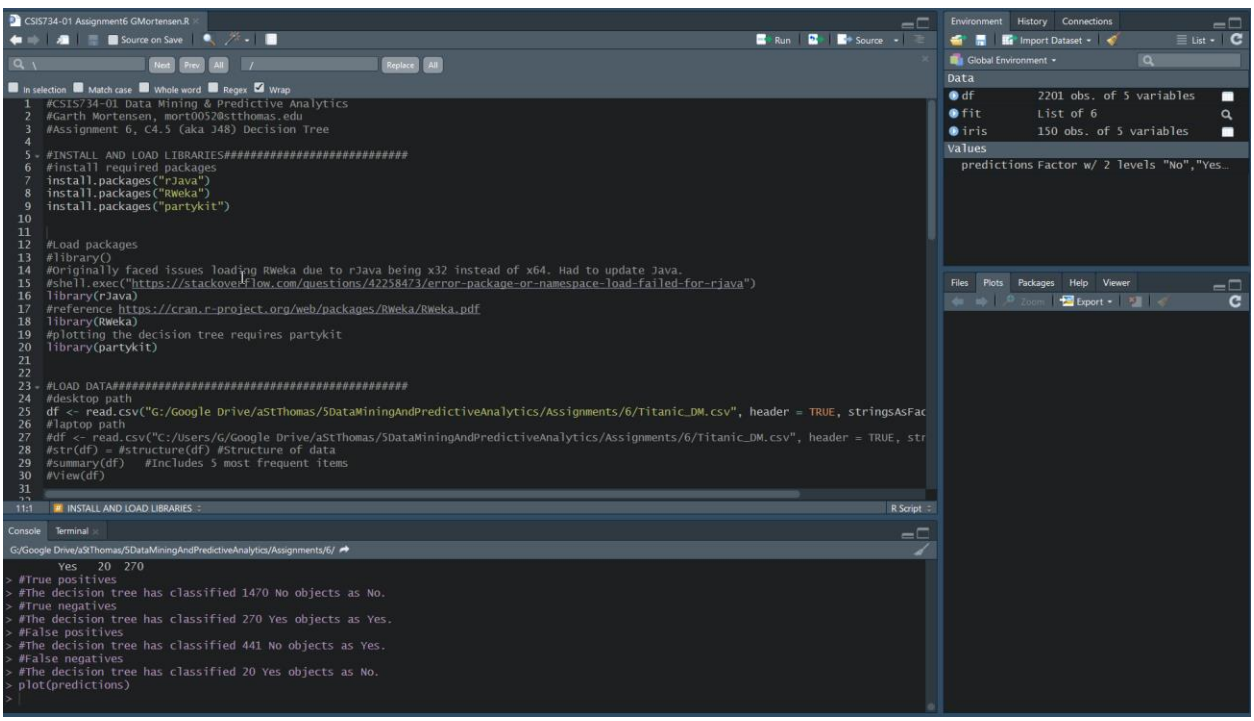
Graduate Program in Software
CSIS 734-01: Data Mining & Predictive Analytics
Assignment #6 (100 points)
Due Date: March 24th, 2018

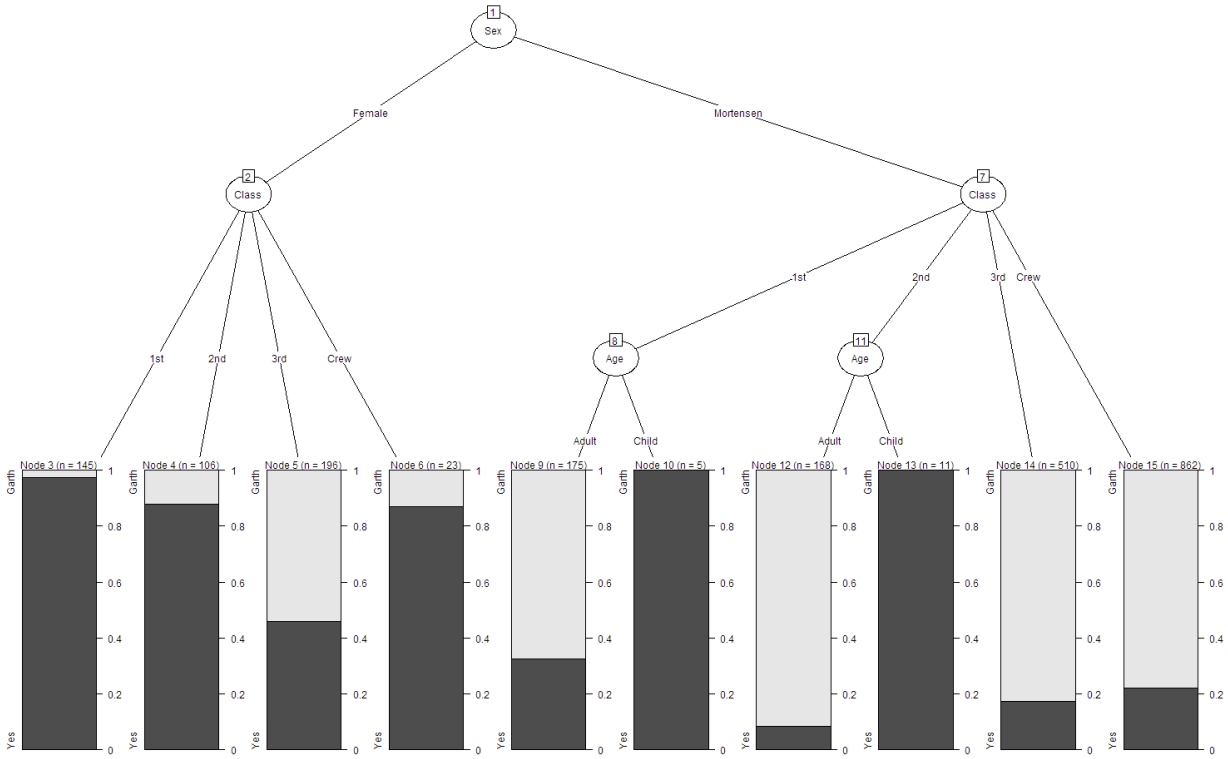
Use the "titanic_DM.csv" file on the Canvas for this assignment.

1. **Carefully examine the dataset** and select meaningful attributes from ALL the data records to build a C4.5 decision tree based on your updated Titanic dataset. No other constraints will be placed on the tree-building process. Your C4.5 decision tree is to explain why passengers survived.

C4.5 (aka J48) = Information Gain (Gain Ratio) is splitting criteria. Post-pruning, error-based pruning. Good for small datasets.

2. Produce your decision tree.





3. Submit a hardcopy of your answers (i.e. a table) and screenshots for this assignment on the due date. Your hardcopy must be as clear as possible. **Anything that cannot be read won't be graded!**

4. Please staple all pages of your submission together! Instructor is not responsible for missing pages if your submission is not stapled together.

5. Please also submit your program (**no screenshot please!**) to clai@stthomas.edu.

```
#CSIS734-01 Data Mining & Predictive Analytics

#Garth Mortensen, mort0052@stthomas.edu

#Assignment 6, C4.5 (aka J48) Decision Tree


#INSTALL AND LOAD LIBRARIES############################

#install required packages

install.packages("rJava")

install.packages("RWeka")

install.packages("partykit")



#Load packages

#library()

#Originally faced issues loading RWeka due to rJava being x32 instead of x64. Had to update Java.

#shell.exec("https://stackoverflow.com/questions/42258473/error-package-or-namespace-load-failed-
for-rjava")

library(rJava)

#reference https://cran.r-project.org/web/packages/RWeka/RWeka.pdf

library(RWeka)

#plotting the decision tree requires partykit

library(partykit)



#LOAD DATA##############################################

#desktop path

df <- read.csv("G:/Google
Drive/aStThomas/5DataMiningAndPredictiveAnalytics/Assignments/6/Titanic_DM.csv", header = TRUE,
stringsAsFactors = TRUE)

#laptop path
```

```
#df <- read.csv("C:/Users/G/Google Drive/aStThomas/5DataMiningAndPredictiveAnalytics/Assignments/6/Titanic_DM.csv", header = TRUE, stringsAsFactors = TRUE)

#str(df) = #structure(df) #Structure of data

#summary(df)   #Includes 5 most frequent items

#View(df)


#First column to categorical

#categorical variables are called factors in R.

#Because I'm loading categorical variables, I need to first convert them.

#R's default behavior when creating data frames is to convert all characters into factors.

#But column one is integer.

df$Passenger <- as.factor(df$Passenger)

#column 1 is now compatible.



#FIT MODEL##########################################

#This appears to be a complete tutorial on C4.5 decision tree and end visualization.

#On C4.5   http://data-mining.business-intelligence.uoc.edu/home/j48-decision-tree

#Working example! http://data-mining.business-intelligence.uoc.edu/home/j48-decision-tree


#The C4.5 algorithm is an extension of the ID3 algorithm and constructs a decision tree to

#maximize information gain (difference in entropy).

#The following demonstrates the C4.5 (called J48 in Weka) decision tree method.

#As seen at https://cran.r-project.org/web/packages/RWeka/RWeka.pdf

#J48 generates unpruned or pruned C4.5 decision trees (Quinlan, 1993).


fit <- J48(Survived~., data=df) #Must use ~.
```

```
png(file="DecisionTreePlot.png",width=1500,height=1000) #Prep for image export
plot(fit)
dev.off() #turn off so the file can save.
#of course...I can't figure out how to re-enable plots.

#Summarize the fit. this also displays confusion matrix
summary(fit)
#Otherwise, an improved confusion matrix,
#where predictions is for column headers and actual is for row headers
predictions <- predict(fit, df)
table(predictions, df$Survived)
#True positives
#The decision tree has classified 1470 No objects as No.
#True negatives
#The decision tree has classified 270 Yes objects as Yes.
#False positives
#The decision tree has classified 441 No objects as Yes.
#False negatives
#The decision tree has classified 20 Yes objects as No.
#plot(predictions)


#CLEAN UP##############################################
rm(list = ls())  #Clear workspace
dev.off() #Clear plots
cat("\014")  #Clear console (ctrl+L)
```