Assignment 4

Stat 515

Saurabh Rao Donthineni

A .

Overall proportion of web robot can be calculated by :

Total number of web robot : 449

Total number of observations : 935

Proportion of web robots : 449/935 = 0.4802

= 48.02 % .

B .

By setting the cutoff level at 0.5, the classification table is as follows :

	Reference	
Prediction	F	T
	F 167	102
	T 25	79

The accuracy rate is 0.66

The misclassification error value is 0.34

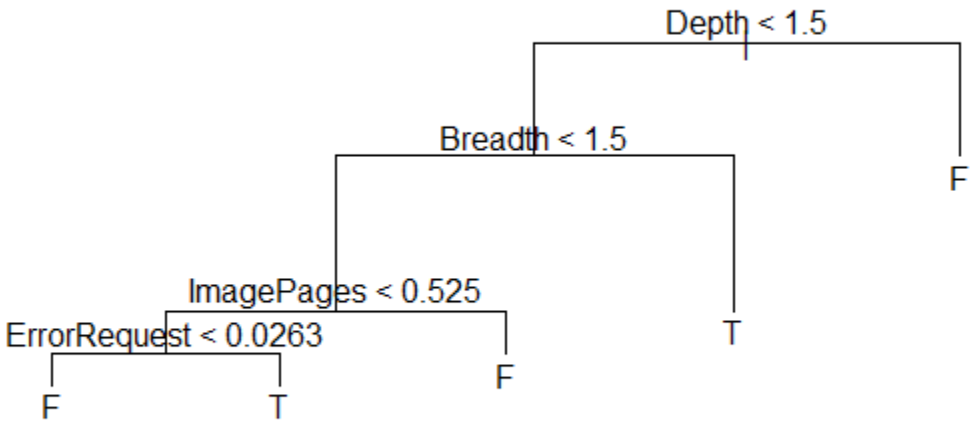
```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)     1.811e+05  3.564e+07   0.005   0.9959
ImagePages     -1.868e+00  4.052e-01  -4.610 4.03e-06 ***
RepeatedAccess  1.536e+00  1.077e+00   1.426   0.1537
ErrorRequest    8.761e-01  3.523e-01   2.487   0.0129 *
GET            -1.811e+05  3.564e+07  -0.005   0.9959
POST           -1.811e+05  3.564e+07  -0.005   0.9959
HEAD           -1.811e+05  3.564e+07  -0.005   0.9959
Breadth         1.457e+00  3.048e-01   4.781 1.74e-06 ***
Depth          -1.917e+01  7.181e+02  -0.027   0.9787
```

From the above table , the significant predictors are ImagePages, ErrorRequest and Breadth. The logistic regression is :

Robot = -1.868(ImagePages) + 0.876(ErrorRequest) + 1.457 (Breadth)

C .

The classification tree is :



The accuracy is 0.62030
Misclassification error value is 0.38
The classification table is :

tree.pred	F	T
F	169	118
T	23	63

Gini Index for top node :

Gini Index value = (515/562)(1-(0.48*0.48+0.51*0.51))+(48/562)(1-1*1)

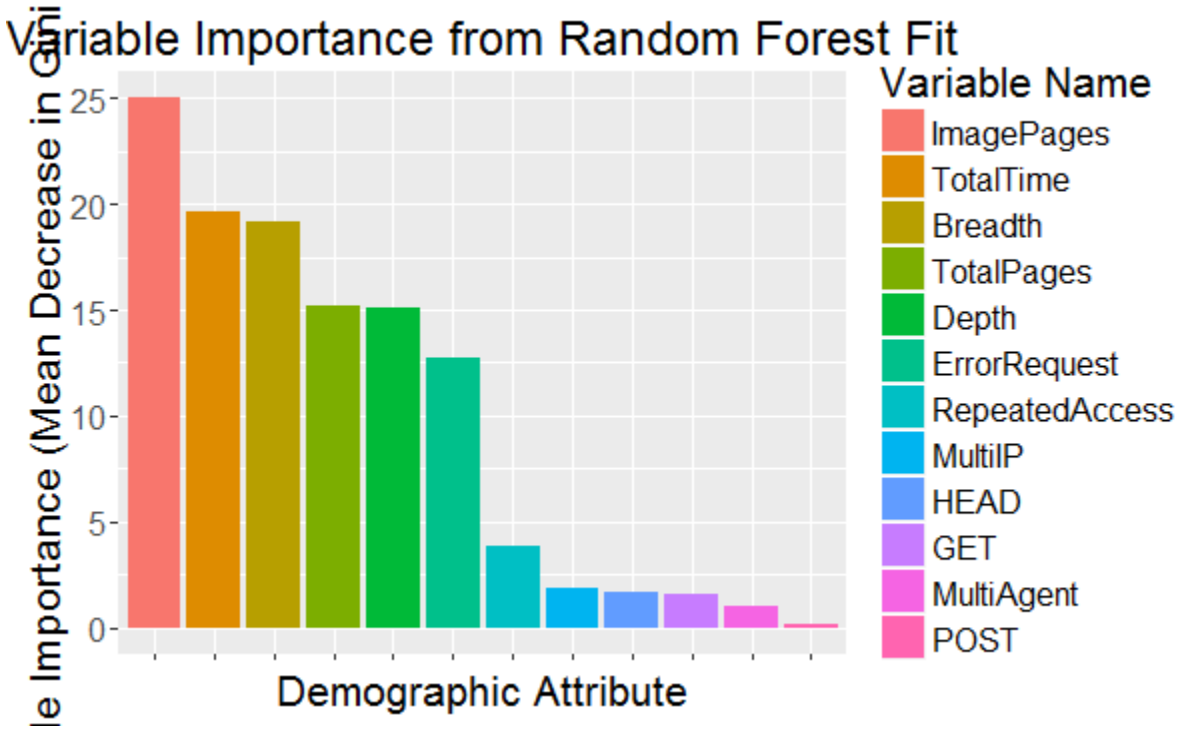
= 0.4753

Overall Gini split value is 0.354

D.

Given that Depth=1, Breadth=1, ImagePages=0.75.
As the ImagePages value is greater than 0.5 , it will go to the right of the tree.
We can therefore say that this is not a robot based on the tree that have obtained.

E.



Top 3 important variables are ImagePages, Breadth and TotalTime
Classification table :

predict_rf	F	T
F	167	105
T	25	76

Accuracy value is 0.652

Misclassification value is 0.348

F.

Misclassification error values are :

Question B: 0.3422

Question C: 0.38

Question E: 0.3476

Therefore , the backward model used in question B is the best.

Appendix

```
install.packages('caret')

library(caret)

install.packages('tree')

library(tree)

install.packages('dplyr')

library(dplyr)

install.packages('ggplot2')

library(ggplot2)

install.packages('randomForest')

library(randomForest)



# clear existing variables from global environment
```

```
rm(list = ls())

# read file using read.csv

mydata=read.csv("C:\\Users\\SOURAV\\Desktop\\Web Robot.csv",header = TRUE)



data=mydata

WebRobot = ifelse(mydata$Robot==1,"T","F")

data=data.frame(data,WebRobot)

data=data[,c(1:12,14)]

robot_data=data[which(data$WebRobot=='T'),]



#problem b



# split the dataset into training and validation

set.seed(12345)
```

```
trainindex=sample(935,562,replace = FALSE)

training=data[trainindex,]

validation=data[-trainindex,]



# build a logistic regression model with backward selection method

fit_temp=glm(WebRobot~TotalPages+ImagePages+TotalTime+RepeatedAccess+ErrorRequest+
GET+POST+HEAD+Breadth+Depth+MultiIP+MultiAgent,family=binomial,data=training)

backward=step(fit_temp,direction = 'backward')



val<-predict(backward,validation, type='response')

df_ques2<-cbind(validation,val)

response_ques2 <- as.factor(ifelse(df_ques2$val>0.5, 'T', 'F'))

df_ques2$response <- response_ques2



# get the confusion matrix , accuracy rate and misclassification rate
```

```
confmat_ques2=confusionMatrix(data=factor(df_ques2$response),
reference=factor(df_ques2$WebRobot), positive='T')

# Confusion matrix

confmat_ques2$table

# Accuracy rate

round(confmat_ques2$overall[1],3)

# Misclassification error

round(1-confmat_ques2$overall[1],3)



#get the coefficients for getting significant predictors

summary(backward)



#problem C

tree_ques3 = tree(WebRobot~., training)
```

```
summary(tree_ques3)



plot(tree_ques3)

text(tree_ques3,pretty=0)

tree_ques3



# size , dev , k and method values

cvtree_ques3=cv.tree(tree_ques3,FUN = prune.misclass)

names(cvtree_ques3)

cvtree_ques3



prune_tree=prune.misclass(tree_ques3,best='5')

plot(prune_tree)

text(prune_tree,pretty = 0)

tree.pred = predict(prune_tree , validation, type="class")
```

```
table(tree.pred,validation$WebRobot)



acc=round((169+63)/374,5)

# accuracy

acc

#Misclassification

1-accuracy



# problem d is answered in the word file



# problem e

set.seed(12345)

rf_out <- randomForest(WebRobot ~ ., data=training)

var_importance <- data_frame(variable=setdiff(colnames(training), "WebRobot"),

                             importance=as.vector(importance(rf_out)))
```

```
var_importance <- arrange(var_importance, desc(importance))

var_importance$variable <- factor(var_importance$variable, levels=var_importance$variable)



p <- ggplot(var_importance, aes(x=variable, weight=importance, fill=variable))

p <- p + geom_bar() + ggtitle("Variable Importance from Random Forest Fit")

p <- p + xlab("Demographic Attribute") + ylab("Variable Importance (Mean Decrease in Gini Index)")

p <- p + scale_fill_discrete(name="Variable Name")

p + theme(axis.text.x=element_blank(),

        axis.text.y=element_text(size=12),

        axis.title=element_text(size=16),

        plot.title=element_text(size=18),

        legend.title=element_text(size=16),

        legend.text=element_text(size=12))

predict_rf=predict(rf_out, newdata=validation)
```

```
table(predict_rf,validation$WebRobot)

#accuracy

x=round((168+76)/374,3)

x

#misclassification

1-x

# problem f is answered in the word file
```