MIS 510 Portfolio Project Option 1

Joshua Humphreys

6/12/2022

## Loading data and libraries

setwd("C:/Users/jdhum/OneDrive/Documents/Grad School/MIS510/sandbox")  
germancredit.df<-read.csv("GermanCredit.csv")  
library(gains)  
library(MASS)  
library(neuralnet)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

## Preliminary Data Analysis

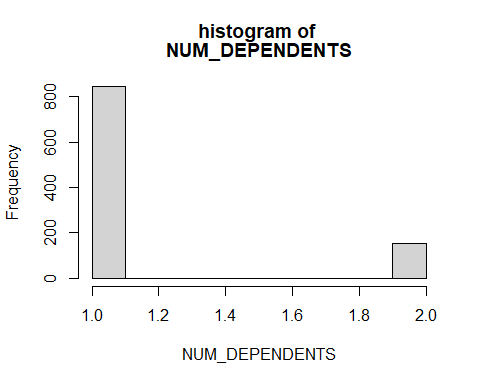
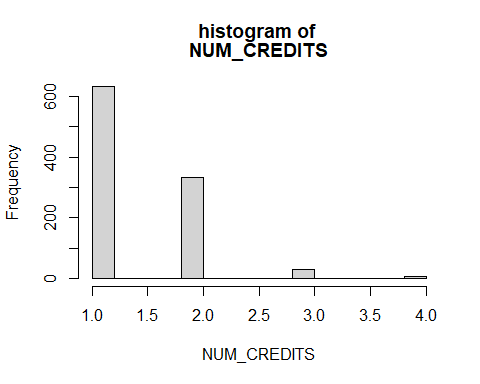
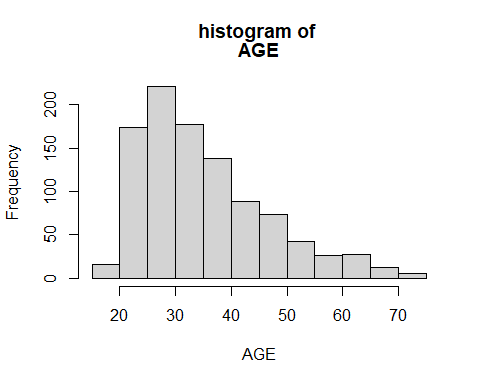
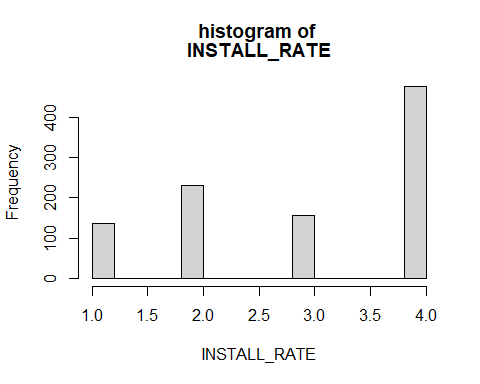
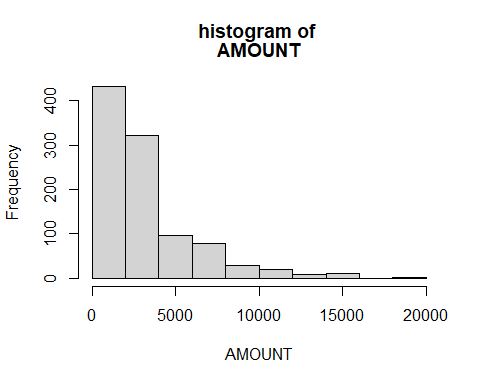
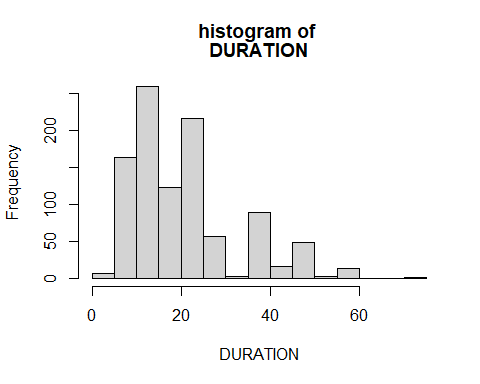
#display first 6 (default) rows of dataset  
head(germancredit.df)

## OBS. CHK\_ACCT DURATION HISTORY NEW\_CAR USED\_CAR FURNITURE RADIO.TV EDUCATION  
## 1 1 0 6 4 0 0 0 1 0  
## 2 2 1 48 2 0 0 0 1 0  
## 3 3 3 12 4 0 0 0 0 1  
## 4 4 0 42 2 0 0 1 0 0  
## 5 5 0 24 3 1 0 0 0 0  
## 6 6 3 36 2 0 0 0 0 1  
## RETRAINING AMOUNT SAV\_ACCT EMPLOYMENT INSTALL\_RATE MALE\_DIV MALE\_SINGLE  
## 1 0 1169 4 4 4 0 1  
## 2 0 5951 0 2 2 0 0  
## 3 0 2096 0 3 2 0 1  
## 4 0 7882 0 3 2 0 1  
## 5 0 4870 0 2 3 0 1  
## 6 0 9055 4 2 2 0 1  
## MALE\_MAR\_or\_WID CO.APPLICANT GUARANTOR PRESENT\_RESIDENT REAL\_ESTATE  
## 1 0 0 0 4 1  
## 2 0 0 0 2 1  
## 3 0 0 0 3 1  
## 4 0 0 1 4 0  
## 5 0 0 0 4 0  
## 6 0 0 0 4 0  
## PROP\_UNKN\_NONE AGE OTHER\_INSTALL RENT OWN\_RES NUM\_CREDITS JOB NUM\_DEPENDENTS  
## 1 0 67 0 0 1 2 2 1  
## 2 0 22 0 0 1 1 2 1  
## 3 0 49 0 0 1 1 1 2  
## 4 0 45 0 0 0 1 2 2  
## 5 1 53 0 0 0 2 2 2  
## 6 1 35 0 0 0 1 1 2  
## TELEPHONE FOREIGN RESPONSE  
## 1 1 0 1  
## 2 0 0 0  
## 3 0 0 1  
## 4 0 0 1  
## 5 0 0 0  
## 6 1 0 1

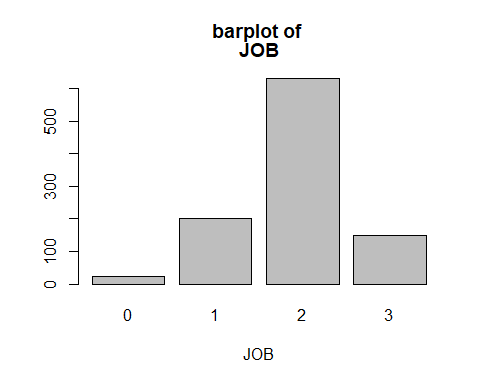
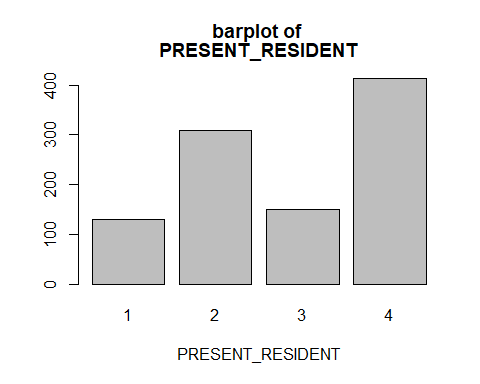
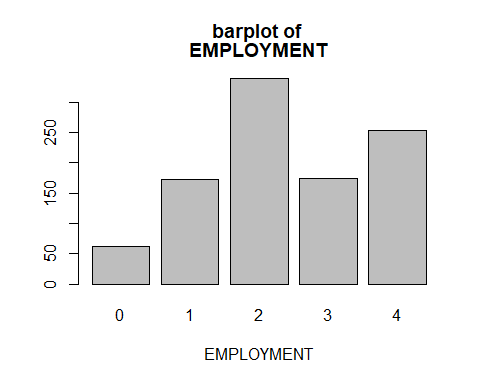
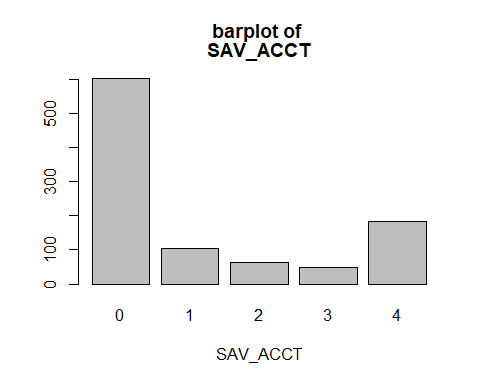
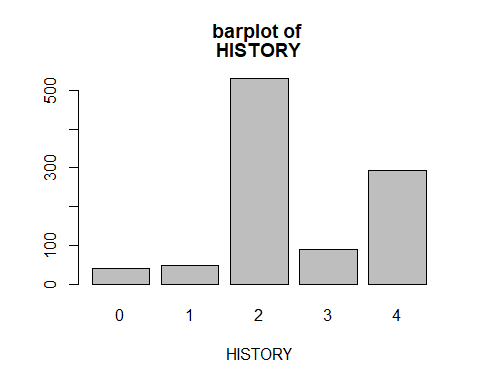
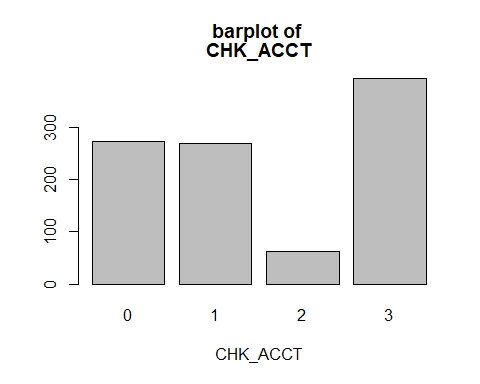
#isolate numeric variables for analysis  
germancredit.num<-germancredit.df[,c(3,11,14,23,27,29)]  
  
#summary statistics of numeric variables  
data.frame(mean=sapply(germancredit.num,mean,na.rm=TRUE),  
 sd=sapply(germancredit.num,sd,na.rm=TRUE),  
 min=sapply(germancredit.num,min,na.rm=TRUE),  
 max=sapply(germancredit.num,max,na.rm=TRUE),  
 median=sapply(germancredit.num,median,na.rm=TRUE),  
 length=sapply(germancredit.num,length),  
 miss.val=sapply(germancredit.num,  
 function(x)sum(length(which(is.na(x))))))

## mean sd min max median length miss.val  
## DURATION 20.903 12.0588145 4 72 18.0 1000 0  
## AMOUNT 3271.258 2822.7368760 250 18424 2319.5 1000 0  
## INSTALL\_RATE 2.973 1.1187147 1 4 3.0 1000 0  
## AGE 35.546 11.3754686 19 75 33.0 1000 0  
## NUM\_CREDITS 1.407 0.5776545 1 4 1.0 1000 0  
## NUM\_DEPENDENTS 1.155 0.3620858 1 2 1.0 1000 0

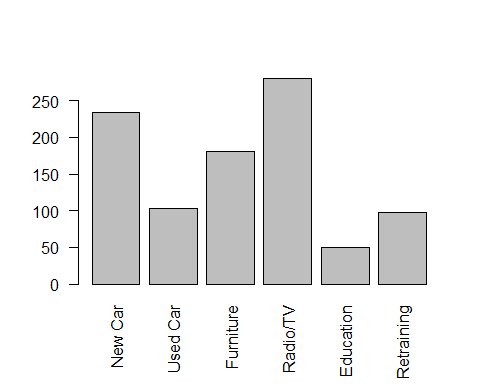
#produce histograms of numeric variables  
for(i in c(3,11,14,23,27,29)){  
 cname=colnames(germancredit.df)[i]  
 hist(germancredit.df[,i],xlab=cname,main=c("histogram of ",cname))  
}



#produce barplot of categorical variables  
for(i in c(2,4,12,13,20,28)){  
 cname=colnames(germancredit.df)[i]  
 barplot(table(germancredit.df[,i]),xlab=cname,main=c("barplot of ",cname))  
}



#Barplot of purpose variable  
#The variable "purpose" was originally a categorical variable, but was partitioned   
#in to dummy variables. The following code is kind of a brute force way to   
#visualize that data.  
  
Purpose<-c(sum(germancredit.df$NEW\_CAR),sum(germancredit.df$USED\_CAR),  
 sum(germancredit.df$FURNITURE),sum(germancredit.df$RADIO.TV),  
 sum(germancredit.df$EDUCATION),sum(germancredit.df$RETRAINING))  
Purpose.names<-c('New Car','Used Car', 'Furniture','Radio/TV','Education','Retraining')  
  
barplot(Purpose,names.arg = Purpose.names, las=2)

 ## Data Preparation for Logistic Regression

#partition data set for logistic regression  
set.seed(1)  
credittrain.index<-sample(c(1:dim(germancredit.df)[1]),dim(germancredit.df)[1]\*0.6)  
credittrain.df<-germancredit.df[credittrain.index,]  
creditvalid.df<-germancredit.df[-credittrain.index,]

## Logistic model creation

The following model uses all predictors in the data set.

#create logistic regression model  
logit.reg<-glm(RESPONSE~., data=credittrain.df[,-1], family="binomial")  
summary (logit.reg)

##   
## Call:  
## glm(formula = RESPONSE ~ ., family = "binomial", data = credittrain.df[,   
## -1])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4877 -0.7199 0.4212 0.7085 1.9293   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.170e-01 1.192e+00 0.769 0.44161   
## CHK\_ACCT 5.226e-01 9.122e-02 5.729 1.01e-08 \*\*\*  
## DURATION -2.930e-02 1.194e-02 -2.455 0.01409 \*   
## HISTORY 3.124e-01 1.139e-01 2.743 0.00610 \*\*   
## NEW\_CAR -9.934e-01 5.252e-01 -1.892 0.05854 .   
## USED\_CAR 1.226e-01 6.381e-01 0.192 0.84767   
## FURNITURE -2.691e-01 5.450e-01 -0.494 0.62141   
## RADIO.TV -1.552e-01 5.287e-01 -0.294 0.76909   
## EDUCATION -9.818e-01 6.735e-01 -1.458 0.14493   
## RETRAINING -4.702e-01 5.931e-01 -0.793 0.42791   
## AMOUNT -5.704e-05 5.798e-05 -0.984 0.32522   
## SAV\_ACCT 2.522e-01 7.928e-02 3.181 0.00147 \*\*   
## EMPLOYMENT 1.229e-01 9.576e-02 1.283 0.19935   
## INSTALL\_RATE -3.641e-01 1.162e-01 -3.133 0.00173 \*\*   
## MALE\_DIV -5.080e-02 4.934e-01 -0.103 0.91800   
## MALE\_SINGLE 4.915e-01 2.624e-01 1.873 0.06100 .   
## MALE\_MAR\_or\_WID 5.266e-01 4.176e-01 1.261 0.20732   
## CO.APPLICANT -2.516e-01 5.315e-01 -0.473 0.63601   
## GUARANTOR 1.815e-01 5.160e-01 0.352 0.72508   
## PRESENT\_RESIDENT 1.232e-01 1.101e-01 1.119 0.26329   
## REAL\_ESTATE 2.646e-01 2.716e-01 0.974 0.32993   
## PROP\_UNKN\_NONE -8.773e-01 5.298e-01 -1.656 0.09776 .   
## AGE 1.461e-02 1.087e-02 1.344 0.17884   
## OTHER\_INSTALL -6.424e-01 2.590e-01 -2.480 0.01314 \*   
## RENT -6.782e-01 6.388e-01 -1.062 0.28837   
## OWN\_RES -1.249e-01 6.093e-01 -0.205 0.83763   
## NUM\_CREDITS -2.754e-01 2.030e-01 -1.357 0.17485   
## JOB -1.776e-01 1.871e-01 -0.949 0.34253   
## NUM\_DEPENDENTS 1.944e-02 3.123e-01 0.062 0.95037   
## TELEPHONE 5.049e-01 2.484e-01 2.033 0.04209 \*   
## FOREIGN 2.713e+00 1.129e+00 2.404 0.01624 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 731.33 on 599 degrees of freedom  
## Residual deviance: 553.79 on 569 degrees of freedom  
## AIC: 615.79  
##   
## Number of Fisher Scoring iterations: 6

#validate model  
logit.reg.pred<-predict(logit.reg, creditvalid.df[,-1], type="response")

## Logistic Regression Model Analysis

#confusion matrix for logistic regression model  
logit.reg.pred.bin<-ifelse(logit.reg.pred>0.5,1,0)  
confusionMatrix(factor(logit.reg.pred.bin),factor(creditvalid.df$RESPONSE))

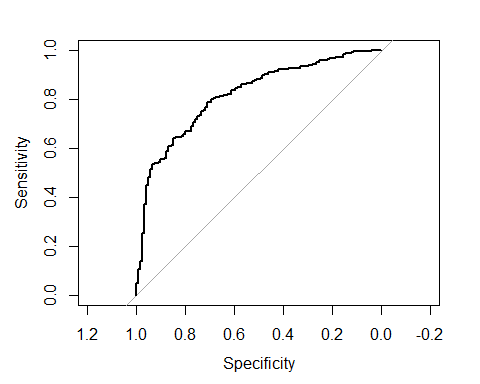
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 63 34  
## 1 58 245  
##   
## Accuracy : 0.77   
## 95% CI : (0.7256, 0.8104)  
## No Information Rate : 0.6975   
## P-Value [Acc > NIR] : 0.0007502   
##   
## Kappa : 0.4225   
##   
## Mcnemar's Test P-Value : 0.0164887   
##   
## Sensitivity : 0.5207   
## Specificity : 0.8781   
## Pos Pred Value : 0.6495   
## Neg Pred Value : 0.8086   
## Prevalence : 0.3025   
## Detection Rate : 0.1575   
## Detection Prevalence : 0.2425   
## Balanced Accuracy : 0.6994   
##   
## 'Positive' Class : 0   
##

#ROC curve and area under curve for logistic regression model  
r<-roc(creditvalid.df$RESPONSE,logit.reg.pred)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

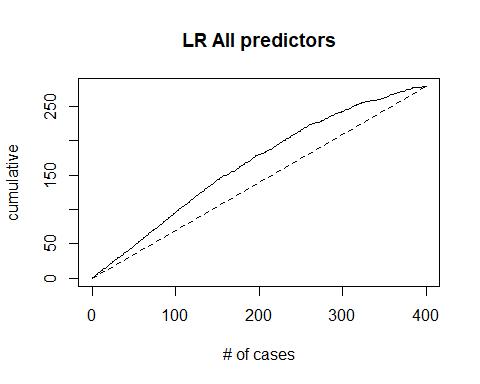
plot.roc(r)



auc(r)

## Area under the curve: 0.8148

#lift chart for logistic regression model with all predictors  
gain<-gains(creditvalid.df$RESPONSE, logit.reg.pred, groups=length(logit.reg.pred))  
plot(c(0,gain$cume.pct.of.total\*sum(creditvalid.df$RESPONSE))~c(0,gain$cume.obs),  
 xlab="# of cases", ylab="cumulative", main="LR All predictors", type ="l")  
lines(c(0,sum(creditvalid.df$RESPONSE))~c(0,dim(creditvalid.df)[1]), lty=2)

 ## Logistic Regression with Stepwise AIC model selection

The following code creates a logistic regression model using stepwise predictor selection based on the Akaike Information Criteria (AIC)

#eliminate unneeded variables using Stepwise AIC model selection  
step.model<-stepAIC(logit.reg, trace=FALSE)  
coef(step.model)

## (Intercept) CHK\_ACCT DURATION HISTORY NEW\_CAR   
## 0.83251374 0.51289697 -0.03747087 0.33936976 -0.80615201   
## EDUCATION SAV\_ACCT EMPLOYMENT INSTALL\_RATE MALE\_SINGLE   
## -0.83119507 0.24955708 0.15173405 -0.30776043 0.48596760   
## MALE\_MAR\_or\_WID PROP\_UNKN\_NONE AGE OTHER\_INSTALL RENT   
## 0.60953202 -0.75713394 0.01723248 -0.65378239 -0.47123846   
## NUM\_CREDITS JOB TELEPHONE FOREIGN   
## -0.28586388 -0.27089123 0.46978483 2.63649423

#make prediction set with improved model  
step.model.pred<-predict(step.model, creditvalid.df[,-1], type = "response")

## Analysis of Logistic Regression with AIC Stepwise Selection

#confusion matrix for stepwise selection model  
step.model.pred.bin<-ifelse(step.model.pred>0.5,1,0)  
confusionMatrix(factor(step.model.pred.bin),factor(creditvalid.df$RESPONSE))

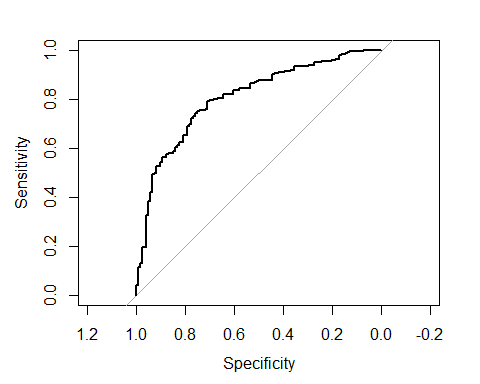
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 57 34  
## 1 64 245  
##   
## Accuracy : 0.755   
## 95% CI : (0.7098, 0.7964)  
## No Information Rate : 0.6975   
## P-Value [Acc > NIR] : 0.006392   
##   
## Kappa : 0.3756   
##   
## Mcnemar's Test P-Value : 0.003396   
##   
## Sensitivity : 0.4711   
## Specificity : 0.8781   
## Pos Pred Value : 0.6264   
## Neg Pred Value : 0.7929   
## Prevalence : 0.3025   
## Detection Rate : 0.1425   
## Detection Prevalence : 0.2275   
## Balanced Accuracy : 0.6746   
##   
## 'Positive' Class : 0   
##

#ROC curve for stepwise selection model  
r<-roc(creditvalid.df$RESPONSE,step.model.pred)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

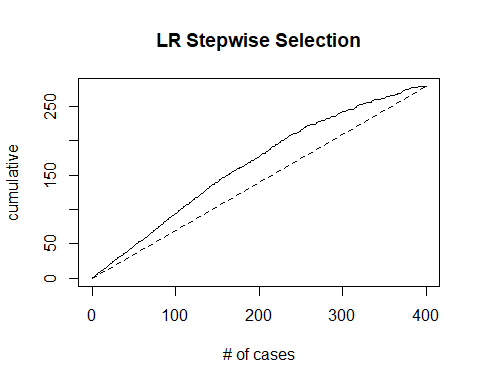
plot.roc(r)



auc(r)

## Area under the curve: 0.8051

#lift chart using stepwise selection method  
gain2<-gains(creditvalid.df$RESPONSE, step.model.pred, groups=length(step.model.pred))  
plot(c(0,gain2$cume.pct.of.total\*sum(creditvalid.df$RESPONSE))~c(0,gain2$cume.obs),  
 xlab="# of cases", ylab="cumulative", main="LR Stepwise Selection", type ="l")  
lines(c(0,sum(creditvalid.df$RESPONSE))~c(0,dim(creditvalid.df)[1]), lty=2)

 ## Data preparation for Neural Net Model

#scaling data and removing obs variable  
maxs <- apply(germancredit.df[,-1], 2, max)   
mins <- apply(germancredit.df[,-1], 2, min)  
scaledcredit.df <- as.data.frame(scale(germancredit.df[,-1], center = mins, scale = maxs - mins))  
  
#partition data set  
set.seed(1)  
scaledtrain.index<-sample(c(1:dim(scaledcredit.df)[1]),dim(scaledcredit.df)[1]\*0.6)  
scaledtrain.df<-scaledcredit.df[scaledtrain.index,]  
scaledvalid.df<-scaledcredit.df[-scaledtrain.index,]

## Neural Net Model Production

The following code produces a neural net with a single hidden layer with 20 nodes.

#neural net  
neural.net<-neuralnet(RESPONSE~.,data = scaledtrain.df, hidden=25)

## Analysis of Neural Net model

The following is my analysis of my neural net.

#  
#confusion matrix of neural net performance on training data  
nn.train.pred=compute(neural.net, scaledtrain.df)  
nn.train.pred.bin<-ifelse(nn.train.pred$net.result>0.5,1,0)  
confusionMatrix(factor(nn.train.pred.bin),factor(credittrain.df$RESPONSE))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 179 0  
## 1 0 421  
##   
## Accuracy : 1   
## 95% CI : (0.9939, 1)  
## No Information Rate : 0.7017   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 1   
##   
## Mcnemar's Test P-Value : NA   
##   
## Sensitivity : 1.0000   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 1.0000   
## Prevalence : 0.2983   
## Detection Rate : 0.2983   
## Detection Prevalence : 0.2983   
## Balanced Accuracy : 1.0000   
##   
## 'Positive' Class : 0   
##

#validate neural net  
nn.valid.pred=compute(neural.net, scaledvalid.df)  
nn.valid.pred.bin<-ifelse(nn.valid.pred$net.result>0.5,1,0)  
  
#confusion matrix for neural net performance on validation dataset  
confusionMatrix(factor(nn.valid.pred.bin),factor(creditvalid.df$RESPONSE))

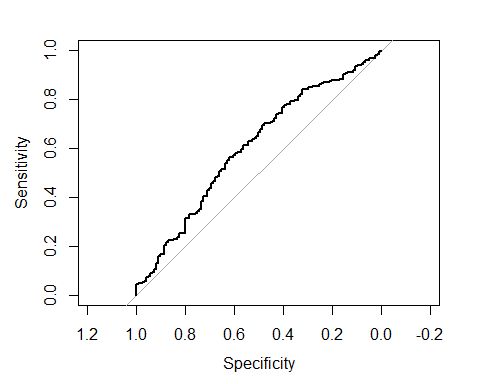
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 59 88  
## 1 62 191  
##   
## Accuracy : 0.625   
## 95% CI : (0.5755, 0.6726)  
## No Information Rate : 0.6975   
## P-Value [Acc > NIR] : 0.99919   
##   
## Kappa : 0.1623   
##   
## Mcnemar's Test P-Value : 0.04123   
##   
## Sensitivity : 0.4876   
## Specificity : 0.6846   
## Pos Pred Value : 0.4014   
## Neg Pred Value : 0.7549   
## Prevalence : 0.3025   
## Detection Rate : 0.1475   
## Detection Prevalence : 0.3675   
## Balanced Accuracy : 0.5861   
##   
## 'Positive' Class : 0   
##

#ROC Curve for Neural Net  
r<-roc(creditvalid.df$RESPONSE,nn.valid.pred$net.result)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

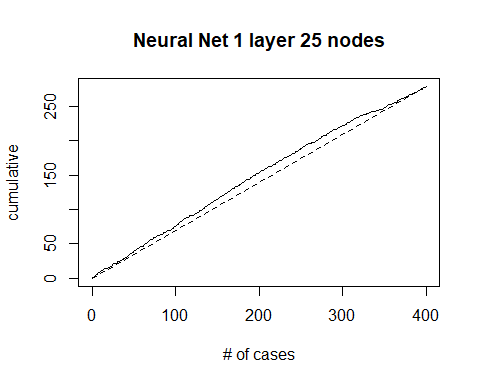
plot.roc(r)



auc(r)

## Area under the curve: 0.602

#Lift Chart for Neural Net  
gain3<-gains(scaledvalid.df$RESPONSE, nn.valid.pred$net.result, groups=length(nn.valid.pred$net.result))  
plot(c(0,gain3$cume.pct.of.total\*sum(creditvalid.df$RESPONSE))~c(0,gain3$cume.obs),  
 xlab="# of cases", ylab="cumulative", main="Neural Net 1 layer 25 nodes", type ="l")  
lines(c(0,sum(creditvalid.df$RESPONSE))~c(0,dim(creditvalid.df)[1]), lty=2)



**Explanation**

To start this project, I downloaded the datafile, converted it to a .csv and uploaded it to Rstudio using the *read.csv()* function. I then used the *head()* function to inspect the first 6 rows of each column in the data set. The *head()* function worked well along with the data description given by Shmueli et al. (2018) in section 21.2 of the text to provide a broad overview of the structure of the dataset.

Next, I selected the columns of numeric data type and produced a dataframe of summary statistics using the *data.frame()* and *sapply()* functions. I then used a for loop to produce histograms of each of the numeric predictors. Analysis of the numeric predictors revealed that none of them followed a normal distribution. *Duration, Amount*, and *Age* each demonstrated a right skew. *Install\_Rate, Num\_Credits,* and *Num\_Dependents*, while technically being numerical, did not appear to be continuous. Rather, they appeared ordinal.

Next, I produced barplots of each of the categorical variables. Analysis of the categorical variables revealed that none of the categories within each of the categorical variables was evenly represented in the dataset. Additionally, the dataset included a categorical variable that had been converted to dummy variables. This variable referred to the purpose of credit and originally contained categories: *New\_Car, Used\_Car, Furniture, Radio/TV, Education,* and *Retraining.* Using the *sum()* function, I produced a vector of the counts within each dummy variable. I then produced a vector of names. I then used the *barplot()* function to display a barplot of the distribution of this variable. This was somewhat of a brute force method but allowed me to visualize that variable without having to coerce the data in any way.

To prepare the data for logistic regression I partitioned the data into a training and validation set. I used a 60-40 split between the two sets using the *sample()* function. I then trained a logistic regression model using all predictors. I used the *glm()* function to do so. I then validated the model using the validation data set. I produced a confusion matrix, ROC curve, and Lift chart for the logistic regression model. Next, I decided to refine my logistic regression model using stepwise selection based upon the Akaike Information Criteria. This is the same method I have used in previous classes to avoid overfitting logistic regression. I loaded the *MASS* library and used the *StepAIC()* function. Once again, I produced a confusion matrix, ROC curve, and Lift Chart for the logistic regression model.

To prepare the data for a neural network, I first scaled the data by using the *scale()* function. According to Alice (2018), Neural nets perform better when their variables are scaled to the interval [0,1] or [-1,1]. To scale the variables in the data set, I followed the procedure given by Alice (2018). Next, I partitioned the scaled data set in to training and validation sets using the same procedure as before. Again, following the procedure prescribed by Alice (2018), I trained a neural net using the training data set. For my hidden layers and nodes, I used a single hidden layer of 25 nodes. This was the result of trial an error with several different configurations. According to the rule of thumb presented by Alice(2018) the number of nodes in the first hidden layer should be somewhere around 2/3 the size of the input layer (in this case, 30). According to Shmueli et al.(2018), one hidden layer is generally sufficient. Furthermore, Shmueli et al.(2018) recommend starting with a number of nodes and check for overfitting. Therefore, I started with 20 and tried several different configurations. Of the layer/node configurations I tried, 25 appeared to produce the most stability. Finally, I produced a confusion matrix, ROC chart, and Lift chart for the neural network model using the same procedures as before.

**Analysis**

The bottom line of my analysis is that the logistic regression model using all predictors appears to be the best model based on an Accuracy of .77, Sensitivity of .5207, Specificity of .8781, and AUC of .8148. This, of course is compared to the stepwise selection model’s accuracy of .755, sensitivity of .4711, specificity of .8781, and AUC of .8051 and the neural net’s accuracy of .625, sensitivity of .4876, specificity of .6846, and AUC of .602. These differences can be well visualized via examination of the ROC and lift Charts as well.

**Final Thoughts**

I believe some further tuning could be performed on the neural net. Both in terms of the number of nodes used and the number of hidden layers. I used the default value for *stepmax* in the *neuralnet()* function (meaning I didn’t change the number of training iterations used in any of my models). I think, given more time, there are several variables to examine that could potentially improve the overall performance of the neural net. I also didn’t explore all available options in terms of predictor selection for the logistic regression model. There’s certainly potential to make the model more accurate. Furthermore, I did not account for asymmetric misclassification costs in any of these models. With those limitations aside, this was an exceptionally revealing assignment. Before studying data analytics, I assumed the most challenging piece of completing a data mining or data analytics project would be the production of models. I couldn’t have been more wrong. What has been consistently reaffirmed to me throughout this class is that data collection and preparation are far more challenging that model production. This, of course, is true even with the data sets we have used in this class where most of the collection work has already been done. Furthermore, understanding and interpreting model results is often quite challenging. A final challenge was revealed in this assignment: model tuning. As I mentioned, there are several ways that each of these models could be tuned beyond what I have accomplished here.

**References**

Alice, M.(2018, February 3). *Fitting a Neural Network in R; neuralnet package.* Datascience+. <https://datascienceplus.com/fitting-neural-network-in-r/>

R Core Team and Contributors Worldwide. (2021). *The R Base Package.* R Core Team.

Shmueli, G., Bruce, P.C., Yahav, I., Patel, N. R. & Lichtendahl, K.C. (2018). *Data Mining For Business Analytics: Concepts, Techniques, and Applications in R.* John Wiley & Sons, Inc.

Kassambara, A.(2018, November 3). *Stepwise Logistic Regression in R.* Statistical tools for high-throughput data analysis. <http://www.sthda.com/english/articles/36-classification-methods-essentials/150-stepwise-logistic-regression-essentials-in-r/>