1. Set the data directory and Read in the data file.

setwd("/Users/xiqia/Documents/Informs/Data")
data<-read.csv('CustR.csv')

2.Setup in the data file for NNET processing

install.packages('neuralnet')
library(neuralnet)

#Check for N.A
> sum(is.na(data))
[1] 4

#gender has one missing value, marital has one missing value and income1K has two missing values.
> colSums(is.na(data))

id	age	income	gender	marital	numkids	numcards	howpaid	mortgage	storecar	loans	risk	income1k
0	0	0	1	1	0	0	0	0	0	0	0	2

#Remove rows with N.A. The number of observations changed from 2457 to 2453. '13'menas there are 13 variables.
> data<-na.omit(Data)
> dim(data)
[1] 2453   13

#Understanding data
> summary(data)

```
       id              age            income          gender         marital         numkids         numcards        howpaid     
 Min.   :100001   Min.   :18.00   Min.   :15018   Min.   :1.000   Min.   :1.000   Min.   :0.000   Min.   :0.000   Min.   :1.000  
 1st Qu.:101026   1st Qu.:23.00   1st Qu.:20558   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000  
 Median :102053   Median :31.00   Median :23611   Median :1.000   Median :2.000   Median :1.000   Median :2.000   Median :2.000  
 Mean   :102051   Mean   :31.88   Mean   :25664   Mean   :1.499   Mean   :2.075   Mean   :1.446   Mean   :2.425   Mean   :1.512  
 3rd Qu.:103086   3rd Qu.:41.00   3rd Qu.:27624   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:2.000  
 Max.   :104117   Max.   :50.00   Max.   :59944   Max.   :2.000   Max.   :3.000   Max.   :4.000   Max.   :6.000   Max.   :2.000  
    mortgage        storecar         loans            risk          income1k    
 Min.   :1.000   Min.   :0.000   Min.   :0.000   Min.   :1.000   Min.   : 4.33  
 1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:20.55  
 Median :2.000   Median :2.000   Median :1.000   Median :2.000   Median :23.61  
 Mean   :1.778   Mean   :2.496   Mean   :1.372   Mean   :1.944   Mean   :25.69  
 3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:27.64  
 Max.   :2.000   Max.   :5.000   Max.   :3.000   Max.   :3.000   Max.   :84.56  
```

#We changed the type of variable 'risk' to character since values '1', '2', '3' are categorical means "Low", "Medium", "High"
data$risk<-as.character(data$risk)
data$risk <- factor(data$risk, levels = c("1","2","3"),labels = c("Low","Medium","High"))

#We made churn matrix with appropriate column names.

churn.matrix<-
model.matrix(~risk+age+income+gender+marital+numkids+numcards+howpaid+mortgage+storecar+loans-1, data=data)
colnames(churn.matrix)<-
c('Low','Medium','High','age','income','gender','marital','numkids','numcards','howpaid','mortgage','storecar','loans')

3. Create train and test data subset

#For simulation, we make 100 random numbers
set.seed(100)
#We partition the source data for 60% training set and 40% testing set
index <- sample(1:nrow(data),floor(nrow(data)*0.6))
train_set <- churn.matrix[index,]
test_set <- churn.matrix[-index,]

4. Set NNET execution parameters

#We created predictor data set and all rows and columns after the 3 target outcome variables
train_predictors <- train_set[,4:13]
test_predictors <- test_set[,4:13]

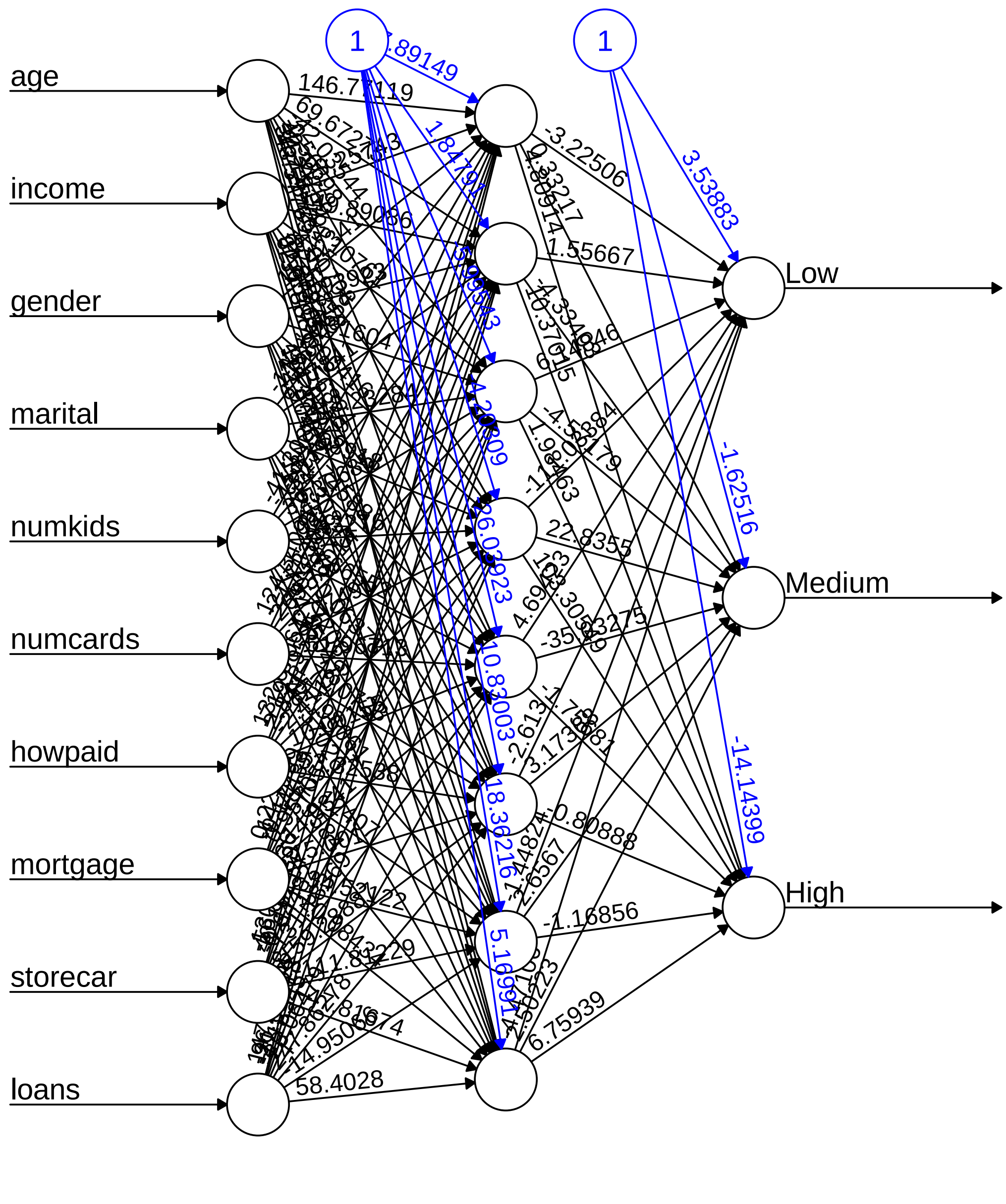
#We converted actual predictor values into scaled values and apply finds minimum value for each predictor variable. Next line calculates the range (max-min) for each predictor. Next 2 lines scales and appends prior created predictor values to target variables for training and test data set
min_vector <- apply(train_predictors,2,min)
range_vector <- apply(train_predictors,2,max)-apply(train_predictors,2,min)
train_scaled <- cbind(scale(train_predictors,min_vector,range_vector),train_set[,1:3])
test_scaled <- cbind(scale(test_predictors,min_vector,range_vector),test_set[,1:3])
summary(train_scaled)

```
      age               income               gender              marital             numkids             numcards      
 Min.   :0.0000000   Min.   :0.0000000   Min.   :0.0000000   Min.   :0.0000000   Min.   :0.0000000   Min.   :0.0000000  
 1st Qu.:0.1562500   1st Qu.:0.1277545   1st Qu.:0.0000000   1st Qu.:0.5000000   1st Qu.:0.2500000   1st Qu.:0.1666667  
 Median :0.4062500   Median :0.1920491   Median :0.0000000   Median :0.5000000   Median :0.2500000   Median :0.3333333  
 Mean   :0.4391783   Mean   :0.2393850   Mean   :0.4989803   Mean   :0.5305914   Mean   :0.3708362   Mean   :0.4083390  
 3rd Qu.:0.7187500   3rd Qu.:0.2815519   3rd Qu.:1.0000000   3rd Qu.:1.0000000   3rd Qu.:0.5000000   3rd Qu.:0.6666667  
 Max.   :1.0000000   Max.   :1.0000000   Max.   :1.0000000   Max.   :1.0000000   Max.   :1.0000000   Max.   :1.0000000  
    howpaid             mortgage            storecar              loans                Low                Medium       
 Min.   :0.0000000   Min.   :0.0000000   Min.   :0.0000000   Min.   :0.0000000   Min.   :0.0000000   Min.   :0.0000000  
 1st Qu.:0.0000000   1st Qu.:1.0000000   1st Qu.:0.2000000   1st Qu.:0.3333333   1st Qu.:0.0000000   1st Qu.:0.0000000  
 Median :1.0000000   Median :1.0000000   Median :0.4000000   Median :0.3333333   Median :0.0000000   Median :1.0000000  
 Mean   :0.5139361   Mean   :0.7763426   Mean   :0.5046907   Mean   :0.4640834   Mean   :0.2358939   Mean   :0.5968729  
 3rd Qu.:1.0000000   3rd Qu.:1.0000000   3rd Qu.:0.6000000   3rd Qu.:0.6666667   3rd Qu.:0.0000000   3rd Qu.:1.0000000  
 Max.   :1.0000000   Max.   :1.0000000   Max.   :1.0000000   Max.   :1.0000000   Max.   :1.0000000   Max.   :1.0000000  
      High         
 Min.   :0.0000000  
 1st Qu.:0.0000000  
 Median :0.0000000  
 Mean   :0.1672332  
 3rd Qu.:0.0000000  
 Max.   :1.0000000  
```

5. Run NNET function and Plot the NNET

set.seed(100)

```
fit_1 <- neuralnet(Low + Medium + High ~
age+income+gender+marital+numkids+numcards+howpaid+mortgage+storecar+loans,train_scal
ed,hidden = c(8),linear.output = F)
plot(fit_1)
```



6. We created and display ‘3x3 table accuracy’

 # Max is the location of the maximum value in the data vector
predictions <- compute(fit_1,test_scaled[,1:10])

output_sequence <- c('Low','Medium','High')
y_predicted <- apply(predictions$net.result,1,which.max)
y_true <- apply(test_scaled[,11:13],1,which.max)

#We created predictive accuracy table
confusion_matrix <- table(y_true,y_predicted)
rownames(confusion_matrix) <- colnames(confusion_matrix) <- output_sequence
accuracy <- sum(diag(confusion_matrix))/sum(confusion_matrix)
print(confusion_matrix)
print(paste('Accuracy',accuracy))

```
> print(confusion_matrix)
        y_predicted
y_true   Low Medium High
  Low     64    128   19
  Medium  17    548   31
  High    20     76   79
> print(paste('Accuracy',accuracy))
[1] "Accuracy 0.703665987780041"
```

	Y_Predicted		
Y_True	Low	Medium	High
Low	64	128	19
Medium	17	548	31
High	20	76	79

Accuracy Low
64/(64+128+19)=0.30331754

Accuracy Medium
548/(17+548+31)=0.91946309

Accuracy High
79/(20+76+79)=0.45142857

Overall Accuracy
(64+548+79)/(64+128+19)+ (17+548+31) + (20+76+79)=0.7036659877