Report for Neural Network on Prudential Life Insurance Dataset

Dataset : Prudential Life Insurance Dataset

Motive: To predict the decision of sanction of the loan using Neural Networks for loan dataset

R-studio code:

```
getwd()
setwd("E:/ADS/Assignment/Assignment 6")
```

Load file onto R

```
# Read CSV into R

require(XLConnect)
library(xlsx)
loan= read.xlsx("loan.xlsx",sheetIndex = 1)
```

Data Preprocessing

```
# Read CSV into R

mydata <- read.csv(file="train.csv", header=TRUE, sep=",")
options(max.print=1000000)

#Pre-processing starts here

#Deleting columns with empty cells more than 60%

mydata$Family_Hist_3 <- NULL
mydata$Family_Hist_5 <- NULL
mydata$Medical_History_10 <- NULL
mydata$Medical_History_15 <- NULL
mydata$Medical_History_24 <- NULL
mydata$Medical_History_32 <- NULL
mydata$Id <- NULL

#Filling missing values with mean value for Employment_Info_1

mydata$Employment_Info_1[is.na(mydata$Employment_Info_1)] <-
mean(mydata$Employment_Info_1, na.rm = T)

#Filling missing values with mean value for Employment_Info_4

mydata$Employment_Info_4[is.na(mydata$Employment_Info_4)] <-
mean(mydata$Employment_Info_4, na.rm = T)

#Filling missing values with mean value for Employment_Info_6
```

```
mydata$Employment_Info_6[is.na(mydata$Employment_Info_6)] <-
mean(mydata$Employment_Info_6, na.rm = T)

#Filling missing values with mean value for Insurance_History_5

mydata$Insurance_History_5[is.na(mydata$Insurance_History_5)] <-
mean(mydata$Insurance_History_5, na.rm = T)

#Filling missing values with mean value for Family_Hist_2

mydata$Family_Hist_2[is.na(mydata$Family_Hist_2)] <- mean(mydata$Family_Hist_2, na.rm = T)

#Filling missing values with mean value for Family_Hist_4

mydata$Family_Hist_4[is.na(mydata$Family_Hist_4)] <- mean(mydata$Family_Hist_4, na.rm = T)

#Filling missing values with mean value for Medical_History_1

mydata$Medical_History_1[is.na(mydata$Medical_History_1)] <- mean(mydata$Medical_History_1,
na.rm = T)

#Convert Categorical using 1 to C Coding

data_ctgr <- mydata[c("Medical_History_1","Product_Info_1", "Product_Info_2", "Product_Info_3",
"Product_Info_5", "Product_Info_6", "Product_Info_7", "Employment_Info_2",
"Employment_Info_3", "Employment_Info_5", "InsuredInfo_1", "InsuredInfo_2", "InsuredInfo_3",
"InsuredInfo_4", "InsuredInfo_5", "InsuredInfo_6", "InsuredInfo_7", "Insurance_History_1",
"Insurance_History_2", "Insurance_History_3", "Insurance_History_4", "Insurance_History_7",
"Insurance_History_8", "Insurance_History_9", "Family_Hist_1", "Medical_History_2",
"Medical_History_3", "Medical_History_4", "Medical_History_5", "Medical_History_6",
"Medical_History_7", "Medical_History_8", "Medical_History_9", "Medical_History_11",
"Medical_History_12", "Medical_History_13", "Medical_History_14", "Medical_History_16",
"Medical_History_17", "Medical_History_18", "Medical_History_19", "Medical_History_20",
"Medical_History_21", "Medical_History_22", "Medical_History_23", "Medical_History_25",
"Medical_History_26", "Medical_History_27", "Medical_History_28", "Medical_History_29",
"Medical_History_30", "Medical_History_31", "Medical_History_33", "Medical_History_34",
"Medical_History_35", "Medical_History_36", "Medical_History_37", "Medical_History_38",
"Medical_History_39", "Medical_History_40", "Medical_History_41")]

OneToCconv <- acm.disjonctif(data_ctgr)

#Prepare the data

data_cntg <- mydata[c("Product_Info_4", "Ins_Age", "Ht", "Wt", "BMI", "Employment_Info_1",
"Employment_Info_4", "Employment_Info_6", "Insurance_History_5", "Family_Hist_2",
"Family_Hist_4")]

data_dummy<-
mydata[c("Medical_Keyword_1","Medical_Keyword_2","Medical_Keyword_3","Medical_Keyword_4
","Medical_Keyword_5","Medical_Keyword_6","Medical_Keyword_7","Medical_Keyword_8","Medi
```

```
cal_Keyword_9","Medical_Keyword_10","Medical_Keyword_11","Medical_Keyword_12","Medical_
Keyword_13","Medical_Keyword_14","Medical_Keyword_15","Medical_Keyword_16","Medical_Key
word_17","Medical_Keyword_18","Medical_Keyword_19", "Medical_Keyword_20",
"Medical_Keyword_21", "Medical_Keyword_22", "Medical_Keyword_23","Medical_Keyword_24",
"Medical_Keyword_25", "Medical_Keyword_26", "Medical_Keyword_27", "Medical_Keyword_28",
"Medical_Keyword_29","Medical_Keyword_30", "Medical_Keyword_31", "Medical_Keyword_32",
"Medical_Keyword_33","Medical_Keyword_34", "Medical_Keyword_35","Medical_Keyword_36",
"Medical_Keyword_37", "Medical_Keyword_38", "Medical_Keyword_39", "Medical_Keyword_40",
"Medical_Keyword_41", "Medical_Keyword_42", "Medical_Keyword_43", "Medical_Keyword_44",
"Medical_Keyword_45","Medical_Keyword_46", "Medical_Keyword_47","Medical_Keyword_48")]

final_data <- data.frame(c(OneToCconv, data_cntg,data_dummy))
Insurance <- data.frame(c(final_data, mydata[c("Response")]))
```

Converting data to Numeric

```
#Convert to Numeric

Insurance$Response<-as.numeric(Insurance$Response)
print(head(Insurance))
```

Moving only required columns to another dataframe

```
#Move required columns to one dataframe

myInsurance<-Insurance[c("BMI",
"Medical_History_4.1","Medical_History_4.2","Medical_Keyword_3", "Response")]

print(head(myInsurance))
library(ISLR)
library(neuralnet)
```

Separate train and test data

```
#Separate Train and Test data
train<-myInsurance[1:49382,]
print(head(train))
test<-myInsurance[49383:59382,]
```

Apply Neural Networks to train dataset

```
#apply neural network
nn <- neuralnet(train$Response ~
train$BMI+train$Medical_History_4.1+train$Medical_History_4.2+train$Medical_Keyword_3,data=t
rain, hidden=c(2,2,2),linear.output=FALSE)
```

Plot Neural network

```
#Plot Neural Network
plot(nn)
```

Predict on Test data

```
predicted.nn.values <- compute(nn,test[1:4])
predicted.nn.values$net.result
predicted.nn.values <- compute(nn,test[1:5])
idx <- apply(predicted.nn.values$net.result, 1, which.max)
head(idx)
pred <- c( '1', '2','3','4','5','6','7','8')[idx]
head(pred)
predicted.nn.values$net.result
```

#Draw Tabular form for noting all response

```
table(test$Response,predicted.nn.values$net.result)
length(nn$model.list$variables)
length(test[1,1:5])
length(test[1:5])
```

Plot Neural Network

#Plot Neural Network

```
plot(nn)
```