

Activity Sheet

Learning outcomes:

After solving these exercises, you should be able to understand the following:

1. Applying Neural Networks on multiple data sets
2. Importance of Standardizing data
3. Visualizing and interpreting the results

Today's activity

1. Importing "UniversalBank.csv" file's data into R dataframe using read.csv function.
`data = read.csv(file="UniversalBank.csv", header=TRUE, sep=",")`
2. Understand the structure and summary of the data using str and summary R commands
3. Convert all the variables to appropriate type
 - To numeric using as.numeric()
 - To categorical using as.factor()
 - To ordinal using as.factor() with ordered argument set to TRUE or using as.ordered()
4. Using subset remove 'ID' and 'ZIP.Code' columns from the data
`data = subset(data , select = -c(ID, ZIP.Code))`
5. R neuralnet library takes only numeric attributes as input
 - Convert all categorical and ordinal attributes to numeric using appropriate technique.
E.g. Convert "Education" categorical attribute to numeric using dummy function in dummies R library
 - Drop actual Education attribute from original data set
 - Add created dummy Education variables to original data set
6. Separate Target Variable and Independent Variables.
 - In this case "Personal.Loan" is a target variable and all others are independent variable.
7. Standardization the independent variables using decostand function in vegan R library
Note: To standardize the data using 'Range' method
8. Split the data into train and test data. *Note: 60:40 ratio*
9. In both Train and Test, see the data distribution w.r.t response variable
10. Load neuralnet R library and build a Neural Network having 1 hidden layer with 2 nodes

```
formula = as.formula(paste("Personal.Loan ~",  
                           paste(setdiff(names(train_Data), "Personal.Loan"),  
                                 collapse = " + ")))  
nn = neuralnet(formula, data=train_Data, hidden=2) |
```

11. See covariate and result variables of neuralnet model

```
out = cbind(nn$covariate, nn$net.result[[1]])
```

12. Remove row names and set column names

```
dimnames(out) = list(NULL, c("Age", "Experience",  
                             "Income", "Family",  
                             "CCAvg", "Mortgage",  
                             "Securities.Account",  
                             "CD.Account", "online",  
                             "CreditCard", "Education1",  
                             "Education2", "Education3",  
                             "nn_Output"))
```

13. To view top records in the data set

14. Plot the neural network

```
plot(nn)
```

15. Compute confusion matrix and calculate recall on Train Data

```
predicted = factor(ifelse(nn$net.result[[1]] > 0.5, 1, 0))  
conf_Matrix = table(train_Data$Personal.Loan, predicted)
```

16. Remove target attribute from Test Data

17. Predict

```
nn_predict = compute(nn, covariate= test_Data_No_Target)
```

18. View the predicted values

19. Compute confusion matrix and calculate recall for Test Data

```
predicted = factor(ifelse(nn_predict$net.result > 0.5, 1, 0))  
conf_Matrix = table(test_Data$Personal.Loan, predicted)
```