DATA MINING – 5334

Last Name: Alagiya
First Name: Ravi
UTA ID: 1001452485 | rxa2485
Assignment: Assignment 7
Due Date: 11/30/2016 11:28 PM

Questions:
Choose 3 classifiers from Bayesian, **KNN**, **Neural Network**, SVM, **Naive Bayes**, Ensemble
Build 3 classifiers from the chosen list.
Plot results.
Analyze results

Answer: I have selected **KNN**, **Neural Network and Naive Bayes**

About Data set:

The Insurance Company Benchmark (COIL 2000) from http://kdd.ics.uci.edu/ This data set used in the CoIL 2000 Challenge contains information on customers of an insurance company. The data consists of 86 variables and includes product usage data and socio-demographic data derived from zip area codes. The data was collected to answer the following question: Can you predict who would be interested in buying a caravan insurance policy?
Link of the raw data: http://kdd.ics.uci.edu/databases/tic/tic.html

Data Mining Task:

In this dataset, the data mining task is to create classification rules based on 85 attributes of the user. Such rule can help The Insurance Company to improve its prediction of sales and knowing their target customer.

This assignment shows how we can create regression and decision tree classifier in R.

Neural Network Classifier:

R Code:

KNN_Classifier.R | NaiveBayes.R | NeuralNetwork.R

Source on Save | Run | Source

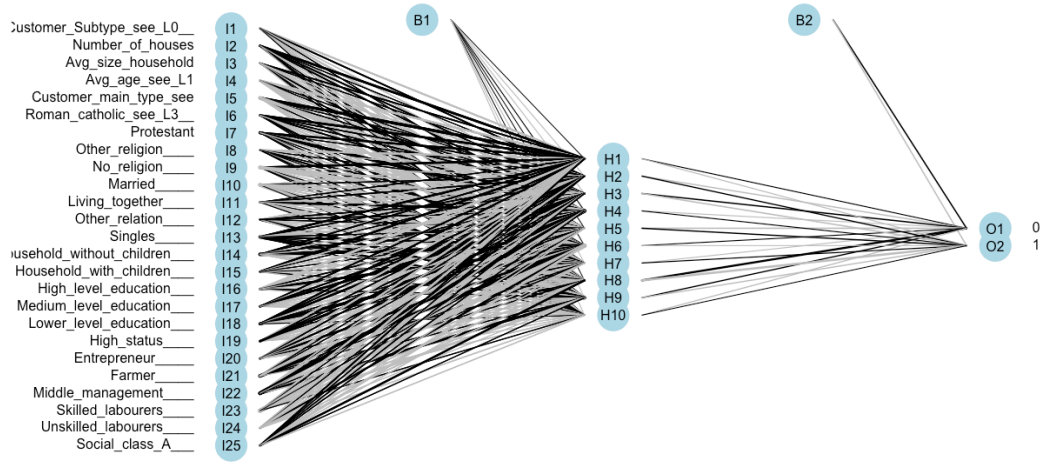
```r
library(nnet)
#Loading Training Data
Training.Data<-read.table("/Users/Ravi/Documents/Data Mining/Assignments/Assignment_7/data/trainingData.txt",header=T,sep="\t")

#Loading Test Data
Test.Data<-read.table("/Users/Ravi/Documents/Data Mining/Assignments/Assignment_7/data/TestData_Insurance.txt",header=T,sep="\t")

# Defining Class Attribute - Number_of_mobile_home_policies
class_label <- class.ind(Training.Data$Number_of_mobile_home_policies)

#creating  Neural Network
model = nnet(Training.Data, class_label , size=10, softmax=TRUE)

#Checking accuracy of the model
accuracy<-table(predict(model,Training.Data, type="class"),Training.Data$Number_of_mobile_home_policies)
print(accuracy)

#applying model on the Test Data
TestResult<-predict(model, Test.Data, type="class")
print(TestResult)

#plotting model this function requires clusterGeneration and devtools library
plot.nnet(model)
```

21:1 (Top Level) R Script

Output:

Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish



```
Console ~/
+ "Lower_level_education___","High_status____","Entrepreneur_____",
+ "Farmer_____","Middle_management____","Skilled_labourers____",
+ "Unskilled_labourers____","Social_class_A___" )], class_label , size=10, softmax=TRUE)
# weights:  282
initial  value 3060.011421
iter  10 value 1280.772352
iter  20 value 1239.395676
iter  30 value 1210.133986
iter  40 value 1191.119242
iter  50 value 1175.897324
iter  60 value 1159.139138
iter  70 value 1154.285614
iter  80 value 1146.821362
iter  90 value 1133.098038
iter 100 value 1119.797309
final  value 1119.797309
stopped after 100 iterations
> plot.nnet(model)
>
```

Contingency Table:

```
      0   1
  0 5473   4
  1    1 344
```

Above table shows how accurate is created classifier model.

KNN (K Nearest Neighbor) :

R code:

KNN_Classifier.R | Untitled1* | NaiveBayes.R | NeuralNetwork.R

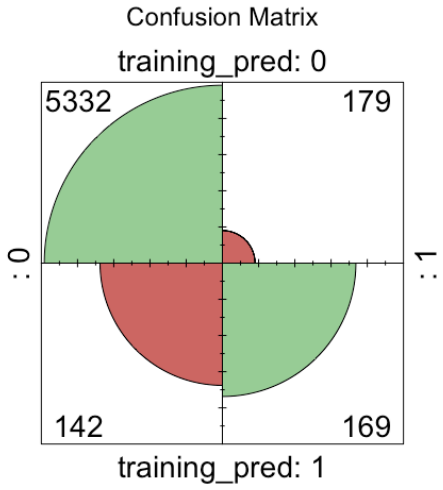
Source on Save | Run | Source

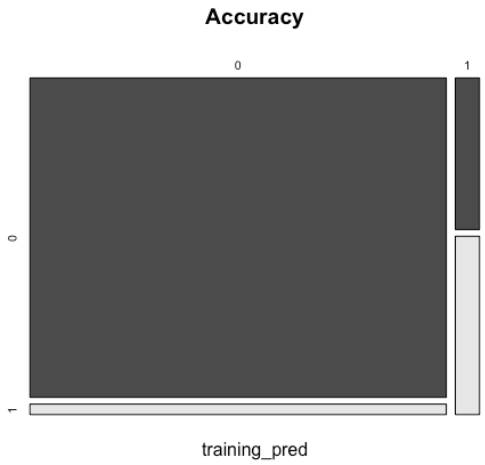
```
#loading Library
library(class)

#Loading Training Data
Training.Data<-read.table("/Users/Ravi/Documents/Data Mining/Assignments/Assignment_7/data/trainingData.txt",header=T,sep="\t")

#Loading Test Data
Test.Data<-read.table("/Users/Ravi/Documents/Data Mining/Assignments/Assignment_7/data/TestData_Insurance.txt",header=T,sep="\t")


#creating K nearest neighbor model to classify data
training_pred <- knn(train = Training.Data[,-86], test = Training.Data[,-86],cl = Training.Data[,86], k=2)

#calculating accuracy
accuracy<-table(training_pred,Training.Data$Number_of_mobile_home_policies)
#accuracy of the model
print(accuracy)
#printing graph of the accuracy
fourfoldplot(accuracy, color = c("#CC6666", "#99CC99"), conf.level = 0, margin = 1, main = "Confusion Matrix")
mosaicplot(accuracy, main = "Accuracy", color = TRUE)

#Applying Model on Test Data
test_pred <- knn(train = Training.Data[,-86], test = Test.Data,cl = Training.Data[,86], k=2)
print(test_pred)
```

19:90 (Top Level) R Script

Confusion Matrix :

```
> source('~/KNN_Classifier.R')

training_pred    0    1
            0 5332  179
            1  142  169
```





Naïve Bayes :

KNN_Classifier.R | NaiveBayes.R | NeuralNetwork.R

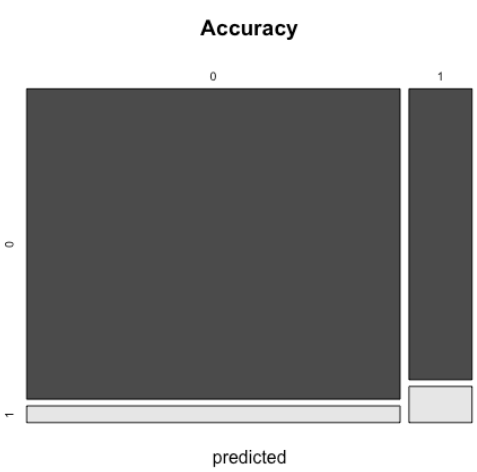
Source on Save | Run | Source

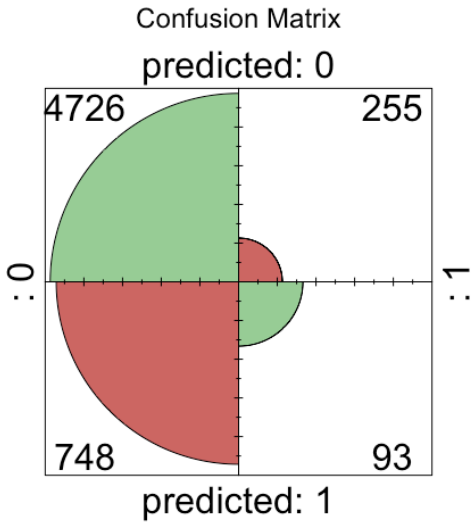
```
1
2  #install.packages("e1071");
3  library(e1071);
4
5  #Loading Training Data
6  Training.Data<-read.table("/Users/Ravi/Documents/Data Mining/Assignments/Assignment_7/data/trainingData.txt",header=T,sep="\t")
7
8  #Loading Test Data
9  Test.Data<-read.table("/Users/Ravi/Documents/Data Mining/Assignments/Assignment_7/data/TestData_Insurance.txt",header=T,sep="\t")
10
11 #creading navive Bayes Model with important attributes (removed trivial attributes for accuracy)
12 model <- naiveBayes(as.factor(Number_of_mobile_home_policies) ~Customer_Subtype_see_L0__ +
13                       Number_of_houses + Avg_size_household + Avg_age_see_L1 +
14                       Customer_main_type_see + Roman_catholic_see_L3__ + Protestant +
15                       Other_religion____ + No_religion____ + Married_____ + Living_together____ +
16                       Other_relation____ + Singles_____ + Household_without_children___ +
17                       Household_with_children___ + High_level_education___ +
18                       Medium_level_education___ +
19                       Lower_level_education___ + High_status____ + Entrepreneur_____ +
20                       Farmer_____ + Middle_management____ + Skilled_labourers____ +
21                       Unskilled_labourers____ + Social_class_A___ , data = Training.Data);
22
23 #Testing model on Training Data
24 predicted<-predict(model, Training.Data[,-86]);
25 print(predicted);
26
27 #Showing contigency Table to see accuracy of the model
28 accuracy<-table(predicted, Training.Data$Number_of_mobile_home_policies);
29 print(accuracy)
30
31
32 #Testing model on Test data
33 Predicted.Test<-predict(model, Test.Data);
34 print(Predicted.Test)
35
```

24:1 (Top Level) R Script

Confusion Matrix:

```
predicted    0    1
        0 4726  255
        1  748   93
   [1] 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
  [65] 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0
 [129] 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0
 [193] 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```





Analysis of Classifiers:

Based on above three classifiers, Neural Network performed well for the given data set. Also the accuracy for the Neural Network is very high than the other two classifiers.

We can observe in the confusion matrix that how Neural Network predicts correctly for the given data set. Hence, it is the best classifier for the data set.