Applied Data Mining: Homework #4

Due on Fill-in this please

Instructor: Hasan Kurban

Naimesh Chaudhari

October 6, 2017

In this homework, you will fit several classifiers to Orange Juice data set (OJ) and compare and discuss the results. The data set can be found in the ISLR package.

```
> library(ISLR)
> mydata <- OJ
> names(mydata)
> dim(mydata)
> View(mydata)
```

Problem 1

Discussion of Data

This data is tracking customer purchase of Citrus Hill or Minute Main OJ. With the purchase it trakces multiple characteristics such as store price discout. We can use this data to predict based on multiple parameters what is a customer most likely to buy.

R Code

Using R, show code that answers the following questions:

1. How many entries are in the data set? There are 1070 rows in the dataset.

Listing 1: Sample R Script With Highlighting

```r
nrow(mydata)
```

2. How many unknown or missing data are in the data set? There is no missing data.

Listing 2: Sample R Script With Highlighting

```r
mydata[mydata == ""] <- NA
mydata[mydata == "?"] <- NA
table(is.na(mydata))  #No NA's
```

3. How many Citrus Hill and Minute Maid Orange Juice identifiers are there? There are 653 Citrus Hill and 417 Minute main identifiers.

Listing 3: Sample R Script With Highlighting

```r
table(mydata$Purchase)
```

Problem 2

Create a training data set containing a random sample of 900 data points and a test set containing the remaining observations. Name the training data and test data as mydata.training and mydata.testing, respectively. Place the R code below. You will use mydata.training and mydata.testing to answer the rest of the questions. Thus, create them once and use mydata.training to train the models (classifiers) and mydata.testing to test the models. Purchase variable (1st variable in the data) is the response and the other variables are predictors. In addition to answering homework questions, you are encouraged to tune the parameters of the classifiers and test the parameters that are not included in the homework questions.

Problem 2 continued on next page...

R Code

Listing 4: Sample R Script With Highlighting

```
library(ISLR)
library(DMwR2)
library(rpart.plot)
set.seed(1234)
rndSample <- sample(1:nrow(mydata), 900)
mydata.training <- mydata[rndSample, ]
mydata.test <- mydata[-rndSample, ]
```

Problem 3

Fit three decision trees to mydata.training with $se = 0, 0.5, 1$. Visualize the trees in R. Use trees to classify mydata.testing. Calculate a confusion matrix and accuracy value for each model to evaluate the models. Discuss the results, i.e., which model works best? explain se parameter. How does se parameter affect the results?

R Code

Place the R codes below

1. Training code:

Listing 5: Sample R Script With Highlighting

```
ct1 <- rpartXse(Purchase ~ ., mydata.training, se=0)
ct2 <- rpartXse(Purchase ~ ., mydata.training, se=0.5)
ct3 <- rpartXse(Purchase ~ ., mydata.training, se=1)
par( mfrow=c(1,3))
prp(ct1,type=0,extra=101, main = 'CT1 SE = 0')
prp(ct2,type=0,extra=101, main = 'CT2 SE = 0.5')
prp(ct3,type=0,extra=101, main = 'CT3 SE = 1')
```

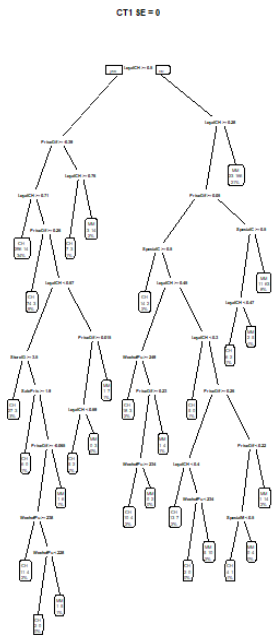
2. Testing code:

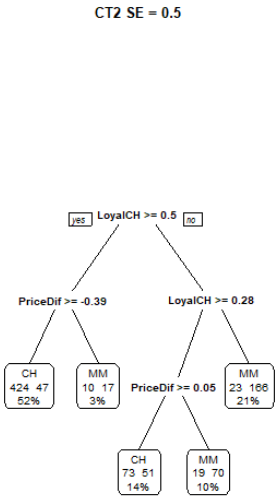
Listing 6: Sample R Script With Highlighting

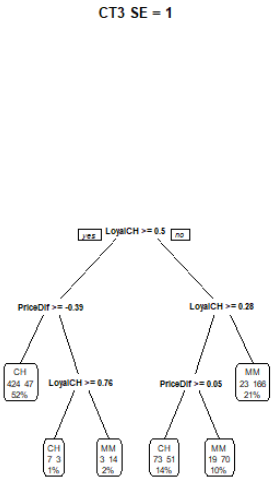
```
ps1 <- predict(ct1, mydata.test, type="class")
ps2 <- predict(ct2, mydata.test, type="class")
ps3 <- predict(ct3, mydata.test, type="class")


cm1 <- table(ps1, mydata.test$Purchase) #confusion matrix for cm1
cm2 <- table(ps2, mydata.test$Purchase) #confusion matrix for cm2
cm3 <- table(ps3, mydata.test$Purchase) #confusion matrix for cm3



100*(1-sum(diag(cm1))/sum(cm1))  # the error rate is 18.25%
100*(1-sum(diag(cm2))/sum(cm2))  # the error rate is 18.82%
100*(1-sum(diag(cm3))/sum(cm3))  # the error rate is 18.25%
```

Problem 3 continued on next page...

Tree Figures







Results and Discussion

Both SE = 0 and SE = 1 had similar results for me. Generally they had and 18.235 percent error. SE 0.5 did the worst for me with a percent error of 18.82 . From looking at the charts we can see that as SE increased the number of branches within a decision tree decreased. SE measures the standard error threshold within each of the models.

Problem 4

In this question, you are asked to train two SVMs over mydata.train and test the models over mydata.testing. (1) Train an SVM with the default settings (radial kernel with constraints violations of cost of 1), (2) Train another SVM with tuning the parameters as follows: cost=3, kernel="polynomial", degree=3. Create a confusion matrix and calculate the accuracy value for each model. Discuss the results, i.e., Which one performed better? explain the parameters.

R Code

Place the R codes below

1. Training code:

Listing 7: Sample R Script With Highlighting

```
library(e1071)
svm1 <- svm(Purchase ~ ., data = mydata.training, cost = 1, kernal = 'radial
    basis')
svm2 <- svm(Purchase ~ ., data = mydata.training, kernal = 'polynomial',
    degree = 3 , cost = 3)
```

Problem 4 continued on next page...

2. Testing code:

Listing 8: Sample R Script With Highlighting

```
spred1 <- predict(svm1,mydata.test)
spred2 <- predict(svm2,mydata.test)
tbl1 <- table(spred1,mydata.test$Purchase)
tbl2 <- table(spred2,mydata.test$Purchase)
100*(1-sum(diag(tbl1))/sum(tbl1)) #17.64%
100*(1-sum(diag(tbl2))/sum(tbl2)) #16.47%
```

Results and Discussion

The second model definitely performed better. It had an error difference of about 1 percent.

The parameters kernel represents different ways of calculating the distance to predict the proper class.

The parameter cost stands for the cost of constraint violation. It generally defines the range and how soft of hard algorithm can be.

The parameter degree represents the type of polynomial when using polynomial kernel. For example 2 = quadratic.

Problem 5

Fit two Artificial Neural Networks (ANNs) to mydata.training as follows: Set the trace=FALSE and maxit=1000 for both ANNs. The size parameter for the first ANN is 10 (size=10) and the second one is 50 (size =50). Test the ANNs over mydata.testing and discuss the results. Which one performed better? explain the parameters? Visualize the ANNs.

R Code

Place the R codes below

1. Training code:

Listing 9: Sample R Script With Highlighting

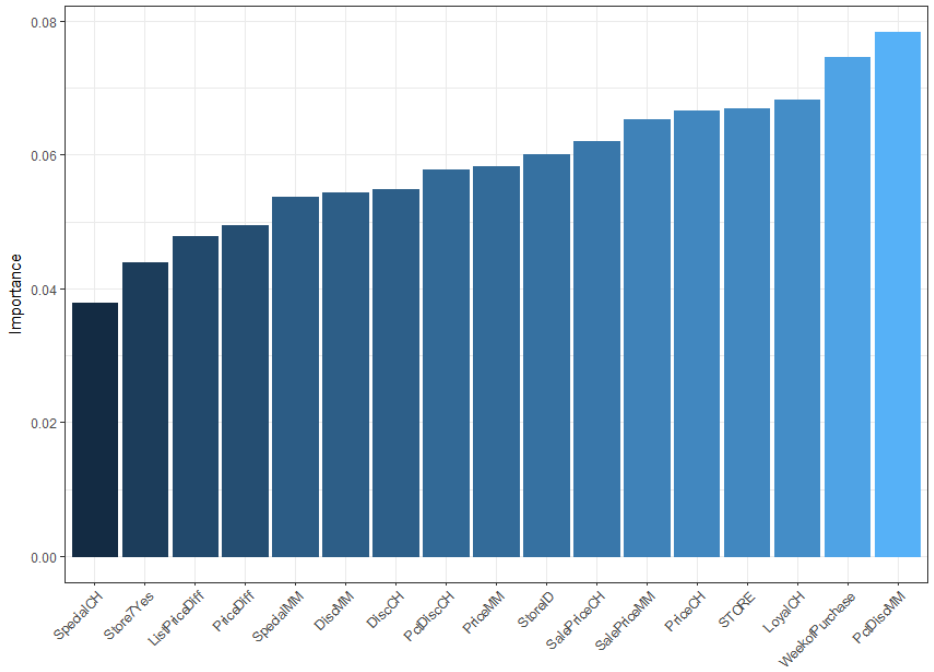
```
library(nnet)
library(ggplot2)
nr1 <- nnet(Purchase ~ ., mydata.training,trace=FALSE, size=10, maxit=1000)
nr2 <- nnet(Purchase ~ ., mydata.training,trace=FALSE, size=50, maxit=1000)
garson(nr1) + theme(axis.text.x = element_text(angle = 45, hjust = 1))
plot(nr1, main = "NR1 Size = 10")
garson(nr2) + theme(axis.text.x = element_text(angle = 45, hjust = 1))
plot(nr2, main = "NR1 Size = 50")
```

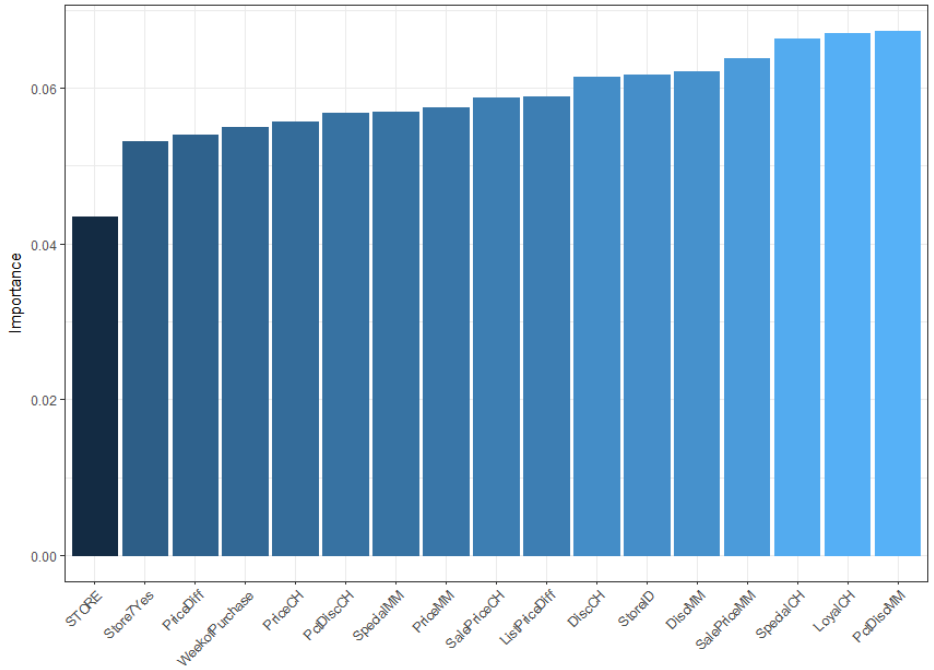
2. Testing code:

Listing 10: Sample R Script With Highlighting

```
npred1 = predict(nr1,mydata.test, type = 'class')
npred2 = predict(nr2,mydata.test,type = 'class')
tb1 = table(npred1,mydata.test$Purchase)
tb2 = table(npred2,mydata.test$Purchase)
100*(1-sum(diag(tb1))/sum(tb1))     #38.82%
100*(1-sum(diag(tb2))/sum(tb2))    #16.47%
```

ANN Figures





Results and Discussion

The ANN with size 50 performed much better than the ANN with size 10.

The parameter size represents the number of nueros on the hidden network.

The parameter trace is a Boolean for trace optimization.

The parameter maxit represents the maximum number of iteration the algorithm will run.

Problem 6

In this problem, you are asked to train a deep learning model on mydata.training with defaults settings. Test the model over mydata.testing. Calculate the confusion matrix and accuracy value for the model.

R Code

Place the R code below

1. Training code:

Listing 11: Sample R Script With Highlighting

```
library(h2o)
h2oInstance <- h2o.init(ip="localhost")
trH  <- as.h2o(mydata.training,"trH") #
tsH <- as.h2o(mydata.test,"tsH")
```

2. Testing code:

Listing 12: Sample R Script With Highlighting

```
mdl <- h2o.deeplearning(x=2:18, y=1, training_frame=trH)
preds <- h2o.predict(mdl,tsH)[,"predict"]
cm <- table(as.vector(preds), as.vector(tsH$Purchase))
100*(1-sum(diag(cm))/sum(cm)) #15.88%
```

Results and Discussion

The accuracy value of deep learning is 15.89 percent

Problem 7

Compute the classifiers trained in question 3,4,5,6 for Orange Juice data set. Discuss the results.

Results and Discussion

It looks like the error rate for all the different results was generally between 18 to 16 percent. There was one huge error rate when doing ANN with size = 10. The best result was the deep learning method that had a results of 15.88 percent. One thing to note is deep learning also took the longest. If the error rate is flexible I would suggest using either SVM or ANN, if not then deep learning would be the best model to use for this scenario.