Mid Term Take Home Test
CSCE 587
Fall 2016
Due: 11/03/2016 via Dropbox
Name: Sendurr Selvaraj
VIP ID: 00323540

Part 1: Setting up the data.

Install the package "titanic". It contains two data sets: titanic_train and titanic_test. We will only be using titanic_train. The data set titanic_train consists of 891 rows of data in which each row corresponds to one of 891 passengers on the titanic. Each row consists of 12 columns. When viewed as a classification problem, column 2 (Survived) specifies the class of each observation/instance. The remaining columns are attributes that might be used to infer column 2. We will focus on Columns 3, 5, 6, 7, & 8 (Pclass, Sex, Age, SibSp, and Parch). See https://cran.r-project.org/web/packages/titanic/titanic.pdf for a description of each feature.

1) Start by casting column 2 so that it is interpreted by R as a factor instead of an integer value.

Code Snippet:

```
library(titanic)
titanic_train
s= factor(titanic_train[,2])
titanic_train[,2] = s
```

2) Next, make your life easier by creating a subset consisting only of columns 2, 3, 5, 6, 7 & 8. The resulting set is comprised of 891 observations of 6 variables.

Code Snippet:

```
subset_temp = titanic_train[,c( 2, 3, 5, 6, 7, 8)]
```

3) Partition the set of 891 instances/observations from the previous step into a new test set comprised of the first 291 rows and a new training set comprised of the remaining 600 rows. Call the test set test_set, and the training set train_set.

Code Snippet:

```
test_set = subset_temp[c(1:291),]
test_set
train_set = subset_temp[c(292:891),]
train_set
```

Part 2: Naive Bayes Analysis of the titanic dataset

1) Using the naiveBayes() method from the package e1071, train a Naive Bayes model using the train_ set produced in part 1. Review the slides on Naïve Bayes (in particular the last several slides that cover the in-class lab). The arguments to naiveBayes() are the independent features (columns 2,3,4,5,6 of train_set) and the dependent feature (column 1), the feature you are predicting. Review the documentation for Naïve Bayes to see how to specify a model formulae http://ugrad.stat.ubc.ca/R/library/e1071/html/naiveBayes.html

Code Snippet:

```
 m1 <-naiveBayes(train_set[,2:6],train_set[,1])
```

2) Continuing with the training data set train_set, create the confusion matrix using the table command as in slide 61 of the Naïve Bayes slides. Note: R will complain about NAs introduced by coercion.

Code Snippet:

```
table(predict(m1,train_set[,2:6], train_set[,1))
```

Output:

```
    0   1
 0 270  94
 1  93 143
```

3) Calculate the model accuracy from the model by summing up the correct classifications and dividing by the total number of observations. Recall that the table() command produces a confusion matrix which is a square matrix. The diagonal entries are the correct classifications. You can calculate the accuracy by summing the diagonal entries and dividing by the sum of all of the entries.

Answer:

For train_set

True Positive = 270, True Negative = 143 , False Negative = 94, False Positive = 93,

Accuracy = (270 + 143) / (270 + 143 + 94 + 93)   = 0.68

4) Repeat steps 2 & 3, using the testing data set, test_set.

Code Snippet:

```
m2 <-naiveBayes(test_set[,2:6],test_set[,1])
table(predict(m2,test_set[,2:6]), test_set[,1)
```

Output:

```
    0   1
 0 148  66
 1   38  39
```

Answer:

For test_set

True Positive = 148, True Negative = 39 , False Negative = 66, False Positive = 38,

Accuracy = (148 + 39) / (148 + 39 + 66 + 38)   = 0.64

Part 3: Decision Tree Analysis of the titanic dataset

1) Using the rpart() method from the package rpart, train a Decision Tree model using the training data set train_set produced in part 1. Assign this decision tree model to the variable tree1, i.e., tree1 <- rpart(......) Review the slides from the Decision Tree Lab. As in Part 2, the arguments to rpart() are the independent features (columns 2-6) and the dependent feature (column 1), the feature you are predicting. In addition, use the parameter values cp=0.02 and minsplit=2.

Code Snippet:

```
library(rpart)
tree1 <-rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch, method = "class",
            data=train_set, control =rpart.control(minsplit=2, cp=0.002))
tree1
```

2) Using the training data set train_set produced in part 1, create the confusion matrix using the table command along with tree1, the decision tree model you produced in step 1 of this part. Note: to make this work you will need to include a 3rd parameter to the predict() function. This parameter is type='class'

Code Snippet:

```
table(predict(tree1,train_set[,2:6],type="class"), train_set[,1)))
```

Output:

```
    0   1
 0 352  42
 1  11 195
```

3) Calculate the model accuracy from the model by summing up the correct classifications and dividing by the total number of observations. Recall that the table() command

produces a confusion matrix which is a square matrix. The diagonal entries are the correct classifications. You can calculate the accuracy by summing the diagonal entries and dividing by the sum of all of the entries.

Answer:

For train_set

True Positive = 352, True Negative = 195 , False Negative = 42, False Positive = 11,

Accuracy = (352 + 195) / (352 + 195 + 42 + 11)   = 0.91

4) Repeat steps 2 & 3, using the testing data set, test_set.

Code Snippet:

```
tree2 <-rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch, method = "class",
          data=test_set, control =rpart.control(minsplit=2, cp=0.002))
tree2
table(predict(tree2,test_set[,2:6],type="class"), test_set[,1)
```

Output:

```
    0   1
 0 181  15
 1   5 90
```

Answer:

For test_set

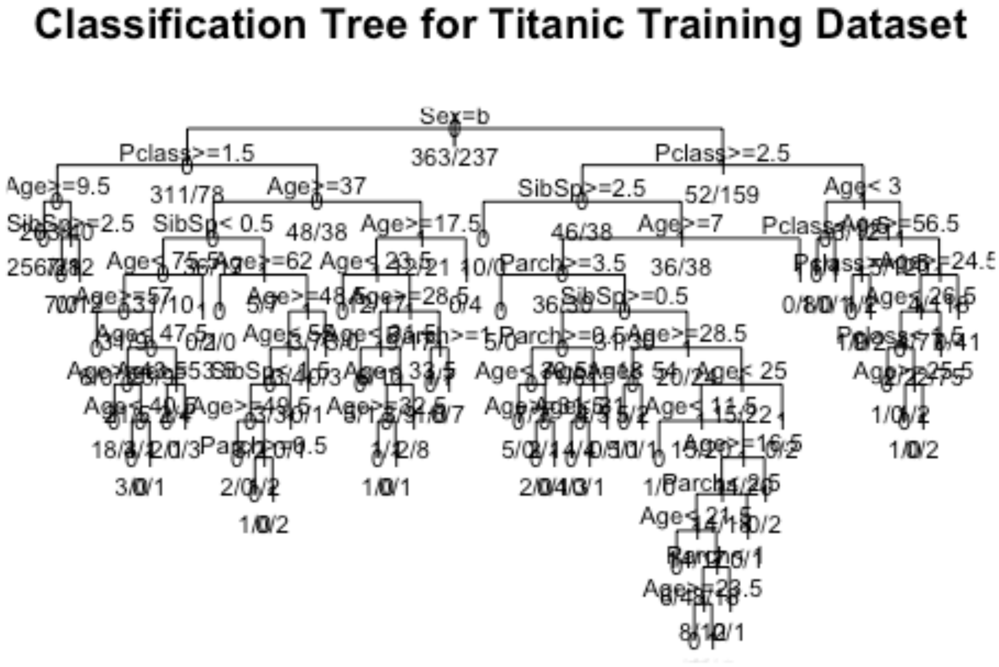
True Positive = 181, True Negative = 90 , False Negative = 15, False Positive = 5,

Accuracy = (181 + 90) / (181 + 90 + 15 + 5)   = 0.93

Part 4: Graduate Students Only: Comparison of Part 2 & Part 3 results

Explain why the Decision Tree results are somewhat better than the Naïve Bayes Results. (Hint: for insight, plot and label the decision tree.)

Answer:


Classification Tree for Titanic Training Dataset

As shown in the decision tree plot, the decision tree represents a clear prioritization of the variables on which the outcome is dependent. I.e., the best attribute forms the root of the tree followed by the next attribute forming the node. For the titanic data set, the decision tree shows that the no of survived people is most dependent on the sex, followed by age, Sibp and Parch. Also the range of the variable affecting the outcome is also visible. While going down the tree from node to node, more concrete and finite value of the dependent variables are obtained. Decision trees help us generate rules for the outcome. Whereas, Naïve Bayes helps predict the outcome of the event, however does not provide additional details as learnt from decision trees.

Code Snippet:

```
plot(tree1, uniform=TRUE, main="Classification Tree for Titanic Training Dataset")
text(tree1, use.n=TRUE, all=TRUE, cex=.7)

Splot(tree2, uniform=TRUE, main="Classification Tree for Titanic Test Dataset")
text(tree2, use.n=TRUE, all=TRUE, cex=.7))
```

Overall Code:

```
#***** Part 1: Setting up the data. *******#
library(titanic)
titanic_train
s= factor(titanic_train[,2])
titanic_train[,2] = s
subset_temp = titanic_train[,c( 2, 3, 5, 6, 7, 8)]
test_set = subset_temp[c(1:291),]
test_set
train_set = subset_temp[c(292:891),]
train_set

#***** Part 2: Naive Bayes Analysis of the titanic dataset  *******#
library(e1071)
m1 <-naiveBayes(train_set[,2:6],train_set[,1])
table(predict(m1,train_set[,2:6]),train_set[,1])

m2 <-naiveBayes(test_set[,2:6],test_set[,1])
table(predict(m2,test_set[,2:6]),test_set[,1])

#***** Part 3: Decision Tree Analysis of the titanic dataset   *******#
library(rpart)
tree1 <-rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch, method = "class",
          data=train_set, control =rpart.control(minsplit=2, cp=0.002))
tree1
table(predict(tree1,train_set[,2:6],type="class"),train_set[,1])

tree2 <-rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch, method = "class",
          data=test_set, control =rpart.control(minsplit=2, cp=0.002))
tree2
table(predict(tree2,test_set[,2:6],type="class"),test_set[,1])

#***** Part 4: Graduate Students Only: Comparison of Part 2 & Part 3 results   *******#

plot(tree1, uniform=TRUE, main="Classification Tree for Titanic Training Dataset")
text(tree1, use.n=TRUE, all=TRUE, cex=.7)

plot(tree2, uniform=TRUE, main="Classification Tree for Titanic Test Dataset")
text(tree2, use.n=TRUE, all=TRUE, cex=.7)
```