Tien Thuy Ho

CSCE 587: Big Data Analytics

Midterm Exam

**Part 1: Setting up the data**

# Import the data

dataset <- read.csv("~/Documents/College/Computer Science/CSCE 587/tae.dat")

# Cast column 6 as a factor

dataset$Class <- factor(dataset$Class)

# Partition the dataset into a test set and a training set

test\_set <- dataset[1:51, ]

training\_set <- dataset[52:151, ]

**Part 2: Naïve Bayes Analysis**

# Install and load the package *e1071*

install.packages("e1071")

library(e1071)

# Train a Naïve Bayes model using the training set

training\_model <- naiveBayes(training\_set[, 1:5], training\_set[, 6])

# Create the confusion matrix for the Naïve Bayes model

confusion\_matrix <- table(predict(training\_model, test\_set[, 1:5]), test\_set[, 6])

|  |  |  |  |
| --- | --- | --- | --- |
|  | **1** | **2** | **3** |
| **1** | 9 | 10 | 9 |
| **2** | 1 | 2 | 4 |
| **3** | 1 | 2 | 13 |

# Calculate the model accuracy

total\_observations <- sum(colSums(confusion\_matrix))

correct\_classifications <- sum(diag(confusion\_matrix))

model\_accuracy <- correct\_classifications / total\_observations

# The accuracy for the Naïve Bayes model on the test set is 0.4705882

**Part 3: Decision Tree Analysis**

# Install and load the package *rpart*

install.packages("rpart")

library("rpart")

# Train a Decision Tree model using the training set

tree1 <- rpart(Class ~ Nat + Inst + C + Sem + Size, method = "class", data = training\_set, control = rpart.control(minsplit = 2, cp = 0.02))

# Create the confusion matrix

tree1\_confusion\_matrix <- table(predict(tree1, test\_set[, 1:5], type="class"), test\_set[, 6])

|  |  |  |  |
| --- | --- | --- | --- |
|  | **1** | **2** | **3** |
| **1** | 8 | 3 | 6 |
| **2** | 2 | 10 | 5 |
| **3** | 1 | 1 | 15 |

# Calculate the model accuracy

tree1\_total\_observations <- sum(colSums(tree1\_confusion\_matrix))

tree1\_correct\_classifications <- sum(diag(tree1\_confusion\_matrix))

tree1\_model\_accuracy <- tree1\_correct\_classifications / tree1\_total\_observations

# The accuracy for the Decision Tree model on the test set is 0.6470588.

**Part 4: Test data results vs. Training data results**

# Create the confusion matrix using the model **tree1** from Part 3 and the training set

training\_confusion\_matrix <- table(predict(tree1, training\_set[, 1:5], type="class"), training\_set[, 6])

|  |  |  |  |
| --- | --- | --- | --- |
|  | **1** | **2** | **3** |
| **1** | 28 | 6 | 2 |
| **2** | 5 | 25 | 7 |
| **3** | 5 | 5 | 17 |

# Calculate the accuracy of the model **tree1** on the training set

training\_total\_observations <- sum(colSums(training\_confusion\_matrix))

training\_correct\_classifications <- sum(diag(training\_confusion\_matrix))

training\_model\_accuracy <- training\_correct\_classifications / training\_total\_observations

# The accuracy for the model tree1 on the training set is 0.7.

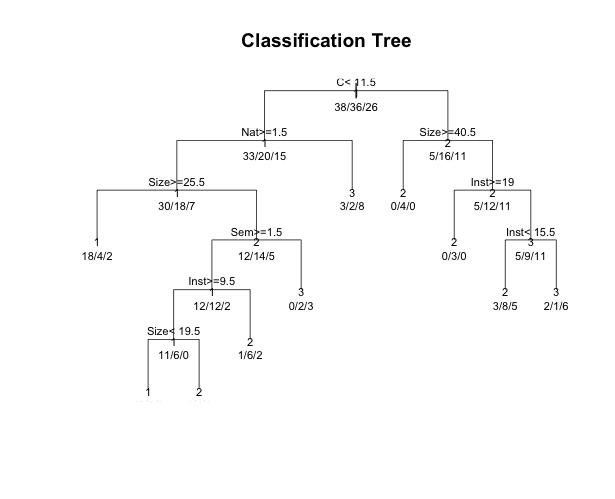
# The model **tree1** has a slightly higher degree of accuracy on the training data set than on the test set. This difference in accuracy results from the split of the original data in which the training set and the test set do not have equal proportions of classifications. We have six classifications to test the accuracy of the model (1-1, 2-1, 3-1, 1-2, 2-2, 3-2, 1-3, 2-3, 3-3). The first number is the predicted outcome by the model, and the second number is the actual outcome. The comparison for the classification proportions between the test set and the training set is displayed in the following table.

|  |  |  |
| --- | --- | --- |
|  | **Test Set** | **Training Set** |
| **1-1** | 16% | 28% |
| **2-1** | 6% | 6% |
| **3-1** | 12% | 2% |
| **1-2** | 4% | 5% |
| **2-2** | 20% | 25% |
| **3-2** | 10% | 7% |
| **1-3** | 2% | 5% |
| **2-3** | 2% | 5% |
| **3-3** | 29% | 17% |

As seen, all classifications have approximately equal proportions in both data subsets except for 1-1, 3-1, and 3-3. The test set is missing the classification for 1-1 whereas the training set is missing the classifications for 3-1 and 3-3. This biased division of data in which classifications are not equally represented in both subsets results in the discrepancy of the model accuracy for each subset. Moreover, since the model is trained using the training data set, it is more represented by the training model and therefore fits it better than the test set which the model has not seen before.

**Part 5: Comparison of Part 2 & Part 3 results**

# The plot for the decision tree model for the test set in part 3



The plot shows that there are some dependencies between different predictor features. For instance, the feature “Size” depends on the feature “Nat” when the values for “Nat” are greater than or equal to 1.5. Similarly, the feature “Sem” depends on “Size” when the values of “Size” are less than 25.5. This decision tree naturally shows the correlation between features by making the dependent features the descendants of another feature. This variable correlation interferes with the accuracy of the Naïve Bayes classifier in which predictor features are assumed to be independent. Therefore, the Decision Tree results are somewhat better than the Naïve Bayes results for this dataset because the Decision Tree classifier can handle correlation between the variables.