**COSC 757 Data Mining Lab 5 Classification**

**Spring 2016**

Before moving forward in this section of the lab, make sure to download and install the rpart package in R. In this lab we will use the popular iris dataset from the UCI Machine Learning Repository.

1) Load data into R

> data(iris)

2) Display the internal structure of the dataset

> str(iris)

**Decision Tree Classification Using rpart**

3) The rpart package uses the CART method to produce decision tree classifiers. First, install the rpart package by issuing the following command:

> install.packages(“rpart”)

Once the rpart package is installed, load the package:

>library(rpart)

4) First we need to create our training and testing dataset. Here we will set our random seed using the set.seed() function. This allows us to be able to reproduce the result of the random sample. Then the sample() function is used to randomly sample the dataset to produce two samples. The first (trainData) is the training dataset and consists of 70% of the tuples. The second (testData) is the testing dataset and consists of 30% of the tuples. Do create the testing and training datasets, issue the following commands

> set.seed(1234)

> ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7,0.3))

Now examine ind and you should see that you will have a vector where 70% of the values are 1 and 30% are 2. We will use this vector to select our training and testing datasets with the following command.

> trainData <- iris[ind==1,]

> testData <- iris[ind==2,]

5) Now we can build the tree using the rpart() function.

> iris\_rpart <- rpart(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = trainData)

6) We can now evaluate the tree by viewing the cross validation error at each split of the tree. This information can be used to prune the tree. In the case of the iris dataset, because the tree is so small, there is no need to prune. The error matrix can be viewed by issuing the following command:

> printcp(iris\_rpart)

We can also view this data using a graph:

> plotcp(iris\_rpart)

7) The tree can be viewed by using the following commands:

> plot(iris\_rpart)

> text(iris\_rpart, use.n=TRUE)

8) Now we can test the classification model using the predict() function

> iris\_pred <- predict(iris\_rpart, newdata = testData)

**Decision Tree Classification Using party**

9) The party package is another decision tree package for R that has some nice features. The algorithm used in party uses recursive partitioning and conditional inference. If you are interested in reading more about the algorithm, the article can be found here:

<http://amstat.tandfonline.com/doi/pdf/10.1198/106186006X133933>

Now download and install the party package.

10) We can start with the same training and testing datasets that we produced in the previous example. To create the tree issue the following command:

> iris\_ctree <- ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = trainData)

11) One nice attribute of the party package is that it includes a method to compute a confusion table

> table(predict(iris\_ctree), trainData$Species)

12) It is also easy to plot the decision tree using the plot() function.

> plot(iris\_ctree)

13) Finally, we can easily evaluate the test dataset using a confusion table

> testPred <- predict(iris\_ctree, newdata = testData)

> table(testPred, testData$Species)

**Random Forest**

14) Download and install the randomForest package

15) Fit a random forest model to the iris dataset

> fit<-randomForest(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width, data = trainData)

16) View the results:

> print(fit)

17) View the attribute importance of the model

> importance(fit)

**Naïve Bayes Classification Using e1071**

18) Download and install the e1071 package

19) Train and evaluate the naïve Bayesian classifier

> classifier <- naiveBayes(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = trainData)

> pred <- predict(classifier,testData[,-5])

> table(pred,testData$Species)