Hannah Roach
3/20/2019

Lab Exercise 9: Decision Trees

Purpose:	This lab is designed to investigate and practice Decision Tree (DT) models covered in the course work. After completing the tasks in this lab you should able to: • Use R functions for Decision Tree models • Predict the outcome of an attribute based on the model

Tasks:	Tasks you will complete in this lab exercise include: • Use the R –Studio environment to code Decision Tree Models • Build a Decision Tree Model based on data whose schema is composed of attributes • Predict the outcome of one attribute based on the model

Workflow Overview

1. Set the Working Directory
2. Read in the Data
3. Build the Decision Tree
4. Plot the Decision Tree
5. Prepare Data to Test the Fitted Model
6. Predict a Decision from the Fitted Model

LAB Instructions

Step	Action
1	Log in with GPADMIN credentials on to R-Studio.

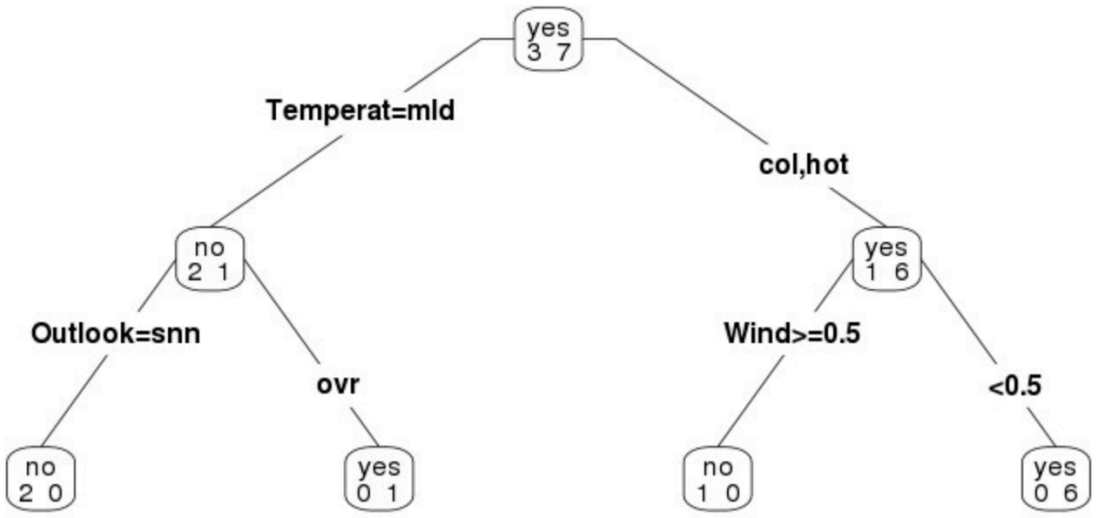
2	**Set the Working Directory:**
	
	1. Execute the command:
	
	`setwd("~/LAB09")`
	
	`> library("rpart")`
	
	`> library("rpart.plot")`

3	**Read in the Data:**
	
	• Use a data table with columns for data attributes : Play, Outlook, Temperature, Humidity and Windy
	• A Decision Tree allows for predicting the values of the attribute Play, given that you know the values for attributes like Outlook, Humidity and Windy.
	
	1. Read in the data from the "Dtdata.csv" file in the working directory and display the contents:
	
	`> #Read the data`
	`> play_decision`
	`> play_decision<- read.table("DTdata.csv",header=TRUE,sep=",")`
	`> play_decision`
	
	2. How many observations did you read in?
	10
	3. How many variables (attributes) did you read in?
	5 – Play, Outlook, Temperature, Humidity, and Wind
	4. Use the command "summary" for a detailed list of the table object you read in
	
	`summary(play_decision)`
	
	5. Review the results. (The Summary is located in the console window.)
	
	`> summary(play_decision)`
	`  Play       Outlook  Temperature   Humidity    Wind`
	`no :3   overcast:2   cool:5      high  :4   Mode :logical`
	`yes:7   rainy    :4   hot :2      normal:6   FALSE:7`
	`        sunny    :4   mild:3                 TRUE :3`
	`                                            NA's :0`
	`>`

Step	Action

4	**Build the Decision Tree:**
	
	Use the "rpart" package in R for classification by Decision Trees. The RPart Programs build classification or regression models of a very general structure using a two stage procedure; the resulting models can be represented as binary trees.
	
	1. Use the following rpart commands to grow a Decision Tree:
	
	`rpart (formula, data=, method=, control=)`
	
	• **formula** is in the format: outcome ~ predictor1+predictor2+predictor3+ect.
	• **data=** specifies the dataframe
	• **method=** "class" for a classification tree "anova" for a regression tree
	• **control=** optional parameters for controlling tree growth. For example, control=rpart.control(minsplit=30, cp=0.001) requires that the minimum number of observations in a node be 30 before attempting a split and that a split must decrease the overall lack of fit by a factor of 0.001 (cost complexity factor) before being attempted.
	• **parms=** Optional parameters for the splitting function. Anova splitting has no parameters. Poisson splitting has a single parameter, the coefficient of variation of the prior distribution on the rates. The default value is 1. Exponential splitting has the same parameter as Poisson.
	For classification splitting, the list can contain any of: the vector of prior probabilities (component prior), the loss matrix (component loss) or the splitting index (component split). The priors must be positive and sum to 1. The loss matrix must have zeros on the diagonal and positive off-diagonal elements. The splitting index can be gini or information. The default priors are proportional to the data counts, the losses default to 1, and the split defaults to gini.
	
	The "Play" attribute is the outcome that will be predicted.
	
	2. Use the command:
	
	`> fit <- rpart(Play ~ Outlook + Temperature + Humidity + Wind, method="class", data=play_decision,`
	`+ control=rpart.control(minsplit=1)`
	`+ parms=list(split='information')`
	
	3. You can now display "fit" and review the results:
	
	`> summary(fit)`
	Note that the leaf nodes information includes both the class label and the class probabilities (P(no), P(yes))

Step	Action

5	**Plot the Decision Tree:**
	
	1. Review the arguments for rpart.plot function. Type in:
	`> ?rpart.plot`
	
	We will use the arguments “type” and “extra” in our plot.
	
	2. Type in the following :
	`> rpart.plot(fit, type=4, extra=1)`
	3. Review the Decision Tree plot on the graphics window.
	***Screenshot
	
	 

Step	Action

6	**Prepare Data to Test the Fitted Model:**
	
	You must use "fit" for a new set of data to create predictions from the DT:
	
	Play Decision ? Outlook rainy Temperature mild Humidity high Wind FALSE
	
	1. "newdata" is a data frame object and can be built for our test data. Type in the following statement:
	
	`newdata <- data.frame(Outlook="rainy",Temperature="mild",Humidity="high",Wind=FALSE)`
	
	2. Review the "newdata" displaying the dataframe
	
	`> newdata`
	
	3. The data displayed as follows:
	
	`Outlook Temperature Humidity Wind`
	`1 rainy mild high FALSE`

7	**Predict a Decision from the Fitted Model:**
	
	The "predict" function is used to generate predictions from a fitted rpart object.
	• "type" is a character string denoting the type of the predicted value
	• Use both "prob" and "class" to predict from a Decision Tree model
	
	`predict(object, newdata = list(),`
	`type = c("vector", "prob", "class", "matrix"))`
	1. The **type="prob"** gives the class probabilities for the prediction for newdata Type in
	`> predict(fit,newdata=newdata,type="prob")`
	
	2. Repeat the prediction with type="class"
	
	`> predict(fit,newdata=newdata,type="class")`
	
	Review the results.
	3. What is the prediction for the "newdata"?
	
	`> predict(fit,newdata=newdata,type=c("class"))`
	`1`
	`no`
	`Levels: no yes`

End of Lab Exercise