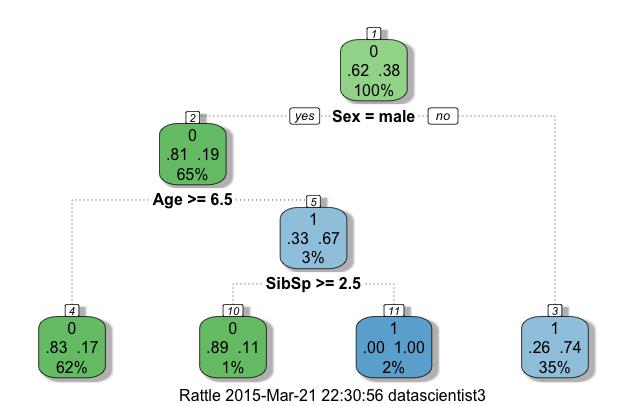
**A Short Introduction To Recursive Partitioning & Decision Trees:**

David Goody

1. Decision trees by splitting the data multiple times based on individual variables to try and classify the data. For example with the simplified tree below looking at whether passengers on the Titanic survived:
   1. We first divide up the dataset into 2 groups based on gender. Females have a 74% chance of survival and made up 35% of the passengers.
   2. Males are then subdivided by age. Males who are aged 7 or over only have a 17% chance of survival. They made up 62% of passengers
   3. Males aged under 7 are subdivided by the number of siblings and spouses they have. Those with 3 or more have an 11% chance of survival whilst those with 2 or fewer have a 100% chance of survival



1. In the plot above the groups who are more likely to survive are shown in blue whilst those who are less likely to survive are shown in green. The tree is built by finding the variable that creates the greatest difference between the two resulting groups. Looking at all the passengers on the Titanic there was a 38% chance of survival. If you split by gender you find that females had a 74% chance of survival whilst males had a 19% chance of survival. The difference between this 74% and 19% is greater than could be achieved by any other indicator. The model then continues to apply the process to continue sub-dividing the data. An ideal model would end up with the final groups having either a 100% survival rate or a 0% survival rate.
2. One of the main advantages of this approach is that it doesn’t have to fit a linear relationship across each variable in the same way the linear regression models do. All it needs to do it look at each variable in turn and identify the point at which splitting it is most helpful in identifying the at risk variables. This means it can tailor itself to the data more closely than logistic, lasso or ridge regression.
3. The decision tree can partition each variable multiple times (for example splitting the fare a passenger on the Titanic paid multiple times – for example £0-£10, £11-£50 and over £50). One downside of the decision tree approach is that is creates cliff-edges. This means a school with 151 pupils may not be flagged as a risk but one with 150 pupils could be. This can lead to over-fitting or the model being overly specific.
4. The diagram below gives a graphical example of how recursive partition works for a case with two variables. Each split on the decision tree is shown as being equivalent to a slice on a rectangular cake.

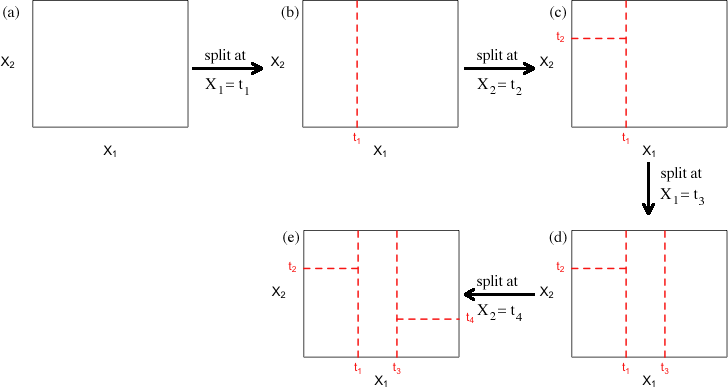


image taken from <http://www.unc.edu/courses/2010spring/ecol/562/001/docs/lectures/lecture21.htm>

**Comparison Of Decision Tree Methodologies**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Decision Trees** | **Bagging** | **Boosting** | **Random Forest** |
| **No. of trees** | One | Many | Many | Many |
| **Weighting of trees** | - | Equal | Variable | Equal |
| **Variables in scope** | All | All | All | Randomly selected for each tree |
| **Cases used for fitting** | All training data | Random selection without replacement | All training data | Random selection with replacement |

**Further Reading**

* A visual introduction to machine learning & decision trees: <http://www.r2d3.us/visual-intro-to-machine-learning-part-1/>
* Data Mining Map - <http://www.saedsayad.com/decision_tree_reg.htm>
* AI Depot - <http://ai-depot.com/Tutorial/DecisionTrees-Partitioning.html>
* RPART documentation - <http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf>
* Intro to decision trees & random forests: <http://www.slideshare.net/DerekKane/data-science-v-decision-tree-random-forests>
* Wikipedia - <http://en.wikipedia.org/wiki/Decision_tree_learning>

**Annex – Creating decision trees and recursive partitioning in R**

#EXAMPLE OF APPLYING DECISION TREES AND RECUSRIVE PARTITIONING IN R

#This example uses is based on the passenger list from the Titanic. We are trying to

#predict who survived and who died based on their age (Age), the fare they paid (Fare)

#gender (Female), embarkation point (South), how many siblings/spouses in each party (SibSp)

#how many parents/children in each passengers party (Parch)

#The dataset is used is publically available. It is the basis of a machine learning

#example on Kaggle (https://www.kaggle.com/c/titanic-gettingStarted).

##########################################################################

#SET-UP

#Install the recursive partitioning and regression trees package

install.packages("rpart")

install.packages("rpart.plot")

install.packages("rattle")

require("rpart")

require("rpart.plot")

require("rattle")

#Set the working directory where you will be loading and saving data from

setwd("/Users/datascientist3/Desktop/R Packages/Kaggle Titanic")

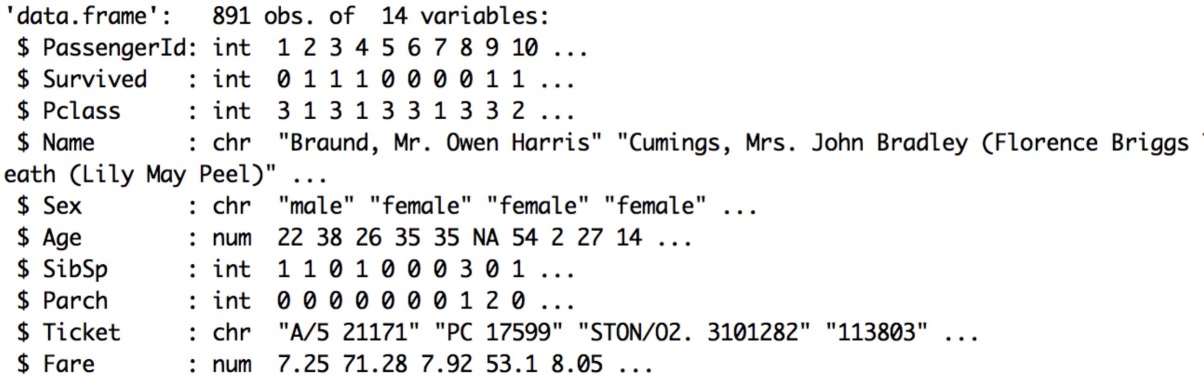
#Load Titanic dataset that we will train the data on

Titanic.Train <- read.csv("titanic\_train\_kaggle.csv", stringsAsFactors=FALSE)

#HAVE A LOOK AT THE DATA YOU ARE GOING TO USE

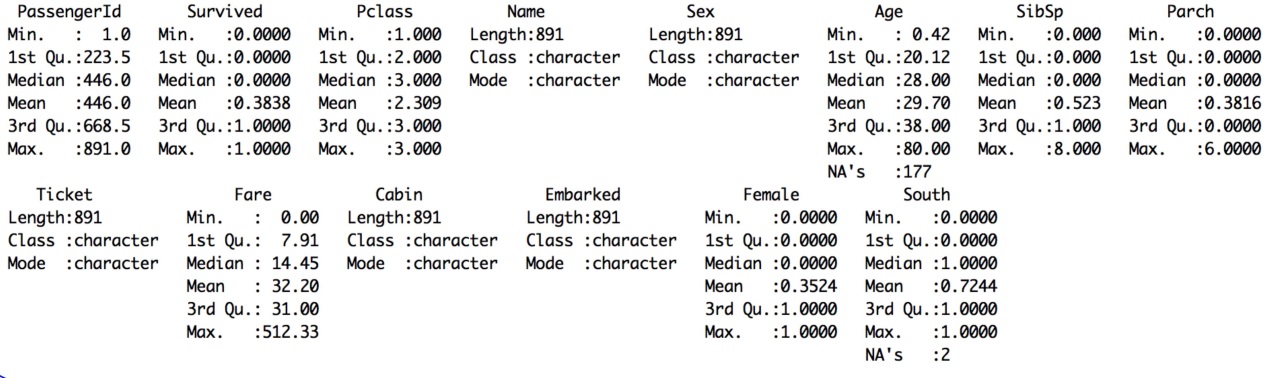
#Review structure of the data (shows field type and outputs form first few cases)

str(Titanic.Train)



#Look at a summary of the data (shows distribution of data for numeric fields)

summary(Titanic.Train)



#FIT THE MODEL

#Apply Decision Tree model to predict whether passengers survived based on the 6 variables

#Note that the control variables are option. These control the the size and structure of the

#decision tree. Type ?rpart.control help documentation for more information on how these work.

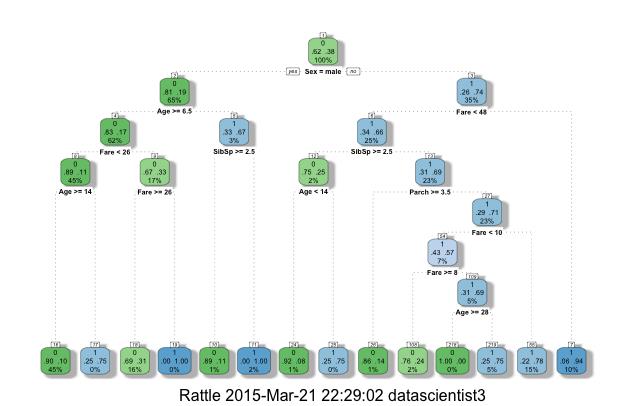
Titanic.Train.RP.model <- rpart(Survived ~ Sex + Age + SibSp + Parch + Fare + Embarked, #Equation

data=Titanic.Train, method="class", #Data source. Method = class is needed for binomial model

control=rpart.control(minsplit=8, minbucket=4, cp=0.005, maxdepth = 10)) #control variables

#Plot the decision tree

fancyRpartPlot(Titanic.Train.RP.model)

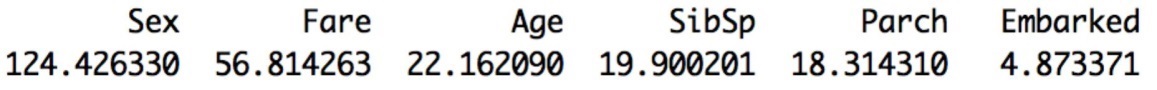


#An alternative style of plotting the tree

prp(Titanic.Train.RP.model)

#Show the importance of each of the variables. This is a reflection of the goodness of fit

#each variable provides. The high the number the more important the variable

Titanic.Train.RP.model$variable.importance 

#APPLY THE MODEL

#We can take the model we've fitted and apply it to our training data

#This will give us a percentage value for each record.

#This is our prediction of how likely they were to survive

#The predict function is of the form predict([model name],[dataset name], [output required])

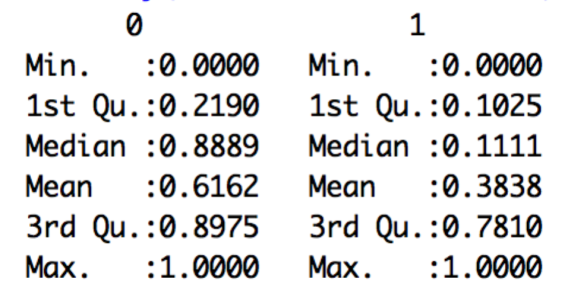
#In this case response gives us the % chance that they survive according to our model

Titanic.Train.RP.Preds <- predict(Titanic.Train.RP.model,type="prob")

#Look at a summary of our predictions. The % chance of survival varies from 5% to 100%

summary(Titanic.Train.RP.Preds)

**Did Not Survive Survived**



#Match the predictions back onto the main dataset to create a new file

#We need to take the second column of the predictions (the survival percentage)

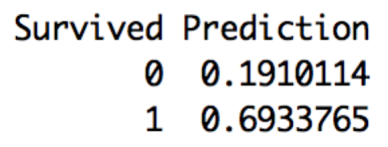
Titanic.Train.With.Predictions <- data.frame(Titanic.Train, Prediction = Titanic.Train.RP.Preds[,2])

#Show the average prediction chance of survival for those who did live or die (Survived = 1 or 0)

#Model gives an average prediction of 27% for those who died and 61% for those who lived

aggregate(Prediction ~ Survived, data=Titanic.Train.With.Predictions,

FUN=function(x) {sum(x)/length(x)})

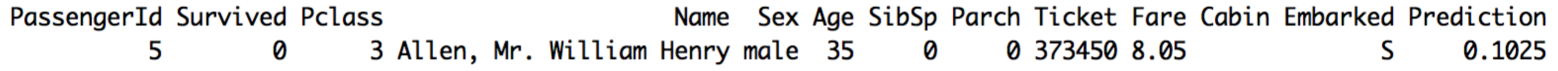


#Look at an individual result - the fifth case in the dataset

#Mr William Henry Allen did not survive (Survived = 0)

#Our model gave him a 10% chance of survival. Not very good odds! The model is doing OK here.

Titanic.Train.With.Predictions [5,] #Square brackets mean take 5th row (then comma) show all columns



#We can save our results to a csv file

write.csv(Titanic.Train.With.Predictions, file="Titanic.Train.With.Predictions.RP.csv")