Decision Trees Case Study

Applications of CART and Random Forest

Overview- LETTER RECOGNITION

One of the earliest applications of the predictive analytics methods was to automatically recognize letters, which post office machines use to sort mail.

In this problem, you will build a Decision Tree model that uses statistics of images of four letters in the Roman alphabet -- **A, B, P, and R** -- to predict which letter a particular image corresponds to.

Data

The file letters_ABPR.csv contains 3116 observations, each of which corresponds to a certain image of one of the four letters A, B, P and R. The images came from 20 different fonts, which were then randomly distorted to produce the final images; each such distorted image is represented as a collection of pixels, each of which is "on" or "off". For each such distorted image, we have available certain statistics of the image in terms of these pixels, as well as which of the four letters the image is. This data comes from the UCI Machine Learning Repository**.**

This dataset contains the following 17 variables:

- *letter* = the letter that the image corresponds to (A, B, P or R)
- *xbox* = the horizontal position of where the smallest box covering the letter shape begins.
- *ybox* = the vertical position of where the smallest box covering the letter shape begins.
- *width* = the width of this smallest box.
- *height* = the height of this smallest box.
- *onpix* = the total number of "on" pixels in the character image
- *xbar* = the mean horizontal position of all of the "on" pixels
- *ybar* = the mean vertical position of all of the "on" pixels
- *x2bar* = the mean squared horizontal position of all of the "on" pixels in the image
- *y2bar* = the mean squared vertical position of all of the "on" pixels in the image
- *xybar* = the mean of the product of the horizontal and vertical position of all of the "on" pixels in the image

- *x2ybar* = the mean of the product of the squared horizontal position and the vertical position of all of the "on" pixels
- *xy2bar* = the mean of the product of the horizontal position and the squared vertical position of all of the "on" pixels
- *xedge* = the mean number of edges (the number of times an "off" pixel is followed by an "on" pixel, or the image boundary is hit) as the image is scanned from left to right, along the whole vertical length of the image
- *xedgeycor* = the mean of the product of the number of horizontal edges at each vertical position and the vertical position
- *yedge* = the mean number of edges as the images is scanned from top to bottom, along the whole horizontal length of the image
- *yedgexcor* = the mean of the product of the number of vertical edges at each horizontal position and the horizontal position\

Set the seed to 1000 before making the split

Case Study Problems

1. **Predicting the letters A, B, P, R - The Baseline Model**

    1.1. Load the file letters_ABPR.csv into R, and call it letters. The variable in our data frame which we will be trying to predict is "letter". Start by converting letter in the original data set (letters) to a factor.
    1.2. Now, generate new training and testing sets of the letters data frame using letters$letter as the first input to the sample.split function.
    1.3. What is the baseline accuracy on the testing set?

2. **Predicting the letters A, B, P, R - The CART Model**
    2.1. Now build a classification tree to predict "letter", using the training set to build your model.
    2.2. What is the accuracy of the model in the train data set?
    2.3. What is the accuracy of the model in the test data set?

3. **Predicting the letters A, B, P, R - The Random Forest Model**

    3.1. Now build a random forest model on the training data, using the same independent variables as in the previous problem. Please set the seed to 1000 before building the model
    3.2. What is the test set accuracy of your random forest model?

Deliverables Required

- A word document with answers for all the questions
- R code