(https://www.edx.org)

MITx: 15.071x The Analytics Edge

zhushun0008 (/dashboard) ▾

Courseware (/courses/MITx/15.071x/1T2014/courseware) Course Info (/courses/MITx/15.071x/1T2014/info)

Discussion (/courses/MITx/15.071x/1T2014/discussion/forum) Progress (/courses/MITx/15.071x/1T2014/progress)

Syllabus (/courses/MITx/15.071x/1T2014/4264e68418f34d839cf0b33a5da644b2/)

Schedule (/courses/MITx/15.071x/1T2014/2891f8bf120945b9aa12e6601739c3e6/)

LETTER RECOGNITION

One of the earliest applications of the predictive analytics methods we have studied so far in this class was to automatically recognize letters. One application is for post office machines to sort mail. In this problem, we will build a 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.

This is slightly different from the problems we have considered so far. We have previously focused on binary classification problems (e.g., predicting whether an individual voted or not, earns more than $50,000 annually or not, is at risk for a certain disease or not, etc.). In this problem, we have more than two classifications that are possible for each observation, like in the D2Hawkeye lecture. Such problems are called **multiclass classification problems**.

Help

The file letters_ABPR.csv (/c4x/MITx/15.071x/asset/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 (http://archive.ics.uci.edu/ml/datasets/Letter+Recognition).

The available variables include:

- *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

PROBLEM 1.1 - PREDICTING B OR NOT B (1 point possible)

Let's warm up by attempting to predict just whether a letter is B or not. To begin, load the file letters_ABPR.csv into R, and call it letters. Then, create a new variable isB in the dataframe, which takes the value "yes" if the observation corresponds to the letter B, and "no" if it does not. You can do this by typing the following command into your R console:

```
letters$isB = as.factor(letters$letter == "B")
```

Now split the data set into a training and testing set, putting 50% of the data in the training set. Set the seed to 1000 before making the split. The first argument to sample.split should be the dependent variable "letters$isB". Remember that TRUE values from sample.split should go in the training set.

Before building models, let's consider a baseline method that always predicts the most frequent outcome, which is "not B". What is the accuracy of this baseline method on the test set?

Show Answer *You have used 0 of 3 submissions*

PROBLEM 1.2 - PREDICTING B OR NOT B (1 point possible)

Now build a classification tree to predict whether a letter is a B or not, using the training set to build your model. Remember to remove the variable "letter" out of the model, as this is related to what we are trying to predict! To just remove one variable, you can either write out the other variables, or remember what we did in the Billboards problem in Week 3, and use the following notation:

```
CARTb = rpart(isB ~ . - letter, data=train, method="class")
```

We are just using the default parameters in our CART model, so we don't need to add the minbucket or cp arguments at all. We also added the argument method="class" since this is a classification problem.

What is the accuracy of the CART model on the test set? (Use type="class" when making predictions on the test set.)

Show Answer *You have used 0 of 5 submissions*

PROBLEM 1.3 - PREDICTING B OR NOT B (1 point possible)

Now, build a random forest model to predict whether the letter is a B or not. Use the default settings for ntree and nodesize (don't include these arguments at all). Right before building the model, set the seed to 1000. (NOTE: You might get a slightly different answer on this problem, even if you set the random seed. This has to do with your operating system and the implementation of the random forest algorithm.)

What is the accuracy of the model on the test set?

In lecture, we noted that random forests tends to improve on CART in terms of predictive accuracy. Sometimes, this improvement can be quite significant, as it is here.

Show Answer *You have used 0 of 5 submissions*

PROBLEM 2.1 - PREDICTING THE LETTERS A, B, P, R (1 point possible)

Let us now move on to the problem that we were originally interested in, which is to predict whether or not a letter is one of the four letters A, B, P or R. However, we have only seen so far how CART and random forests can be applied to binary classification problems (e.g. B or not B). What can we do?

One approach is to build several layers of predictive models corresponding to each of the letters. The idea would be that we would have one tree to predict whether a letter is a B or not. If it is predicted to be a B, we would output B as our final prediction. Otherwise, we would run the observation through another tree to predict (say) whether the letter is an A or not. We would then repeat this process for the other layers.

Although such a proposal may sound reasonable, it is rather complicated. Fortunately, it turns out that CART and random forests generalize to multiclass classification problems such as, for example, predicting one of our four letters of interest. As we will see shortly, building the corresponding models in R turns out to be no harder than building the models for binary classification problems.

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 by running the following command in R:

```
letters$letter = as.factor( letters$letter )
```

Now, generate new training and testing sets of the letters data frame using letters$letter as the first input to the sample.split function. Before splitting, set your seed to 2000. Again put 50% of the data in the training set. (Why do we need to split the data again? Remember that sample split balances the outcome variable in the training and testing sets. With a new outcome variable, we want to re-generate our split.)

In a multiclass classification problem, a baseline model is to predict the most frequent class of all of the options.

What is the baseline accuracy on the testing set?

Show Answer *You have used 0 of 3 submissions*

PROBLEM 2.2 - PREDICTING THE LETTERS A, B, P, R (1 point possible)

Now build a classification tree to predict the letter, using the training set to build your model. (Remember to remove the variable "isB" out of the model, as this is related to what we are trying to predict!) Just use the default parameters in your CART model. Add the argument method="class" since this is a classification problem. Even though we have multiple classes here, nothing changes in how we build the model from the binary case.

What is the test set accuracy of your CART model?

(HINT: When you are computing the test set accuracy using the confusion matrix, you want to add everything on the main diagonal and divide by the total number of observations in the test set, which can be computed with nrow(test), where test is the name of your test set).

Show Answer *You have used 0 of 5 submissions*

PROBLEM 2.3 - PREDICTING THE LETTERS A, B, P, R (1 point possible)

Now estimate a random forest model on the training data -- again, don't forget to remove the isB variable. Set the seed to 1000 right before building your model. (Remember that you might get a slightly different result even if you set the random seed.)

What is the test set accuracy of your random forest model?

You should find this value rather striking, for several reasons. The first is that it is significantly higher than the value for CART, highlighting the gain in accuracy that is possible from using random forest models. The second is that while the accuracy of CART decreased significantly as we transitioned from the problem of predicting B/not B (a relatively simple problem) to the problem of predicting the four letters (certainly a harder problem), the accuracy of the random forest model decreased by a tiny amount.

Show Answer *You have used 0 of 5 submissions*

Please remember not to ask for or post complete answers to homework questions in this discussion forum.

Show Discussion

New Post