IS6481-Fall2018-Assignment-4: Binary Classifier Bake-Off

Lynd Bacon, lynd.bacon@hsc.utah.edu

READ EVERYTHING THAT FOLLOWS CAREFULLY.

Estimating and Comparing Binary Classifiers

Classifiers are used to predict limited, and usually unordered, dependent variables, variables whose values can assume a finite number of values, like U.S. States, gender, brands of automobiles. Your assigned readings describe several different kinds of classifiers. The following textbook covers several of them as well. (It's not required reading, but it is a handy reference.)

Instroduction To Statistical Learning, 7th Ed.

Binary classifiers are used to predict membership in one of two categories. You've already used two binary classifiers, the binary logistic regression model, and the binary probit regression model. In this assignment you're going to use logistic regression and three additional binary classifier models to predict Titanic passenger survival. You're going to compare your models using a classifier performance description called the *Receiver Operating Characteristic* (ROC) curve, and a metric that uses the ROC curve to summarize performance that's called the *Area Under the Curve*, "AUC."

When doing this assignment you will need to install some R packages. You will need to rely on R's help documentation, and on available R "vignettes."

You may find it useful to search online for examples or for solutions to problems that you encounter. Business analytics and data science practitioners look online all the time. There's a good chance that, if you run into a problem or you get stuck on something, someone else already has, and there's already a solution available somewhere *Out There*. Be sure to use the assignment Huddle, too, of course.

Get the Titanic Data

Go to the web page Vanderbilt Dept. of Biostatistics Datasets

The dataset you want is in the section **Data for Titantic Passengers**. The name of the file you want is `titanic.sav`. The page linked by *titanic.html* provides some information about the dataset.

Download `titanic.sav` to a directory you can access from RStudio.

Load this file into RStudio. Note that the command to use for this is the `load()` command, even though the file doesn't have an .RData extension. It's an R binary file.

The Data

If you examine the data you'll note there are 1313 observations (rows), and different kinds of variables including some demographics, information about passenger class, passenger embarcation port, and home destination. Each row is a Titanic passenger record.

Look online to make sure you know what these variables measure. What, for example, is the variable `boat`?

What you are going to predict is the binary variable `survived`. It indicates whether a passenger survived the sinking or not, 1='yes', and 0='no'.

There are some variables that need to be treated as *dicrete*, or categorical, variables if they are to be used to predict `survived`. As indicated above, a limited or discrete variable is one on which the data can only take on a finite, and usually countable, number of values, and not all possible values on the $\mathbb{R}$ (real number) line. Examples in the `titanic` data include sex, name, and passenger class. *Be sure to that all such variables are R data type "factor" before using them in the modeling that follows.* (Not treating them as such will result in erroneous results.)

Estimation and Test Random Subsamples

Split the data randomly into 70%/30% subsamples, the 70% for model estimation, and the 30% for assessing model performance using data not employed for estimation.

The Models

Using all the variables in your estimation subsample, "train" (fit) each of the following models to predict the `survived` variable:

- Binary Logistic Regression
- Naive Bayes Classifier (In the `e1017` R package)
- CART, a "recursive partitioning" tree model (In the `rpart` package)

Be sure to read the R documentation for each kind of model. You might find it useful to run the examples provided in the help.

Training (Fitting) The Models

Train (fit) each model, and then assess it's predictive accuracy using your "test" subsample of data.

Note that the tree models that `rpart` produces will often overfit their training data. How will you be able tell how much they do this? To get better results when applying them to test data, "pruning" a tree may help. See the functions `prune()` and `prune.rpart()` in the `rpart` package.

A good place to start on this assessment is to (1) cross-classify predicted vs. actual survivals using the R `xtabs()` function. You should use `xtabs()` to create two-way, or "fourfold," tables of predicted `survived` vs. actual `survived`.

Then, (2), calculate percent correct classifications. Do (1) and (2) for both the estimation subsample, and the test subsample. ROC analysis will provide more information about the predictive performance of your models.

ROC Analysis and AUC

For some background on ROC and AUC, see the paper **fawcett-ROC-AUC-2005.pdf** that's in this assignment's subdirectory in our GitHub repo.

See also the Wikipedia page:

Receiver Operating Characteristic

There are at least a couple of R packages that do ROC analysis. A good R package for doing this assignment is the package `ROCR`. This package will produce a number of different performance measures.

The curve on a ROC plot relates a model's true positive ("Hit") rate versus the model's false positive ("False Alarm") Rate. Both rates go from zero to one. You can find definitions of these measures in the `ROCR` documentation, as well as information about plotting an ROC curve.

For each of your models, provide its estimated AUC for both the estimation data and for the test data. You should should report six (6) AUC's, two for each type of model.

For the model that has the best AUC for your test data, plot an ROC curve.

Be sure you review the help documentation for the `ROCR` function and for the plot methods of the `ROCR` package. A little reading up front can save you a lot of time and mistakes.

How To Do It

Get started on this assignment ASAP. It may be impossible to complete it successfully if you start doing it on the weekend it is due.

Be sure that you address every issue indicated above, and to include every required analytic result.

Provide a description of your modeling and the results you obtained in a pdf file that you produced from an R Notebook. Explain what you did, and what you got, in enough detail so that others who are familiar with R can understand what you got, and can replicate your findings, given that they have the same data you used. Do *not* turn in just code, or only code plus some outputs. Explain your code, and interpret your results.

Tell the story of what you did and what you got. Communicating results in the interest of informing and facilitating collaboration, is an important objective when doing business analytics.

Which of your models most accurately predicted passenger survival? Do your results have anything to say about which variables were the most powerful predictors?

Be sure that you explain (not just name) each kind of model. Describe how it works in a simple, conceptual way. No need for equations, just a conceptual statement about each model will suffice. Imagine that you are explaining your models to someone who isn't familiar with them.

Do *not* exceed twelve (12) pages in your pdf file. Pages beyond 12 will not be graded. Note that you don't necessarily need to turn 12 pages in. Don't include "fluff" (irrelevant or unnecessary content) in your submission so that you have 12 pages. Use 12 pages only if you need to. (Spare your Reader.) The page limit doesn't indicate how much you need to say, it indicates how much at most you can say.

Try to avoid submitting your assignment late. In case you haven't already done so, review the policy on late work mentioned in the course syllabus. Also, try to get it as "right as possible" the first time. Opportunities for "do-overs" will be rare, and chances are that the late work policy will apply to them.