G4058 Assignment 5

Due: Friday, April 22, 2016 by 6PM

Instructions

Create a RMarkdown file that contains the answers to the following questions. Verify that you can knit the RMarkdown file to an HTML file. Upload both the RMarkdown file and the HTML file to the Assignments section (not the Dropbox section) of CourseWorks when you are finished.

Work on this problem set by yourself, but you can ask questions on Piazza. Just remember to click on the button that says your question will be visible to instructors only.

Smooth Nonlinear Models for a Continuous Outcome

Use the College dataset, which can be accessed by executing

```
stopifnot(require(ISLR))

## Loading required package:  ISLR

str(College)

## 'data.frame': 777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

The variables are described under `help(College)`.

(a) Split the data into a training set and a testing set in some appropriate fashion. Use the `lm()` and `step()` functions to obtain a decent linear model for `Outstate` in the training data that also includes whatever interactions you feel are appropriate.

(b) Use the `gam` function in the gam package to fit a Generalized Additive Model where `Outstate` is the outcome using the predictors from your best model for the training data in part (a). Explain what calling `plot()` on the resulting gam object tells you.

(c) Which predictors, if any, exhibit a non-linear relationship with `Outstate`, conditional on the other predictors?

(d) Use the results of part (b) to predict `Outstate` in the testing data. Is the average squared error in the testing data greater, less than, or about the same than in the training data? Why is that the case?

Tree-Based Models for a Binary Outcome

Download the dataset.RData from CourseWorks to your working directory and load it into R via

```
load("dataset.RData")
str(dataset)
```

These is the same dataset that was used in the last homework where we modeled whether a loan was approproved. The variables are (still):

- `Amount.Requested`: The proposed amount for the loan
- `Debt.To.Income.Ratio`: The ratio of the applicant's debt (excluding mortgages and the proposed loan) payments each month to the applicant's stated monthly income
- `Zip.Code`: The 3-digit zip code of the applicant
- `State`: The state where the applicant lives
- `Employment.Length`: The number of years that the applicant has worked at the same job. `10` indicates at least ten years, `0` indicates less than one year, and `-1` indicates unemployed.
- `y`: A binary variable indicating whether the loan was approved

(a) Split the data into a training set and a testing set in some appropriate fashion. Fit a logit model to the outcome in the training data, using whatever interaction and polynomial terms you feel are appropriate.

(b) Use a random forest and then a single-tree approach to fit the outcome in the training data.

(c) Rank the three approaches in parts (a) and (b) in terms of which is most likely to yield a correct prediction in the testing data, using `predict()` with either `type = "class"` or equivalently classifying according to whether the probability is less than or greater than $\frac{1}{2}$.