Homework 4: Predicting Crime

STAT 665

due Thurs. Feb. 23th 5PM

The Data

This dataset consists of socio-economic data from the 1990 US Census, as well as law enforcement and crime data on a per-community basis. You may find the codebook below helpful:

http://archive.ics.uci.edu/ml/machine-learning-databases/communities/communities.names

Using this dataset, we are interested in predicting the per capita violent crimes variable `ViolentCrimesPerPop`. Violent crimes include murder, rape, robbery, and assault.

Part 1 (30%)

We'll start with some basic explorations of the dataset. Answer the following questions:

(1) How many distinct states are represented in this dataset? How many distinct counties are represented?

(2) Missing values are a problem in this dataset (coded as `NA`). Produce a frequency table with the number of missing values per column. That is to say, if all columns either have 0, 50, or 200 missing values, you should produce a frequency table that tells us how many columns fall into each of these categories. We will now use this reduced dataset for subsequent parts of the assignment.

(3) Discard all columns with more than 50% of the values missing (we probably can't use these variables for something like linear regression). Reproduce the frequency table from (2) using the reduced dataset.

(4) Plot a histogram of the response variable `ViolentCrimesPerPop`. In a sentence, what do you learn about the distribution of this variable from the plot? (You should also skim the Codebook linked above to get a sense of what the units/scale are for the different variables.)

Part 2 (30%)

In this part, you will be asked to run stepwise selection. Please suppress the printout in your stepwise selection process. (In R, if you are using `step()`, you can set the `trace=` option to FALSE.) **Because lengthy extraneous output can make graders grumpy, you will be penalized 10% of the points if you fail to suppress the step-by-step printout.**

There are a lot of identifying variables in this dataset, which we probably don't want to use. For example, this includes the `fold` variable, which should be used for 10-fold cross-validation, but not as a predictor in regression. You should bear this in mind as you construct the model that serves as the upper scope of the stepwise regression procedure (i.e. the model used in the `scope=list(upper=...)` argument for `step()`). If you simply use all the predictors in this maximal full model, you will likely have very long run-times in stepwise selection and run the risk of overfitting the data.

(1) Run forward stepwise selection on your dataset. You will likely get an error/warning message, which you can and should fix by discarding columns with any missing values from consideration. Do not print the output of the model. Explain why you would have a warning message if you use any predictors with missing values in the stepwise selection process.

(2) Also run backward and bi-directional stepwise selection on your dataset. Again, do not print the output of the models. For each of the 3 models obtained via forward, backward, and bi-directional stepwise selection, report the model sizes (i.e. number of predictors) and the R^2 values achieved.

Part 3 (40%)

Select the best model (among the 3) from Part 2 on the basis of R^2 and model complexity (just by reasoning, no calculations/statistical tests needed). We will now focus on this best model in Part 3.

Note: you can extract the formula from your best model in R (say `model`) via:

```
f <- formula(model)
```

This is helpful because now you can fit this exact model on a different dataset (without typing out all of the predictors) using:

```
lm(f, data=...)
```

(1) Which model did you end up choosing? Justify your answer, show the regression output, and in a sentence or two, summarize what you learn from two or three of the significant coefficients.

(2) Run 10-fold cross-validation to estimate the test MSE. Use the `fold` column in the dataset for determining the folds. (Do not create your own random folds.) Report your estimated test MSE.

(3) Use the bootstrap to estimate a 90% confidence interval for multiple R^2.

What to Submit

Just your compiled PDF file, from a Python notebook or from R Markdown. No .csv's to submit this time!