Homework 6

STAT 430, Spring 2017

Due: Friday, March 17 by 11:59 PM

Please see the homework instructions document for detailed instructions and some grading notes. Failure to follow instructions will result in point reductions.

Exercise 1

[**10 points**] For this exercise use the data found in `hw06-data.csv`. We will attempt to classify the y variable.

Do the following:

- Set a seed value equal to your UIN
- Test-train split the data using approximately 50% of the data for training.
- Fit three models:
  - Additive logistic regression
  - Logistic regression with predictors chosen using a variable selection technique of your choice
  - k-nearest neighbors using a well tuned value of k
- Report 5 fold cross-validated, train, and test accuracy for each of the three models
  - You should arrange this in a table
  - **You do not need to cross-validate the selection method, but rather only cross-validate the resulting model**

Also answer the following:

- What is the model you chose via selection?
- Plot the cross-validated k-nearest neighbor results. Argue that this plot verifies that you have considered enough values of k.
- What value of k did you select?
- Based on these results, which model do you prefer?

```
uin = 123456789
set.seed(uin)
```

Exercise 2

[**20 points**] For this question we will use the data in `leukemia.csv` which originates from Golub et al. 1999.

The response variable `class` is a categorical variable. There are two possible responses: `ALL` (acute myeloid leukemia) and `AML` (acute lymphoblastic leukemia), both types of leukemia. We will use the many feature variables, which are expression levels of genes, to predict these classes.

Note that, this dataset is rather large and you may have difficultly loading it using the "Import Dataset" feature in RStudio. Instead place the file in the same folder as your `.Rmd` file and run the following command. (Which you should be doing anyway.) Again, since this dataset is large, use 5-fold cross-validation when needed.

```
library(readr)
leukemia = read_csv("leukemia.csv", progress = FALSE)
```

For use with the `glmnet` package, it will be useful to create a factor response variable `y` and a feature matrix `X` as seen below. We won't test-train split the data since there are so few observations.

```r
y = as.factor(leukemia$class)
X = as.matrix(leukemia[, -1])
```

Do the following:

- Fit an logistic regression with a lasso penalty. (Don't use cross-validation. Also let `glmnet` choose the λ values.) Create a plot that shows the features entering the model.
- Use cross-validation to tune an logistic regression with a lasso penalty. Again, let `glmnet` choose the λ values. Store both the λ that minimizes the deviance, as well as the λ that has a deviance within one standard error. Create a plot of the deviances for each value of λ considered. Use these two λ values to create a grid for use with `train()` in `caret`. Use `train()` to get cross-validated classification accuracy for these two values of λ. Store these values.
- Fit an logistic regression with ridge penalty. (Don't use cross-validation. Also let `glmnet` choose the λ values.) Create a plot that shows the features entering the model.
- Use cross-validation to tune an logistic regression with a ridge penalty. Again, let `glmnet` choose the λ values. Store both the λ that minimizes the deviance, as well as the λ that has a deviance within one standard error. Create a plot of the deviances for each value of λ considered. Use these two λ values to create a grid for use with `train()` in `caret`. Use `train()` to get cross-validated classification accuracy for these two values of λ. Store these values.
- Use cross-validation to tune k-nearest neighbors using `train()` in `caret`. Do not specify a grid of k values to try, let `caret` do so automatically. (It will use 5, 7, 9.) Store the cross-validated accuracy for each.

Also answer the following:

- How many observations are in the dataset? How many predictors are in the dataset?
- Based on the deviance plot, do you feel that `glmnet` considered enough λ values for lasso? For ridge?
- How does k-nearest neighbor compare to the penalized methods? Can you explain any difference?
- Summarize these **seven** models in a table. (Two lasso, two ridge, three knn.) For each report the cross-validated accuracy and the standard deviation of the accuracy.
- Based on your results, which model would you choose?