Homework 08

STAT 430, Fall 2017

Due: Friday, November 10, 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 (Classifying Leukemia)

[**10 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.

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

Do the following:

- Set a seed equal to your UIN.
- Fit the full path of a logistic regression with both a lasso penalty and a ridge penalty. (Don't use cross-validation. Also let `glmnet` choose the λ values.) Create side-by-side plots that shows the features entering (or leaving) the models.
- 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.
- 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. Scale the predictors.
- 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.

Exercise 2 (The Cost of College)

[**10 points**] For this exercise, we will use the `College` data from the `ISLR` package. Familiarize yourself with this dataset before performing analyses. We will attempt to predict the `Outstate` variable.

Test-train split the data using this code.

```
set.seed(42)
library(caret)
library(ISLR)
index = createDataPartition(College$Outstate, p = 0.75, list = FALSE)
college_trn = College[index, ]
college_tst = College[-index, ]
```

Train a total of **six** models using five-fold cross validation.

- An additive linear model.
- An elastic net model using additive predictors. Use a `tuneLength` of 10.
- An elastic net model that also considers all two-way interactions. Use a `tuneLength` of 10.
- A well-tuned KNN model.
- A well-tuned KNN model that also considers all two-way interactions. (Should this work?)
- A default-tuned random forest.

Before beginning, set a seed equal to your UIN.

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

- Create a table which reports CV and Test RMSE for each.

Exercise 3 (Concept Checks)

[**1 point each**] Answer the following questions based on your results from the three exercises.

Leukemia

(a) How many observations are in the dataset? How many predictors are in the dataset?

(b) Based on the deviance plots, do you feel that `glmnet` considered enough λ values for lasso?

(c) Based on the deviance plots, do you feel that `glmnet` considered enough λ values for ridge?

(d) How does k-nearest neighbor compare to the penalized methods? Can you explain any difference?

(e) Based on your results, which model would you choose? Explain.

College

(f) Based on the table, which model do you prefer? Justify your answer.

(g) For both of the elastic net models, report the best tuning parameters from `caret`. For each, is this ridge, lasso, or somewhere in between? If in between, closer to which?

(h) Did you scale the predictors when you used KNN? Should you have scaled the predictors when you used KNN?

(i) Of the two KNN models which works better? Can you explain why?

(j) What year is this dataset from? What was out-of-state tuition at UIUC at that time?