Homework #4
Design of Experiments and Machine Learning

ENGM 182
Data Analytics
Due Friday May 10, 2019

1 Credit Analysis using machine learning

# Load dataset credit.csv (on Canvas). Use the data to predict whether a borrower is likely to default (default=2) or not (default=1). It might be useful to recode 1,2 in default to "no" and "yes" respectively.

1. Split your data into a training set and a test set.
2. Use the C50 package in R to develop a model to predict default. For information, see

   `https://cran.r-project.org/web/packages/C50/vignettes/C5.0.html`

3. Now use the test data to check how well your model works. How did your model perform (show the confusion matrix)? Note, to create a nice confusion matrix, you can use the "gmodels" package as follows where "credit_pred" contains predictions made by your model using the test data.

   ```
   library(gmodels)

   CrossTable(credit_test$default, credit_pred,
             prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
             dnn = c('actual default', 'predicted default'))
   ```

   (watch out for cut and paste; the single quotes may not copy correctly).

4. What is the default cost for misclassification? Now try adding costs to different types of errors. Suppose that a loan default costs the bank four times as much as a missed opportunity (of course, in reality this is a ratio that should be determined through analysis). You can use the "costs = mycostmatrix" option when calling the algorithm to specify costs. How does your new model with costs perform relative to the first model in terms of correct classification? Does this make sense in context of the cost matrix you specified? If so, why? If not, how would you improve it?

2 Extension of class exercises

This is a continuation of the exercises from last Tuesday's class. See the code for both the model comparison script exercise and the hyperparameter optimization exercise on Canvas.

1. Implement a new learning algorithm in Exercise 1 - Intro to Machine Learning (in addition to the others which you have already tested including LDA, random forest, etc.) on the Iris dataset and test its performance using appropriate metrics. For this exercise, accuracy and a confusion matrix are appropriate! Did it perform better or worse than the other models tested in class?

   You can use this link to look at all of the models available to you through the caret package in R:

   `https://topepo.github.io/caret/available-models.html`

   If you're not sure where to start, look for "AdaBoost Classification Trees - adaboost" and try implementing that model first.

2. Now apply your chosen model to the Sonar dataset by modifying code from Exercise 2 - Hyperparameter Optimization. Look up the parameters used by your algorithm and vary one of them using either a manual search, grid search, or random search - you can use the code written in Exercise 2 to do this. What parameter did you optimize? What was the best performance you achieved with this new model? What was the value of the parameter which gave you the best results?

   Just a note Remember that a hyperparameter is simply a parameter which we set before the model begins learning. This parameter affects the way that the model learns and can influence its ability to converge. We can change the value of a single hyperparameter, then retrain a model a number of times examining how this results in different model performance. Feel free to modify the code from the Hyperparameter Optimization exercise from class to do this!

The code for the two class exercises as available as

Intro to Machine Learning.R

Hyperparameter Optimization in R.R