**IST687 – SVM: Food Inspection**

The textbook’s (“What’s Your Vector Victor”) chapter introduces support vector machines – a powerful method of supervised machine learning. Support vector machines have proven value in classification tasks. At the simplest level a classification task is one where the system/algorithm must predict whether an outcome is this or that, true or false, sick or well, or any other dichotomous outcomes. In this final exercise for the semester, you will use a public data set of restaurant inspections to see if you can predict whether or not a restaurant received an inspection violation.

As with the other recent exercises, this one has no sample R code. Likewise, the instructions are quite skeletal. As we reach the end of the semester, your diagnostic skills should be very sharp. As you run into barriers and problems, you should have all of the necessary skills to solve the problems on your own.

1. The data set is available at the NY State Health Data website:  
   <https://health.data.ny.gov/Health/Food-Service-Establishment-Last-Inspection/cnih-y5dw>   
     
   When you view this URL you will find that the data set pops right up in your browser. Obviously to work with it in R you will have to download it to your own computer.
2. After you have loaded the data into R take a close look at the VIOLATIONS variable. Because we want to test a simple dichotomous classification model, you will have to create a new variable that uses the data in VIOLATIONS to encode whether or not the restaurant had one or more violations. For this particular exercise, we don’t care how many violations there were, or what type, but simply whether there were some violations or not.
3. Next, apply all of the techniques you learned from chapter 18. This means that you will have to load the kernlab package, run ksvm(), set the parameters correctly, inspect the results, visualize the results, and make sense out of what you find. You should start with a very simple model that only uses one or two predictor variables. Which predictors in this data set are likely to be most useful? You will have to give some thought to the process of restaurant inspection to consider what factors (among those available in this dataset) are likely to predict a violation.
4. After you have run and interpreted a model with one or two predictors, make sure to run a second model with additional predictors. How good can you make your prediction (i.e., how low can you drive the error rate)? For your final model, make sure to create, display, and interpret a confusion matrix. You should use the confusion matrix to calculate and report at least one error or accuracy rate.

Challenge Layer: There is no additional challenge layer for this exercise, because most people will find Step 4 above sufficiently hard. Note that depending upon the power of the machine you are using to run these models, it might be quite time consuming to develop the support vectors for each model. How can you get around this problem (short of getting a better computer)? The other challenging thing to pay attention to is this: Chapter 18 walks you through the steps of creating and interpreting an SVM model, but some of the code in the chapter will not work exactly as you expect on these new data. You will have to be alert to changes in data structures so that you can modify your R code as needed to get around any problems you encounter.

**Learning Goals for this activity:**

1. Consider how a powerful supervised machine learning technique can be applied to a new data set.
2. Provide practice in conditioning data to prepare for analysis.
3. Develop skill in the setup, execution, and interpretation of support vector machines.
4. Increase familiarity with bringing external data sets into R.
5. Increase familiarity with sources of advice and ideas on R source code.

**Essential Guide for All IST687 Activities (appears at the end of all activity guides)**

1. All IST687 activities work on what some people call a “constructivist learning” model. By developing a product on your own, testing it to find flaws, improving it, and comparing your solution to the solutions of other people, you can obtain a deeper understanding of a problem, the tools that might solve that problem, and a range of solutions that those tools may facilitate. The constructivist model only works to the extent that the student/learner has the drive to explore a problem, be frustrated, fail, try again, possibly fail again, and finally push through to a satisfactory level of understanding.
2. Each IST687 activity builds on skills and knowledge developed in the previous activities, so your success across the span of the course depends at each stage on your investment in earlier stages. Take the time to experiment, play, try new things, practice, improve, and learn as much as possible. These investments will pay off later.
3. Using the expertise of others, the Internet, and other sources of information is not only acceptable - it is expected. You must ***always, always, always*** give credit to your sources. For example, if you find a chunk of code from r-bloggers.com that helps you with developing a solution, by all means borrow that chunk of code, but make sure to use a comment in your code to document the source of the borrowed code chunk. The discussion boards in the learning management system have been setup to encourage appropriate sharing of knowledge and wisdom among peers. Feel free to ask a question or pose a solution on these boards.
4. Building on the previous point, when submitting code as your solution to the activity, the comments matter at least as much, if not more than the code itself. A good rule of thumb is that every line of code should have a comment, and every meaningful block of code should be preceded by a comment block that is just about as long as the code itself. As noted above, you can use comments to give proper credit to your sources and you can use comments to identify your submission as your own.
5. Sometimes the building process reveals unexpected results that are themselves very informative in learning. When you completed the exercise above, what did you find that was unexpected? What did you do about trying to understand what had happened? Did you do further exploration? What did that further exploration reveal?
6. Frustration is actually a powerful source of learning, if you can push through to the “other side” (i.e., you can ultimately work around the source of the frustration). Combining the skills from previous lessons with new skills and applying them to a difficult and novel problem will almost inevitably lead to glitches in the process of constructing your artifact (in this case the R code for the exercise). Embrace that frustration and see if you can get through it to deeper learning.
7. One last thought to close out the semester: What if you were not able to finish this exercise (or any other one) with everything as complete and correct as you might have hoped? Did you still learn something useful? Did you take the opportunity to build out your set of resources, helpers, experts, etc. that can help you solve complex problems the next time around? You can harness today’s failures in service of tomorrow’s successes.