FUNDAMENTALS OF DATA SCIENCE – DSC 441

HOMEWORK 2

Turn your assignment in as a PDF file to the D2L submission folder by the due date associated with that folder (unless otherwise specified explicitly by the Instructor). Late assignments will incur a penalty. Please ask if you need an extension and come to office hours if you need assistance.

Academic integrity is key to maintaining the impact of your degree. You may discuss the problems and methods with others, but everything you submit must be your own work. For violating, you may incur significant sanctions including failing the course or penalties from the University. If in doubt, ask first.

To get credit, you must clearly label your responses with the questions they answer. Include results and visualizations that you generate, and clearly written answers to any questions included in the problem. Your code file must be submitted alongside your document in D2L.

This homework assignment covers material from Modules 3 and 4. You will get additional practice with data preprocessing and then we begin our first uses of data mining algorithms. There are multiple problems where you will build and evaluate support vector machines to give you the chance to develop comfort not only with this particular type of algorithm, but with the general classification process.

Problem 1 (25 points):

For this problem, you will load and perform some cleaning steps on a dataset in the provided *BankData.csv,* which is data about loan approvals from a bank in Japan (it has been modified from the original for our purposes in class, so use the provided version). Specifically, you will use visualization to examine the variables and normalization, binning and smoothing to change them in particular ways.

a. Visualize the distributions of the variables in this data. You can choose bar graphs, histograms and density plots. Make appropriate choices given each type of variables and be careful when selecting parameters like the number of bins for the histograms. Note there are some numerical variables and some categorical ones. The ones labeled as a ‘bool’ are Boolean variables, meaning they are only true or false and are thus a special type of categorical. Checking all the distributions with visualization and summary statistics is a typical step when beginning to work with new data.
b. Now apply normalization to some of these numerical distributions. Specifically, choose to apply z-score to one, min-max to another, and decimal scaling to a third. Explain your choices of which

normalization applies to which variable in terms of what the variable means, what distribution it starts with, and how the normalization will affect it.

c. Visualize the new distributions for the variables that have been normalized. What has changed from the previous visualization?
d. Choose one of the numerical variables to work with for this problem. Let's call it *v.* Create a new variable called *v_bins* that is a binned version of that variable. This *v_bins* will have a new set of values like *low, medium, high*. Choose the actual new values (you don't need to use *low, medium, high*) and the ranges of *v* that they represent based on your understanding of *v* from your visualizations. You can use equal depth, equal width or custom ranges. Explain your choices: why did you choose to create that number of values and those particular ranges?
e. Building on (d), use *v_bins* to create a smoothed version of *v*. Choose a smoothing strategy to create a numerical version of the binned variable and explain your choices.

Problem 2 (15 points):

This is the first homework problem using machine learning algorithms. You will perform a straightforward training and evaluation of a support vector machine on the bank data from *Problem 1*. Start with a fresh copy, but be sure to remove rows with missing values first.

a. Apply SVM to the data from *Problem 1* to predict *approval* and report the accuracy using 10-fold cross validation.
b. Next, use the grid search functionality when training to optimize the C parameter of the SVM. What parameter was chosen and what is the accuracy?
c. Sometimes even if the grid of parameters in (b) includes the default value of C = 1 (used in (a)), the accuracy result will be different for this value of C. What could make that different?

Problem 3 (25 points):

We will take SVM further in this problem, showing how it often gets used even when the data are not suitable, by first engineering the numerical features we need. There is a Star Wars dataset in the *dplyr* library. Load that library and you will be able to see it (`head(starwars)`). There are some variables we will not use, so first remove *films, vehicles, starships* and *name*. Also remove rows with missing values

a. Several variables are categorical. We will use dummy variables to make it possible for SVM to use these. Leave the *gender* category out of the dummy variable conversion to use as a categorical for prediction. Show the resulting *head*.
b. Use SVM to predict *gender* and report the accuracy.
c. Given that we have so many variables, it makes sense to consider using PCA. Run PCA on the data and determine an appropriate number of components to use. Document how you made the decision, including any graphs you used. Create a reduced version of the data with that number of principle components. Note: make sure to remove *gender* from the data before running PCA

because it would be cheating if PCA had access to the label you will use. Add it back in after reducing the data and show the result.

d. Use SVM to predict *gender* again, but this time use the data resulting from PCA. Evaluate the results with a confusion matrix and at least two partitioning methods, using grid search on the C parameter each time.
e. Whether or not it has improved the accuracy, what has PCA done for the complexity of the model?

~~Problem 4 (25 points)~~ Bonus Problem (10 points)

Use the *Sacremento* data from the *caret* library by running `data(Sacremento)` after loading *caret*. This data is about housing prices in Sacramento, California. Remove the *zip* and *city* variables.

a. Explore the variables to see if they have reasonable distributions and show your work. We will be predicting the *type* variable – does that mean we have a class imbalance?
b. There are lots of options for working on the data to try to improve the performance of SVM, including (1) removing other variables that you know should not be part of the prediction, (2) dealing with extreme variations in some variables with smoothing, normalization or a log transform, (3) applying PCA, and (4) to removing outliers. Pick one now and continue.
c. Use SVM to predict *type* and use grid search to get the best accuracy you can. The accuracy may be good, but look at the confusion matrix as well. Report what you find. Note that the *kappa* value provided with your SVM results can also help you see this. It is a measure of how well the classifier performed that takes into account the frequency of the classes.
d. Return to (b) and try at least one other way to try to improve the data before running SVM again, as in (c).
e. In the end, some data are just so imbalanced that a classifier is never going to predict the minority class. Dealing with this is a huge topic. One simple possibility is to conclude that we do not have enough data to support predicting the very infrequent class(es) and remove them. If they are not actually important to the reason we are making the prediction, that could be fine. Another approach is to force the data to be more even by sampling.

   Create a copy of the data that includes all the data from the two smaller classes, plus a small random sample of the large class (you can do this by separating those data with a filter, sampling, then attaching them back on). Check the distributions of the variables in this new data sample to make sure they are reasonably close to the originals using visualization and/or summary statistics. We want to make sure we did not get a strange sample where everything was cheap or there were only studio apartments, for example. You can rerun the sampling a few times if you are getting strange results. If it keeps happening, check your process.

   Use SVM to predict *type* one this new, more balanced dataset and report its performance with a confusion matrix and with grid search to get the best accuracy.

Bonus Problem (5 points)

To understand just how much different subsets can differ, create a 5 fold partitioning of the cars data included in R (*mtcars*) and visualize the distribution of the *gears* variable across the folds. Rather than use the fancy *trainControl* methods for making the folds, create them directly so you actually can keep track of which data points are in which fold. This is not covered in the tutorial, but it is quick. Here is code to create 5 folds and a variable in the data frame that contains the fold index of each point. Use that resulting data frame to create your visualization.

```
mycars <- mtcars  # make a copy to modify
mycars$folds = 0  # initialize new variable to hold fold indices
# Create 5 folds, get a list of lists of indices.
# Take a look at this result so you understand what is happening.
# Note we are not passing the data frame directly, but a list of its
  indices created by 1:nrow(mycars).  If you don’t understand how
  that works, try the individual parts on their own first
flds = createFolds(1:nrow(mycars), k=5, list=TRUE)
# This loop sets all the rows in a given fold to have that fold’s
  index in the folds variable.  Take a look at the result and use it
  to create the visualization.
for (i in 1:5) { mycars$folds[flds[[i]]] = i}
```