Problem Set 08
Data Mining and Machine Learning – Spring 2016
Due date: 2016-04-25 13:00 (monday)

All assignments must be uploaded to the assignments tab in ClassesV2 (notice that this is **not** the dropbox) by the date and time specified. Make sure that you follow the instructions exactly as described. You may discuss problem sets with others, but must write up your own solutions. This means that you should have no need to look at other's final written solutions.

You need to turn in all of your solutions as a zip compressed file, named `netid_pset08.zip`, with your actual netid filled in in all lower case letters. This archive should contain the following two files:

- `pset08.pdf`
- `pset08.py` (and possibly, `pset08.R`)

The python code (and possibly R code) will **not** be run or autograded, but is just for showing your work for the assignment. The pdf file should contain results and answers to the questions below.

General instructions

This problem set is broken into two parts. The first uses transfer learning to do image classification with the STL-10 dataset[1] (similar to CIFIR-10, but with large images and focused on semi-supervised techniques). The second part has you apply neural networks to the Chicago crime dataset.

Note that you may use R for the non-neural network modeling portions if you are more comfortable with R (though we've been using python for three weeks; I encourage you to learn a little more python, and try to do this entire assignment in python).

I. STL-10 Classification with VGG-19

I have fit the entire training and testing data from the STL-10 dataset using the weights from the VGG-19 model. I saved the output of the convolutional layers (before applying the final 2 hidden layers and output layer). You can download the class labels and training data here:

```
http://www.stat.yale.edu/~tba3/class_data/stl10
```

To better understand specifically what is going on here, look at the following code used to construct these (you do not need to re-run this; it would take many hours using a CPU):

[1] `https://cs.stanford.edu/~acoates/stl10/`

`http://www.stat.yale.edu/~tba3/stat665/psets/pset08/pset08_preprocess_stl10.py`

To load the dataset into python, use the starter code here:

`http://www.stat.yale.edu/~tba3/stat665/psets/pset08/pset08_starter_stl.py`

As a first step, load in the datasets and do some basic exploratory analysis of the input values. Describe the (general) shapes of the variables and explain why the structure of the neural network would lead to such a shape (hint: this should only take 2-3 sentences; do not over think it).

Your goal is to build three types of models using the output from the convolution layer to predict class labels. You should do this using:

1. a dense neural network (no need to use convolutions here, as these have already been taken care of in the layers of VGG-19)
2. support vector machines
3. Lasso logistic or multinomial regression (i.e., ℓ_1-penalized)

For the second two, you may build 10 separate binary classifiers instead of one large classifier. Make sure, however, that you combine these to make final class predictions across all of the classes. Report the details of how you constructed these models (including how you set the tuning parameters for the second two). Give the mis-classification rates and compare them. How does rates compare the mis-classification rates we have observed on CIFAR-10? (Obviously this comparison is not 1-to-1, but the classes and tasks are in fact the same).

Note: You may reasonably ask why we did not use the CIFAR-10 dataset to do this transfer learning. It certainly would have been nice to complete the cycle, and see how well we can do with the power of the VGG-19 learned weights compared to the ANNs and CNN. Unfortunately, while it does okay, it is not particularly helpful because the thumbnail images are just too small for the algorithm to really help. The STL-10 dataset's higher resolution makes it a much better candidate for the task.

II. Chicago crime data and neural networks

I have converted the Chicago crime data into purely numeric matrices (all of the variables are indicators; it includes everything except communityArea, district, and ward) and put them here:

`http://www.stat.yale.edu/~tba3/class_data/chi_python`

There is starter code to help you read this into python, located here:

```
http://www.stat.yale.edu/~tba3/stat665/psets/pset08/pset08_starter_
chicago.py
```

You should be able to use these objects directly in keras.

Your goal is to construct the most predictive neural network model you can to classify crimes into the five categories. Please include the construction of your model in Keras into the pdf solution (no need to include loading the data, fitting the model, or testing the model; we just want to see the structure of your final model). Explain the process that led you to your final model. Compare the final test classification rate you got to those we saw for problem set 3 (same problem, without neural networks). Recall that the modal classification rate for the class was about 75%, and the best rate was around 76.5%.

Outside of the classification rates, extrapolate on one benefit and one drawback of using neural networks to solve this problem as compared to SVMs or penalized linear models.