DAT13 SF: HOMEWORK 2 ASSIGNMENT

Assigned: Wednesday, March 18, 2015
Due: Wednesday, March 25 by 6:00PM
Submission Method: Please push completed homework assignments to your personal homework repo on Github and submit the URL via the Google Form: http://goo.gl/forms/QBZBG4P3bm

Grading: This assignment will be graded on a numeric point scale.

The purpose of this homework is to gain deep, hands-on experience with KNN, cross validation, and selection of model parameters. Although the Scikit-learn package provides nicely packaged methods, cross-validation is such an important concept that we will implement it ourselves in this assignment. We will then use our implementation of cross-validation to select some model parameters – also called "hyperparameters" – for our KNN classifier on the Iris dataset.

DATA & CONTEXT

In this assignment, we will get hands-on practice with KNN classification. This is our first machine learning algorithm!

For this assignment, we will use the Iris dataset that we saw in the lab. This is a very well known dataset in the machine learning world. It is relatively simple, with only four features. Therefore, it is easy to develop an intuition about the data.

SUBMITTING YOUR WORK

Please push your completed homework assignment to your personal homework repo on Github and then submit the URL via the Google Form.

HOMEWORK QUESTIONS

As we proceed through the course and increase our data science familiarity and problem-solving skills, the homework assignments will become less and less structured. For this assignment, some hints are provided with the questions. The questions are higher-level than in HW1.

Questions 1 through 5 are required. Questions 6 and 7 are extra credit questions. We strongly encourage you to work through all 7 questions.

1. Implement KNN classification, using the sklearn package. We learned how to do this in class.
   See also: http://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification

2. Use the sklearn package to implement cross-validation for your classifier. **Use 5 folds for your cross-validation.**
   See also: http://scikit-learn.org/stable/modules/cross_validation.html#

3. Use your KNN classifier and cross-validation code from (1) and (2) above to determine the optimal value of K (number of nearest neighbors to consult) for this Iris dataset. Hint: This hyperparameter will be a number between 1 and 150 ☺.

4. Using matplotlib, plot classifier accuracy versus the hyperparameter K for a range of K that you consider interesting. Explain in words what you are seeing.

5. Now, write your own implementation of cross-validation in Python without using the cross-validation methods from sklearn. Cross validation is a very important concept. Implementing it yourself in Python is the best way to learn and understand it. Compare the results of your cross-validation code with your results using the cross-validation in sklearn.

6. EXTRA CREDIT 1: Using the value of K obtained in (3) above, vary the number of folds used for cross-validation across an interesting range, e.g. [2, 3, 5, 6, 10, 15]. How does classifier accuracy vary with the number of folds used? Do you think there exists an optimal number of folds to use for this particular problem? Why or why not?

7. EXTRA CREDIT 2: Write your own implementation of KNN classification in Python, without using the methods from sklearn. Compare your results with the results you obtained using sklearn.