HOUSECS 59.01: APPLIED MACHINE LEARNING
Programming Assignment #1: Regression & Model Evaluation

GUIDELINES:

Students are expected to adhere to the Duke Community Standard. If a student is responsible for academic dishonesty on a graded item in this course, then the student will have an opportunity to admit the infraction and, if approved by the Office of Student Conduct, resolve it directly through a faculty-student resolution agreement; the terms of that agreement would then dictate the consequences. If the student is found responsible through the Office of Student Conduct and the infraction is not resolved by a faculty-student resolution agreement, then **the student will receive a failing (unsatisfactory) grade for the final grade in the course**.

- Students may work on programming assignments with a maximum of one (1) other individual in the class. However, both individuals should contribute *equally* to the assignment and understand *all* parts of the code written.
- Students are expected to write their adherence to the Duke Community Standard in a README for every assignment. Students are allowed to consult others outside of their group—limited to Duke students and faculty—about the assignment only in a general way, but not actually provide/receive code to/from other students. If assistance is received from other individuals (excluding the instructors), it should be cited in the README. **Students should be prepared to explain any program code they submit.**
- It is acceptable to use *small* pieces of outside code (found on the Internet or otherwise) due to the nature of this course—but not entire methods or programs. Using open source libraries and packages is allowed. If you are concerned whether using a piece of code is within the Duke Community Standard, please ask. *All code used should be properly cited.*
- **All submissions are subject to automated plagiarism detection.** Assignments will be randomly checked using the MOSS Plagiarism Detector.

This assignment will be due on **Wednesday, October 3** and should be completed before the start of class. The policy for turning in late assignments is detailed in the syllabus. In order to receive a passing (satisfactory) grade, in addition to satisfying the attendance requirement, students must complete **all** assignments of this course with individual scores of 70% or greater.

INSTRUCTIONS:

This assignment will assess your understanding of regression and model evaluation. As part of this assignment, you will create various regression models to predict student performance in two classes, mathematics and language, from student attributes. The dataset and full description can be found here.

The final code and write-up should be turned in on Sakai. If you are part of a team, only one member needs to submit the assignment. Both members will earn the same score on the assignment unless the distribution of work is not equal.

TASKS:

1. (30 points) Begin by looking at the *mathematics* dataset. Perform a 4-1 training-test split on the dataset. Using the first 30 attributes as inputs to a linear regression model, predict the final grade received. Note that while the data for final grade is numeric, the prediction should be continuous. Report the mean least squares error on the training and test portions of the dataset.

2. (10 points) Now, include the first and second period grades as inputs to the model, for a total of 32 attributes per student. Again, perform linear regression on the 4-1 training-test split. Report the coefficient values of the recently added features. What is the effect on the mean least squares error on the training and test portions of the dataset? Include potential reasons for your findings. Why might we not want to include the first and second period grades as inputs to a model?

3. (10 points) Again, use the mathematics dataset as the training data (looking only at the first 30 attributes). Use the model to predict final grades on the language dataset, and report the mean least squares error. What might this imply about the underlying distributions of the data between the two datasets?

4. (20 points) Repeat the first task using polynomial regression models with various *n*. (Linear regression is equivalent to polynomial regression with *n*=1.) What do you notice about the relationship between the error on the training and test portions of the dataset as *n* increases? Include a graph plotting the error for each portion over the values of *n*. At what value of *n* have we begun to overfit? Should we choose the value of *n* that performed best on the training or test portion of the dataset?

5. (15 points) We now explore the effect of adding regularization to the model. Choose the value of *n* that performed best on the portion of the dataset you answered in the previous question, and add regularization for various levels of *C*. What do you notice about the relationship between the error on the training and test portions of the dataset as *C* increases? Include a graph plotting the error for each portion over the values of *C*.

6. (5 points) Recommend a regression model based on your findings on the effects of *n* and *C* in the previous questions.

7. (10 points) Suppose we performed spline interpolation on a dataset split into training and test portions. What would this imply about the error on the *training* portion? How would the error on the *test* portion likely compare to the error if we had used regression instead? Provide a brief explanation.