HOUSECS 59.01: APPLIED MACHINE LEARNING
Programming Assignment #2: Classification & Dimensionality Reduction

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 31** 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 classification and dimensionality reduction. As part of this assignment, you will create naive Bayes and support vector machine classification models to differentiate images of handwritten digits, and experiment on the effects of dimensionality reduction. The MNIST 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. (15 points) Perform a 4-1 training-test split on the dataset, and implement multiclass classification using the naive Bayes algorithm, taking vectorized versions of the images as inputs

and digit classification as output. Report the misclassification error on the training and test portions of the dataset.

2. (30 points) Perform a 7-2-1 training-validation-test split on the dataset, and implement multiclass classification using support vector machines. Explore various kernels (e.g. linear kernel, radial basis function kernel) and their respective hyperparameters when performing nested cross-validation. [Note: do *not* optimize over the test portion of the dataset.] Report the misclassification error on each portion of the dataset for the various combinations of kernels and hyperparameters that you explored, and briefly explain any observations you made.

3. (20 points) What is the optimal kernel and respective hyperparameters that you found for this dataset, and what was the accuracy on the test portion of the dataset? Include ROC curves, and associated AUC, for the naive Bayes model and the optimal SVM model. In addition, report precision, recall, and F1-score for both models. Briefly describe what each of the metrics evaluates. Are there any significant differences amongst the evaluation metrics between the two classifiers? If so, explain why you might prefer one model to the other on the basis of the metrics.

4. (35 points) Perform dimensionality reduction using principal component analysis to various lower dimensions k of the data. For each k, perform classification using the optimal kernel and respective hyperparameters you determined in the previous task, and report the misclassification error. Include a graph plotting the misclassification error over the values of k. What is the lowest value of k that you observe such that there is no significant (>1%) increase in error? At this level of k, which digit is misclassified most? For which digit? Briefly explain what these observations imply about the data and/or classifier.