CSE514 – Spring 2022 Programming Assignment 2

This assignment is to give you hands-on experience with dimension reduction and the comparison of different classification models. It consists of a programming assignment (with optional extensions for bonus points) and a report. This project is individual work, no code sharing please, but you may post bug questions to Piazza for help.

Topic

Compare, analyze, and select a classification model for identifying letters in various fonts.

Programming work

A) Data preprocessing

This dataset contains 26 classes to separate, but for this assignment, we'll simplify to three binary classification problems.

Pair 1: H and K Pair 2: M and Y Pair 3: Your choice

For each pair, set aside 10% of the relevant samples to use as a final validation set.

B) Model fitting

For this project, you must consider the following classification models:

1. k-nearest neighbors
2. Decision tree
3. Random Forest
4. SVM
5. Artificial Neural Network

For each model, choose a hyperparameter to tune using 5-fold cross-validation. You must test at least 3 values for a categorical hyperparameter, and at least 5 values for a numerical one. Hyperparameter tuning should be done separately for each classification problem; you might end up with different values for classifying H from K than for classifying M from Y.

Optional extension 1 – Tune more hyperparameters

For bonus points, tune more than just one hyperparameter per model.

2 bonus points for each additional hyperparameter, up to 10 bonus points total.

Optional extension 2 – Consider more classification models

For bonus points, suggest additional classification models to me.

If I give the go-ahead, include one for 5 bonus points or two for 10 bonus points.

C) Dimension reduction

For each of the models, implement a method of dimension reduction from the following:

1. Simple Quality Filtering
2. Filter Methods
3. Wrapper Feature Selection
4. Embedded Methods
5. Feature Extraction

Please refer to the lecture slides for more details on the methods. Implement a total of at least 3 different methods to reduce the number of features from 16 to 4. Retrain your models using reduced datasets, including hyperparameter tuning.

Optional extension 3 – Implement more dimension reduction methods

For bonus points, implement additional unique methods of dimension reduction.

1 bonus point for each additional method, up to 5 bonus points total.

IMPORTANT: You may use any packages/libraries/code-bases as you like for the project, however, you will need to have control over certain aspects of the model that may be black-boxed by default. For example, a package that trains a kNN classifier and internally optimizes the k value is not ideal if you need the cross-validation results of testing different k values.

Data to be used

We will use the Letter Recognition dataset in the UCI repository at

UCI Machine Learning Repository: Letter Recognition Data Set

(https:// archive.ics.uci.edu/ml/datasets/letter+recognition)

Note that the first column of the dataset is the response variable (i.e., y).

There are 20,000 instances in this dataset.

For each binary classification problem, first find all the relevant samples (ex. All the H and K samples for the first problem). Then set aside 10% of those samples for final validation of the models. This means that you cannot use these samples to train your model parameters, your model hyperparameters, or your feature selection methods.

What to submit – follow the instructions here to earn full points

- (80 pts total + 25 bonus points) The report
  - Introduction (15 pts)
    - (5 pts) Your description of the problem and the motivation for trying to determine the "best" classifier. Include some discussion on what factors should be considered in determining a classifier as the "best," e.g. computational complexity, validation accuracy, model interpretability, etc.
    - (5 pts) An introduction to the binary classification problems. Which pair of letters did you choose for the third problem? Which pair do you predict will be the easiest or hardest to classify?
    - (5 pts) An explanation for why dimension reduction is or is not useful for this problem. Include some discussion on which methods are "better," and what factors should be considered in determining a dimension reduction method as "good" or "bad."
  - Results (50 pts + 25 bonus points)
    - For each classifier:
    - (2 pts) Brief description of the classifier and its general advantages and disadvantages.
    - (3 pts) Figure: Graph the cross validation results (from fitting the classification model *without* dimension reduction) over the range of hyperparameter values you tested. There should be three sets of values, one for each binary classification problem.
    - (2 pts) Brief description of the dimension reduction methods used.

        - (3 pts) Figure: Graph the cross validation results (from fitting the classification model *with* dimension reduction) over the range of hyperparameter values you tested. There should be three sets of values, one for each binary classification problem.
        - (+10 bonus points) for additional hyperparameters tuned
        - (+5 bonus points) for addition dimension reduction methods implemented
        - (+10 bonus points) for additional classifiers
    - Discussion (15 pts)
        - (5 pts) Compare the performance and run time of the different classifiers on the final validation sets with either a table or a figure.
        - (5 pts) Compare the performance and run time of the different classifiers after dimension reduction on the final validation sets with either a table or a figure.
        - (5 pts) Lessons learned: What model would you choose for this problem and why? How did dimension reduction effect the accuracy and/or run times of the different classifiers? What would you do differently if you were given this same task for a new dataset? Anything else about this project that made you think?

- (20 pts total) Your code (in a language you choose) including:
    - (10 pts) The code itself
    - (10 pts) Brief instructions or documentation on where to find the specific lines that implement each of the classifiers, and each dimension reduction method.

Due date

Wednesday, April 20 (midnight, STL time). Submission to Gradescope via course Canvas.

A one week late extension is available in exchange for a 20% penalty on your final score.

About the extra credit:

The bonus point values listed are the upper limit of extra credit you can earn for each extension. How many points you get will depend on how well you completed each task. Feel free to include partially completed extensions for partial extra credit!

In total, you can earn up to 25 bonus points on this assignment, which means you can actually get a 125% as your score if you submit it on time, or you can submit this assignment late with the 20% penalty and still get a 100% as your score. It's up to you how you would prefer to manage your time and effort.