CptS 570 Machine Learning, Fall 2018
Homework #3
Due Date: Tue, Nov 13 (9:10am)

NOTE 1: Please use a word processing software (e.g., Microsoft word or Latex) to write your answers and submit a printed copy to me at the beginning of class on Oct 23. The rationale is that it is sometimes hard to read and understand the hand-written answers.

NOTE 2: Please ensure that all the graphs are appropriately labeled (x-axis, y-axis, and each curve). The caption or heading of each graph should be informative and self-contained.

1. (**15 points**) We need to perform statistical tests to compare the performance of two learning algorithms on a given learning task. Please read the following paper and briefly summarize the key ideas as you understood:

   Thomas G. Dietterich: Approximate Statistical Test For Comparing Supervised Classification Learning Algorithms. Neural Computation 10(7): 1895-1923 (1998) `http://sci2s.ugr.es/keel/pdf/algorithm/articulo/dietterich1998.pdf`

2. (**5 points**) Please read the following paper and briefly summarize the key ideas as you understood:

   Thomas G. Dietterich (1995) Overfitting and under-computing in machine learning. Computing Surveys, 27(3), 326-327.
   `http://www.cs.orst.edu/~tgd/publications/cs95.ps.gz`

3. (**10 points**) Please read the following paper and briefly summarize the key ideas as you understood:

   Thomas G. Dietterich (2000). Ensemble Methods in Machine Learning. J. Kittler and F. Roli (Ed.) First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science (pp. 1-15). New York: Springer Verlag.
   `http://web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf`

4. (**10 points**) Please read the first five sections of the following paper and briefly summarize the key ideas as you understood:

   Jerome Friedman (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), pp 1189–1232.
   `https://statweb.stanford.edu/~jhf/ftp/trebst.pdf`

5. (**10 points**) Please read the following paper and briefly summarize the key ideas as you understood:

   Tianqi Chen, Carlos Guestrin: XGBoost: A Scalable Tree Boosting System. KDD 2016.
   `https://www.kdd.org/kdd2016/papers/files/rfp0697-chenAemb.pdf`

6. (**25 points**) Empirical analyis question. Income Classifier using Bagging and Boosting. You will use the *Adult Income* dataset from HW1 for this question. You can use Weka or scikit-learn software.

   a. Bagging (weka.classifiers.meta.Bagging). You will use decision tree as the base supervised learner. Try trees of different depth (1, 2, 3, 5, 10) and different sizes of bag or ensemble, i.e., number of trees (10, 20, 40, 60, 80, 100). Compute the training accuracy, validation accuracy, and testing accuracy for different combinations of tree depth and number of trees; and plot them. List your observations.

b. Boosting (weka.classifiers.meta.AdaBoostM1). You will use decision tree as the base supervised learner. Try trees of different depth (1, 2, 3) and different number of boosting iterations (10, 20, 40, 60, 80, 100). Compute the training accuracy, validation accuracy, and testing accuracy for different combinations of tree depth and number of boosting iterations; and plot them. List your observations.

7. (**25 points**) Automatic hyper-parameter tuning via Bayesian Optimization. For this homework, you need to use BO software to perform hyper-parameter search for Bagging and Boosting classifiers: two hyper-parameters (size of ensemble and depth of decision tree).

   You will employ Bayesian Optimization (BO) software to automate the search for the best hyper-parameters by running it for 50 iterations. Plot the number of BO iterations on x-axis and performance of the best hyper-parameters at any point of time (performance of the corresponding trained classifier on the validation data) on y-axis.

   Additionally, list the sequence of candidate hyper-parameters that were selected along the BO iterations.

   You can use one of the following BO softwares or others as needed.
   Spearmint: https://github.com/JasperSnoek/spearmint
   SMAC: http://www.cs.ubc.ca/labs/beta/Projects/SMAC/

Instructions for Code Submission and Output Format.

Please follow the below instructions. It will help us in grading your programming part of the homework. We will provide a dropbox folder link for code submission.

- Supported programming languages: Python, Java, C++
- Store all the relevant files in a folder and submit the corresponding zipfile named after your student-id, e.g., 114513209.zip
- This folder should have a script file named

  ```
  run_code.sh
  ```

  Executing this script should do all the necessary steps required for executing the code including compiling, linking, and execution
- Assume relative file paths in your code. Some examples:

  ```
  ‘‘./filename.txt’’ or ‘‘../hw2/filename.txt’’
  ```

- The output of your program should be dumped in a file named “output.txt”
- Make sure the output.txt file is dumped when you execute the script

  ```
  run_code.sh
  ```

- Zip the entire folder and submit it as

  ```
  <student_id>.zip
  ```

Grading Rubric

Each question in the students work will be assigned a letter grade of either A,B,C,D, or F by the Instructor and TAs. This five-point (discrete) scale is described as follows:

- **A) Exemplary (=100%)**.
  Solution presented solves the problem stated correctly and meets all requirements of the problem.
  Solution is clearly presented.
  Assumptions made are reasonable and are explicitly stated in the solution.
  Solution represents an elegant and effective way to solve the problem and is not overly complicated than is necessary.

- **B) Capable (=75%)**.
  Solution is mostly correct, satisfying most of the above criteria under the exemplary category, but contains some minor pitfalls, errors/flaws or limitations.

- **C) Needs Improvement (=50%)**.
  Solution demonstrates a viable approach toward solving the problem but contains some major pitfalls, errors/flaws or limitations.

- **D) Unsatisfactory (=25%)**
  Critical elements of the solution are missing or significantly flawed.
  Solution does not demonstrate sufficient understanding of the problem and/or any reasonable directions to solve the problem.

- **F) Not attempted (=0%)**
  No solution provided.

The points on a given homework question will be equal to the percentage assigned (given by the letter grades shown above) multiplied by the maximum number of possible points worth for that question. For example, if a question is worth 6 points and the answer is awarded a B grade, then that implies 4.5 points out of 6.