CS7641 Assignment 1
Supervised Learning
Spring 2025

1 Assignment Weight

The assignment is worth 15% of the total points.

Read everything below carefully as this assignment has changed term-over-term.

2 Objective

The purpose of this project is to explore techniques in supervised learning. It is important to realize that understanding an algorithm or technique requires understanding how it behaves empirically under a variety of circumstances. As such, rather than implement each of the algorithms, you will be asked to experiment with them and compare their performance. This is quite involved and also possibly quite different from what you are used to; however, it is central and in many ways the essence of supervised learning.

3 Procedure

You are given two vastly different datasets. You will design two interesting classification problems after initial data exploration. For the purposes of this assignment, a classification problem is a set of training examples and a set of test examples. You will need to explain why the datasets are interesting from an ML practitioner perspective and be able to discuss context with a deeper understanding of the datasets.

You will go through the process of exploring the data, develop a hypotheses, tune algorithms you've learned about, and writing a thorough analysis of your findings. You need not implement any learning algorithm yourself; however, you must participate in the journey of exploring, tuning, and analyzing. Concretely, this means:

- You may program in any language you wish and are allowed to use any library, **as long as it was not written specifically to solve this assignment**.
- TAs must be able to recreate your experiments on a standard linux machine if necessary.
- The analysis you provide in the report is paramount.

You should experiment with these three learning algorithms on each dataset. They are:

- **Neural Networks**. You may use networks of nodes with as many layers as you like and test two distinct activation functions appropriate for your network and the data.
- **Support Vector Machines**. You must try at least two different kernel functions.
- **k-Nearest Neighbors**. You must try significant values of k for comparison.

Each algorithm is described in detail in your textbook, the assigned readings on Canvas, and on the internet. Instead of implementing the algorithms yourself, you should use libraries that do this for you and make sure to provide proper attribution. Also, note that you'll need to do some tinkering to obtain good results and graphs, and this might require you to modify these libraries in various ways.

Extra Credit Opportunity:

There is an opportunity to add 5 points of extra credit to your report. In addition to the above algorithms, you will also implement **Boosting for Decision Trees**. Be sure to use some form of pruning. You will need to explain and demonstrate how weak learners affect bias and variance. This is not mandatory and may require more time for proper analysis.

3.1 Experiments and Analysis

Your report should contain:

- A description of your classification problems, and why you feel they are interesting from an ML perspective (rather than descriptors or opinions). To be interesting the problems should be non-trivial on the one hand, but capable of admitting comparisons and analysis of the various algorithms on the other.
- The training and testing error rates you obtained running the various learning algorithms on your problems. At the very least you should include graphs that show performance on both training and test data as a function of training size (note that this implies that you need to design a classification problem that has more than a trivial amount of data) and – for the algorithms that are iterative – training times/iterations. Both of these kinds of graphs are referred to as learning curves.
- You must contain a hypothesis about your datasets. This is open-ended as each of you will have a variety of perspective on the features and attributes of the data that may or may not perform a certain way given the required algorithms. Whatever hypothesis you choose, you will need to back it up with experimentation and thorough discussion. It is not enough to just show results.
- Graphs for each algorithm showing training and testing error rates as a function of selected hyperparameter ranges. This type of graph is referred to as a model complexity graph (also sometimes validation curve). Please experiment with more than one hyperparameter and make sure the results and subsequent analysis you provide are meaningful.
- Analyses of your results. Why did you get the results you did? Compare and contrast the different algorithms. What sort of changes might you make to each of those algorithms to improve performance? How do the datasets compare with you hypothesis? How fast were they in terms of wall clock time? Iterations? Would cross-validation help? How much performance was due to data cleaning or preprocessing? Which algorithm performed best? How do you define best? Be creative and think of as many questions you can, and as many answers as you can.

Analysis writeup is limited to 8 pages. The page limit includes your citations. Citations must be in IEEE, MLA, or APA format. Anything past 8 pages will not be read. Please keep your analysis as concise while still covering the requirements of the assignment. As a final check during your submission process, download the submission to double-check everything looks correct on Canvas. Try not wait until the last minute to submit as you will only be tempting Murphy's Law.

In addition, your report must be written in LaTeX on Overleaf. You can create an account with your Georgia Tech email (e.g. gburdell3@gatech.edu). When submitting your report, you are required to include a 'READ ONLY' link to the Overleaf Project. If a link is not provided in the report or Canvas submission comment, 5 points will be deduced from your score. Do not share the project directly with the Instructor or TAs via email. For a starting template, please use the IEEE Conference template.

Update for Spring 2025

The following datasets are required for the Spring 2025 cohort. Each semester these datasets will change. This is due to a variety of reasons concerning simplicity and overuse of common ML datasets. These datasets are mid-sized and provide many angles of analysis due to the complexity of features and domain knowledge. Each dataset can be found on Canvas if access to the original download is limited. If these datasets are not used, you will receive a zero for the assignment.

- **Customer Personality Dataset**: Kaggle Repository: *Customer Personality Dataset*
- **Spotify 2023 Dataset**: Kaggle Repository: *Spotify 2023 Dataset*

3.2 Acceptable Libraries

Here are a few **examples** of acceptable libraries. You can use other libraries as long as they fulfill the conditions mentioned above.

Machine learning algorithms:

- scikit-learn (python)
- Weka (java)
- e1071/nnet/random forest(R)
- ML toolbox (matlab)
- tensorflow/pytorch (python)

Plotting:

- matplotlib (python)
- seaborn (python)
- yellowbrick (python)
- ggplot2 (R)

4 Submission Details

The due date is indicated on the Canvas page for this assignment. Make sure you have set your timezone in Canvas to ensure the deadline is accurate. We are in the Eastern Time Zone for the course.

Due Date: **Indicated as "Due" on Canvas.** Please double check you understand the correct due date.

Late Due Date [20 point penalty per day]: **Indicated as "Until" on Canvas.** The late penalty is not on a racked scale, but rather wholistic day-to-day. Meaning, if you do utilize the late penalty, you have the full 24 hours before another 20 point penalty incurs.

You must submit:

- A file named README.txt containing instructions for running your code. We need to be able to get to your code and your data. Providing entire libraries isn't necessary when a URL would suffice; however, you should at least provide any files you found necessary to change and enough support and explanation so we can reproduce your results on a standard Linux machine.
- A file named yourgtaccount-analysis.pdf containing your writeup (GT account is what you log in with, not your all-digits ID). This file should not exceed 8 pages.
- A 'READ ONLY' link to share your Overleaf Project link and final commit for source code in your personal repository on Georgia Tech's private GitHub. These can be in your README.txt or commented to Canvas submission.

You may submit the assignment as many times as you wish up to the due date, but, we will only consider your last submission for grading purposes.

Note: we need to be able to get to your code and your data. Providing entire libraries isn't necessary when a URL would suffice; however, you should at least provide any files you found necessary to change and enough support and explanation so we can reproduce your results on a standard linux machine.

5 Feedback Requests

When your assignment is scored, you will receive feedback explaining your errors and successes in some level of detail. This feedback is for your benefit, both on this assignment and for future assignments. It is considered a part of your learning goal to internalize this feedback. We strive to give meaningful feedback with a human

interaction at scale. We have a multitude of mechanisms behind the scenes to ensure grading consistency with meaningful feedback. This can be difficult, however sometimes feedback isn't always as clear as you need. If you are confused by a piece of feedback, please start a private thread on Ed and we will jump in to help clarify.

Previously, we have had a different rescore policy in this class which usually resulted in the same grade or lower. Many times there is a disconnect between what may be important or may have been missed in analysis. For this reason, we will not be conducting any rescore requests but rather work towards providing feedback understanding.

6 Plagiarism and Proper Citation

The easiest way to fail this class is to plagiarize. **Using the analysis, code or graphs of others in this class is considered plagiarism**. The assignments are designed to force you to immerse yourself in the empirical and engineering side of ML that one must master to be a viable practitioner and researcher. It is important that you understand why your algorithms work and how they are affected by your choices in data and hyperparameters. The phrase "as long as you participate in this journey of exploring, tuning, and analyzing" is key. We take this very seriously and you should too.

What is plagiarism?

If you copy any amount of text from other students, websites, or any other source without proper attribution, that is plagiarism. The most common form of plagiarism is copying definitions or explanations from wikipedia or similar websites. We use an anti-cheat tool to find out which parts of the assignments are your own and there is a near 100 percent chance we will find out if you copy or paraphrase text or plots from online articles, assignments of other students (even across sections and previous courses), or website repositories.

What does it mean to be original?

In this course, we care very much about your analysis. It must be original. Original here means two things: 1) the text of the written report must be your own and 2) the exploration that leads to your analysis must be your own. Plagiarism typically refers to the former explicitly, but in this case it also refers to the latter explicitly.

It is well known that for this course we do not care about code. We are not interested in your working out the edge cases in k-nn, or proving your skills with python. While there is some value in implementing algorithms yourselves in general, here we are interested in your grokking the practice of ML itself. That practice is about the interaction of algorithms with data. As such, the vast majority of what you're going to learn in order to master the empirical practice of ML flows from doing your own analysis of the data, hyper parameters, and so on; hence, you are allowed to use ML code from libraries but are not allowed to use code written explicitly for this course, particularly those parts of code that automate exploration. You will be tempted to just run said code that has already been overfit to the specific datasets used by that code and will therefore learn very little.

How to cite:

If you are referring to information you got from a third-party source or paraphrasing another author, you need to cite them right where you do so and provide a reference at the end of the document [Col]. Furthermore, "if you use an author's specific word or words, you must place those words within quotation marks and you must credit the source." [Wis]. It is good style to use quotations sparingly. Obviously, you cannot quote other people's assignment and assume that is acceptable. Speaking of acceptable, citing is not a get-out-of-jail-free card. You cannot directly copy text flippantly, but cite it all and then claim it's not plagiarism just because you cited it. Too many quotes of more than, say, two sentences will be considered plagiarism and a terminal lack of academic originality.

All citations need to be in IEEE, MLA, or APA format.

Your README file will include pointers to any code and libraries you used.

If we catch you...

We report all suspected cases of plagiarism to the Office of Student Integrity. Students who are under investigation are not allowed to drop from the course in question, and the consequences can be severe, ranging from a lowered grade to expulsion from the program.

7 Version Control

- v1.0 - 01/10/2025 - TJL finalized A1 for Spring 2025 term.

References

[Col] Williams College. *Citing Your Sources: Citing Basics*. URL: https://libguides.williams.edu/citing.

[Wis] University of Wisconsin - Madison. *Quoting and Paraphrasing*. URL: https://writing.wisc.edu/handbook/assignments/quotingsources.

Assignment description refactored and written by Theodore LaGrow. Updated for Spring 2025 by Theodore LaGrow. Modified for LaTeX by John Mansfield.