CS339 Artificial Intelligence
Project 3: Supervised Learning

The learning goals for this project are:

- To gain experience with the supervised learning paradigm.
- To practice with a large or moderately large dataset.
- To learn the peculiarities of one of the software toolkits.

Algorithm:
For this project I will ask that you conduct an experiment in the supervised learning paradigm. I imagine this will either be a decision-tree based algorithm (simple decision trees, random forests, boosting, bagging) or a neural network based algorithm (deep learning, multi-layer perceptron, perhaps support vector machines). If you wish to use a different algorithm, come see me with your idea.

Dataset:
You are to seek out an appropriate dataset for this project. Try to look for datasets that are at least moderately large. The dataset should have a dozen or more inputs (high dimensionality) and have several hundred or several thousand samples. The data should have a desired target which serves as the learning goal. The target can be a category (classification) or a real value (regression). The target can be a single variable or a multi-valued vector.

A significant part of the project may be data preparation, depending on which dataset you choose. You will have to format and modify the data to fit with the algorithm you are using. You may have to deal with missing values or incomplete datasets.

You can find a wide selection of datasets from the UCI machine learning repository (google UCI machine learning repository). There are datasets to download and descriptions of the data. Many of the datasets have references to research papers and projects that have operated on that data. Keras has some of these UCI datasets built-in to the toolkit distribution. Be sure to cite your dataset source.

Software:
I recommend that you use the built-in toolkits for this project. Coding a decision tree or a neural network is a substantial task and is prone to a lot of errors and lengthy debugging. Furthermore, the toolkits have optimizations that allow the algorithms to run much faster than naive hand-written code from someone learning these algorithms for the first time. The sklearn toolbox has good tree based algorithms. The keras toolbox has good deep learning and neural network algorithms. I have provided sample code for each in class and on notebowl.

Using Resources:
For this project I am significantly relaxing the restrictions on using resources from outside sources. You are permitted to search online for examples of code use for these kind of algorithms. You may find complete solutions, partial solutions, videos, tutorials, books, etc. Feel free to use these resources. If you do follow closely an existing online source, I ask that you avoid a copy/paste operation. Please type the code by hand even if you are following it closely as you will learn more about the steps as you transcribe them. Cite any source that you find and copy closely in your paper references section. Make it clear in your introduction what you borrow from another source and what new thing you add to the project (see below).

Novelty:
While I am allowing you to closely follow other examples that you find, I ask in return that you insert at least one item of novelty into your experiment. For example, suppose you find a neural network algorithm using keras to solve a learning problem with a specific dataset. You might introduce novelty by following their example and then modifying the algorithm in some way to experiment with the outcome. Maybe you change the network structure (number of units, number of layers, kind of activation function). Maybe you change the training routine. Maybe you introduce something else new.

Scope:
Keep the project manageable. You do not have a lot of time to complete the project, so start with an example that you know works well and keep the novelty aspect reasonable in size. This does not have to be a big coding project. Set aside half of the time available to do the writing. Build in a complete draft and full revision cycle into the second half of your project. The temptation will be to expand the experiment to take too much time and then to neglect the work that needs to be done on the writing aspect.

Partners:
You may work alone on this project or you may partner with one other person of your choosing. If you partner with someone, be sure that you share the work equitably. Each partner will share the coding. Each partner will write their own paper independently (do not collaborate or share on this part). Do include both names as authors in each of the papers you submit. You may share graphs or figures produced by the software.

Deliverables:

- You will submit your code for the project. This can be a python program or a jupyter notebook.
- Submit any resources you need to run your code. This includes datasets, inputs, images, etc.
- Submit a pdf of your paper. Your paper should be written in LaTeX.

Evaluation Criteria:

- Do you have an appropriate dataset? Is the learning problem non-trivial? Has the dataset been prepared appropriately for use in your algorithm?
- Have you chosen an appropriate algorithm for solving this problem?
- Are you using a software toolkit appropriately?
- Have you followed good coding practices?
- **Does your paper and your project have an obvious experimental question? Is there a research question that is being posed and being investigated by collecting experimental evidence?**
- Do you collect appropriate evidence to answer your experimental question?
- Do you provide analysis of the evidence?
- Do you support your analysis with appropriate graphs, tables, charts and other visual aids? Are those aids formatted and used correctly?
- Does your paper have a thesis that is based on your research question?
- Does the paper present a good structure and organization?
- Does the paper correctly identify a target audience and meet the needs of that audience?
- Does the writing exhibit clarity?
- Does the writing exercise precision of expression?
- Does the paper appropriately cite sources for data and for previous work?