Final Project Proposal
Due March 22 11:59pm

Submit the following information in PDF form to Blackboard. All team members must submit, but your submissions can be identical within your team. Your answers can be concise as you like, as long as you give us a clear idea of what you're doing.

Elevator Pitch. Describe what you want to do in one sentence.

Topics. Choose the pieces of technology or ideas that your project will embody from the list at the end of this document. The total scope of your project will depend on the size of your group.

1-person team:	**3 points of topics**
2-person team:	**3 points of topics**
3-person team:	**4 points of topics**
4-person team:	**6 points of topics**

These point values reflect my preference for 2-person or 3-person teams.

You may ask my permission to introduce a topic not on the list, but this must occur more than 24 hours before the proposal deadline.

Teammate(s). List all members of your team. Your **teammates should submit an identical proposal** to Blackboard, and will lose points if they forget to do this.

Context. If you are interacting with a technology we may not have heard of, a game we may not have played, or any other kind of jargon or specialized knowledge relevant to your project, provide links or descriptions here. (If you think the rest of your proposal is self-explanatory, you don't need to say anything here.)

Language/Tools. What language will you use to code? Do you know of existing libraries or projects that do what you want to do? If you plan to use one of these libraries or projects, where does your "value add" on top of that existing code come in? (You will be allowed to use code from the Internet in your project, but you must cite your source in your code and final paper, and your grade will depend on how much you added to what was there.)

Technical Source. Cite one written technical resource that can help with what you are trying to do (e.g. pseudocode, ML training advice, etc). You can use Russell and Norvig for this if you give the relevant sections you will use.

Full disclosure. Identify any work you have done on this project prior to joining the class, or for another class (including thesis projects and independent study).

Plan. Describe a reasonable plan to finish your project by the last week of class. Include at least 2 milestones (dates & deliverables), and take into account that you will need to write a short summary paper and do a "lightning talk" in the last week.

If your plan is problematic enough, we may revise your milestones and/or actually require the work for them to be submitted on their target dates. But normally, milestones will be just for your own planning benefit.

Proposed evaluation. How will you demonstrate that your project works well?

The sections above are what you need to fill out by the proposal deadline. The matter below is to explain the point system for topics more fully.

Appendix: More on topics and points

The general expectation is that most projects will do a 2 point topic with 1 or 2 1-point topics to flesh it out (depending on group size). You should try to do topics that naturally go together. You may reuse homework code if applicable.

1 point topics

- Execute an existing machine learning implementation, such as k-nearest neighbors in scikit-learn or code for a more advanced method that you found online (typically this will just be to compare against your project's performance)
  - *Note that all other point values assume you're writing the code yourself*
- Measure the effects of varying a particular hyperparameter on the accuracy of a method (could be repeated for multiple hyperparameters)
- Collect your own novel dataset - for example, with a survey
- Make use of a semantic space like BERT or word2vec to get semantic features
- Get your system to work with a sensor collecting data in real time (including a camera or mic; repeatable for different sensors)
- Implement a genetic algorithm
- Provide significant UI polish (for example, visualization of the algorithm's progress or options)
- Run experiments with human subjects to determine what results they prefer
- Implement Monte Carlo Tree Search on a new game (and implement the game as well)
- Implement minimax with alpha-beta pruning
- Implement AdaBoost from the pseudocode
- Implementing any miscellaneous "easy" AI method, such as k-nearest neighbors

2 point topics

- Use Pytorch or Keras to try different neural architectures to achieve the best performance on some supervised learning task
- Train a Q-learner with a neural network backend (you can use a library)
- Collect a novel dataset that requires significant cleaning or effort to collect
- Implement a variant of A*, such as IDA*
- Implement expectiminimax for a game that involves chance
- Fine-tune BERT to perform a novel task
- Train an LSTM to predict values in a time series
- Design a Bayesian network to reason about a web of dependent events, perhaps in a Bayesian reasoning environment like PyMC3
- Implement any "medium-difficulty" AI method, on par with those listed here

3 point topics

- Implement a classic logic-based planner
- Design a generative adversarial network
- Perform supervised learning using video or audio
- Implement any "advanced-difficulty" AI method, on par with those listed here

Domains & datasets for inspiration

Kaggle, a site for machine learning contests, has many public datasets:
https://www.kaggle.com/datasets

The UCI Machine Learning Repository is a little old, but generally easy to use:
https://archive.ics.uci.edu/ml/index.php

Scikit-learn datasets are convenient for basic tests of ML approaches:
https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets

CIFAR-10 and CIFAR-100 picture datasets provide tiny pictures for faster learning:
https://www.cs.toronto.edu/~kriz/cifar.html

Some interesting environments for reinforcement learning: https://gym.openai.com/

The Arcade Learning Environment is set up to try Atari games:
https://github.com/mgbellemare/Arcade-Learning-Environment

Keras is a high-level deep neural network library with code for many starter experiments:
https://keras.io/examples/

Don't feel constrained by what's here -- if you want to do something different, just ask.

Have fun being creative!