# CS 374 Assignment #11-12
## Final Project

Part 1 due the week of May 10, 2021
Part 2 due at the time of the final presentations

—

This semester you have learned about many aspects of machine learning – fundamental algorithms for classification and clustering, theoretical foundations, evaluation methodology, societal implications, etc. Over the next few weeks you will have the opportunity to craft personalized assignments that will allow you to explore topics of particular interest to you. Ideally, you will do this assignment with your tutorial partner, but other configurations are possible. For instance, a couple of tutorial groups could get together to do a larger project. In the "everyone is virtual" configuration post Thanksgiving, the latter might be more difficult, but I'm open to such requests.

- Part 1: Your deliverable for the week of May 10 will be a project proposal written in the form of a CSCI 374 assignment. It should include an introduction, including motivation (i.e., why is this a valuable topic? why do we care?). It should also include a list of readings. Finally, it should include a description of what you will do beyond any reading. Complete a set of problems? Write a program and demonstrate it? Perform an empirical analysis and write it up? Give a presentation? Note that on some level each of you will do the latter. That is, Part 2 of this assignment requires that you present your project.
- Part 2: This is where you will present (and turn in) the work you have done. We will try to meet in larger groups so you can see what others have done, but again, this might be impossible in an "all virtual" setting. If we can't find times to get larger groups (two or three pairs) together, we'll keep the usual meeting configuration. We will send out an email getting your availability and preferences.

In the following sections, we set out some ideas for projects. These are merely intended to get you thinking about the range of possibilities. By providing them here, we do not mean to imply that you must select one of them. **Note that you are more than welcome to drop by Office Hours to discuss your ideas. We are happy to provide pointers to papers and other resources.**

# 1 Idea 1: Apply Machine Learning to a Real Data Set

You now have a solid set of machine learning tools, so why not try them out on some real data. There are many great sources of data, including the sites below, which overlap with the set of sites I've identified on the course "Resources" page.

- The UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/index.php)
- Kaggle (https://www.kaggle.com/) – a data science website that includes data sets, competitions, etc.
- Google data sets (https://research.google/tools/)
- Microsoft Research open data (https://msropendata.com/)
- Miscellaneous data made public by various institutions (such as the US Government) and companies.

We've identified a few data sets that are rich and interesting in different ways, just as inspiration. Below you'll find descriptions of the data sets as well as a set of special rules, should you decide to pursue a project along these lines.

## 1.1 Sentence Classification http://archive.ics.uci.edu/ml/datasets/Sentence+Classification

From the "README" for the data set: This corpus contains sentences from the abstract and introduction of 30 scientific articles that have been annotated (i.e. labeled or tagged) according to a modified version of the Argumentative Zones annotation scheme. These 30 scientific articles come from three different domains:

(1) PLoS Computational Biology (PLOS), (2) The machine learning repository on arXiv (ARXIV), (3) The psychology journal Judgment and Decision Making (JDM).

Also from the "README": Argumentative Zones (AZ) is a scheme for classifying (i.e. annotating, labeling, or tagging) sentences according to function. There are seven labels in the original AZ scheme:

1. AIM: "A specific research goal of the current paper"
2. TEXTUAL: "Statements about section structure"
3. OWN: "(Neutral) description of own work presented in current paper"
4. BACKGROUND: "Generally accepted scientific background"
5. CONTRAST: "Statements of comparison with or contrast to other work; weaknesses of other work"
6. BASIS: "Statements of agreement with other work or continuation of other work"
7. OTHER: "(Neutral) description of other researchers' work"

There is a lot of data here: labeled articles, unlabeled articles, lists of words that might be associated with the various classes, etc. But the data are in quite raw form. That is, the examples (i.e., the sentences) are not expressed as vectors of attribute values. It would be up to you to decide what the attributes should be. If you wanted to pursue a project involving text, you'd want to do some reading on extracting features from text.

### 1.2 LSVT Voice Rehabilitation https://archive.ics.uci.edu/ml/datasets/LSVT+Voice+Rehabilitation

As described by the contributor of the data set, the aim here is to assess whether voice rehabilitation treatment leads to phonations considered 'acceptable' or 'unacceptable'. (So this is a binary classification problem). The paper associated with the data set shows that it's possible to achieve 90% accuracy, so your goal will be to see whether you can do better. One of the challenges here is that there are 309 attributes but only 126 examples.

### 1.3 Urban Land Cover https://archive.ics.uci.edu/ml/datasets/Urban+Land+Cover

This problem involves "classification of urban land cover using high resolution aerial imagery." As the contributor says, there are a low number of training samples for each class (14-30) and a high number of classification variables (148), so it would be an interesting data set for testing feature selection methods.

### 1.4 Confused Student EEGs https://www.kaggle.com/wanghaohan/confused-eeg

This dataset includes information from EEGs of students performed while they were watching educational videos. The goal is to classify the students as confused or not.

### 1.5 YouTube Audio Clips https://research.google.com/audioset/dataset/index.html

This dataset is a large-scale collection of human-labeled 10-second sound clips drawn from YouTube videos. This could be a lot of fun, but think hard about how much data you actual want to download. (Hint: not the whole thing!)

### 1.6 Special Rules for Projects that Apply Machine Learning to Real Data

If you plan to try your hand at something like this:

- Once you've decided on a data set, let me know. Some data sets are quite large to begin with, and as you generate training sets, test sets, and results, you'll be using both a great deal of disk space and processing time. If you're using your own laptop, that's up to you to manage. But if you're planning to ssh into the lab machines, we'll want to coordinate with Mary to be sure you have the resources you need while not causing problems for other students or for the lab in general.

## 2 Idea 2: Explore a New Learning Algorithm

We've explored classification algorithms quite extensively, but there are many more algorithms out there. For instance, you might want to learn more about clustering algorithms by implementing a hierarchical clustering algorithm. Or you might want to explore reinforcement learning by implementing Q-learning.

## 3 Idea 3: Explore an "Issue" in Machine Learning

Over the course of the semester we've discussed situations that are problematic for our algorithms, such as the "curse of dimensionality" or the difficulty of learning from unbalanced training data. You might want to explore techniques for handling these problems. For example, you might read up on feature selection techniques, implement one or two, and then apply them to a variety of data sets and analyze the results.

This semester we've also focused primarily on nominal-valued data that are static in time. You could explore techniques for dealing with time series data.

## 4 Idea 4: Explore a Machine Learning Toolkit

This is a very practical idea, but it might also allow you to explore a class of algorithms that we haven't explored through implementation. For instance, you might choose to learn about PyTorch (https://pytorch.org/) or TensorFlow (https://www.tensorflow.org/) both of which are well suited for learning with deep networks.

### 4.1 Special Rules for Projects that Involve Exploring a Toolkit

- Both PyTorch and TensorFlow are available in our lab. Do not attempt to download any other significant toolkits to lab machines.
- If you plan to use your own laptop, that's great. Just be sure to ask yourself, "do I really want to install everything I have to install to make this work?"

## 5 What have students done in previous years?

Here's a sampling of some projects from previous years.

**Classifying Classical Music by Composer** This project required first gathering midi and mp3 files, and extracting meaningful features (using jSymbolic, Music21, and Python; 25 hours and 24 GB on a laptop). The students then started their empirical work by following the approach taken in a 2015 research paper by Herremans et al. They ultimately added 22 new features to the paper's 12, but then applied feature selection techniques to choose the most informative features. In the remainder of their project, they performed experiments with various classifier learning algorithms and analyzed the results using several different comparison metrics.

**Automatic Music Genre Classification with Deep Learning** This project involved identifying music genre from music samples as well as high-level music features. The students considered two data sets (GTZAN and the Million Song Dataset) and implemented both convolutional and fully-connected deep networks in Tensorflow. The students needed to do some work with their data first. For instance, the Million Song Dataset did not include genre classification, so they applied the work of a research group at TU Wien. They also did additional data collection to supplement the information in the Million Song Dataset. They presented the architecture of their networks, various ways of considering the data, as well as results from several experiments.

**Do Hungry Mice Dream of Electric Wires?** The inspiration for this project was work being done in Prof. Carter's lab by a thesis student who was a mutual friend of the machine learning students. The biology thesis student had to spend long hours classifying mouse EEG and EMB data as "Wake", "Non-REM", and "REM". They drew inspiration from a research paper that discussed machine learning for classification of human EEG data. Two aspects of the data made it quite different from many of the data sets analyzed in class: (1) EEG data is fundamentally sequential and (2) the data contained very few REM instances. The students applied techniques such as Fast Discrete Fourier Transform, Principle Component Analysis, and other techniques to design and select appropriate features. They then trained several algorithms using different sets of attributes and compared their results.

**Convolutional Neural Networks** This project involved learning about convolutional neural networks. The students did background reading, prepared a thorough and clear presentation, which included intuition as well as mathematics, and then implemented three neural networks (both convolutional and not) using Tensorflow, allowing them to perform a comparison on the MNIST data set.

**Association Rules and the Apriori Algorithm** This pair of students elected to investigate one of the "top ten data mining algorithms" that we did not explore at all in the course: Apriori. They learned about association rules and the algorithm. They then implemented it, applied it to various data sets, and presented both the algorithm and results.

**Random Forests** Another pair of students chose to implement and test Random Forests, as proposed by Breiman.

**Dimensionality Reduction** This project involved consideration of three different techniques for dimensionality reduction: Principle Component Analysis, Linear Discriminant Analysis, and Autoencoders. The students implemented these approaches, applied them to several high-dimensional data sets, and then applied k-NN (at various levels of k) to compare the effectiveness of the approaches.

**"AlphaGo" for Othello** This group of three students implemented an Othello-playing system, inspired by DeepMind's AlphaGo. Their system learned to play well by playing itself and included a deep neural network and Monte Carlo Search trees to generate move probabilities.

## 6 So many other ideas...

For inspiration, you might skim articles in high quality machine learning conferences and journals. You can find pointers to those from the course "Resources" page.