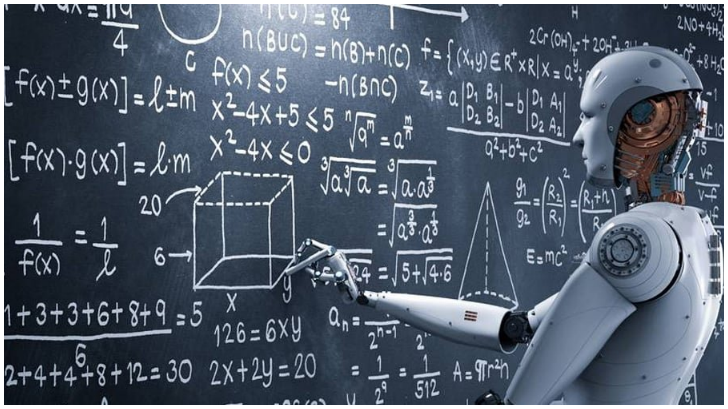
Introduction
ML_2022: Machine Learning

https://people.sc.fsu.edu/∼jburkardt/classes/ml_2022/intro_lecture/intro_lecture.pdf



"Epigram #63: When we write programs that learn, it turns out that we do and they don't."
Alan Perlis, first recipient of the Turing award, first chair of the CMU Computer Science Department, and author of 'Epigrams on Programming'.

Machine Learning

Everyone's going crazy for machine learning. We can join the mob, but let's make sure we know where we are going!

- *what are some examples of machine learning in use today?*
- *why does machine learning seem to have 50 different names?*
- *why is machine learning causing turmoil in universities?*
- *what skills are useful for a machine learning learner?*
- *what are some of the topics in a machine learning course?*

1 What is Machine Learning?

A useful definition of machine learning relates it to computer applications whose development, adaptation, or use is heavily influenced by data.

Machine learning still is a kind of computer programming. The point being made here is that traditionally, we usually think of programming as involving a programmer who sits down and starts writing a program based on some theory or model, and expects things to work just fine.

Machine learning suggests a more trial and error approach, looking at preliminary results, and adjusting theory with the data from previous experience. So in some ways, when we start an investigation, not only don't we know the answer, we also don't know the program. At least some of the features of our program will change as a result of data we already have collected, or results we encounter in our first trials of the program. There will also be a reliance on approximation, iteration, and randomness.

2 The Machine Learning Carnival

We should realize that the results of machine learning are all around us:

- A long time ago, web searches were responded to by people, or by lists that people constructed. This was incredibly expensive, slow, and limited in scope. Google became the king of browsers by discovering a **search engine** that came up with excellent matches to your search request, with a speed that actually should be impossible. And the search engine worked in any language, and didn't have any idea what your search string actually meant.
- A **self driving car** has to "solve" the problem of getting from here to there efficiently, at a safe speed, without breaking any traffic rules or running over pedestrians. How does it know where it is, whether to change lanes, speed up or slow down, or hit the brakes?
- Netflix, Spotify, and other subscription services need you to be hooked. So after you "consume" one item, they offer you more, which they are pretty sure you will be interested in. But they don't know you, do they? A **recommender application** is making these decisions, after reviewing things you have already liked, and comparing them to things that other people liked, to make a model of you.
- Your phone recognizes your voice and "understands" your commands. A sophisticated **voice recognition** application handles this process. This was not done by a programmer writing a simple sequence of procedures!
- Your email program has a "spam folder" full of messages that are probably of no interest to you. A **spam filter** has already gotten rid of messages that are certainly of no interest, or that contain harmful viruses. Sometimes, the spam filter makes a mistake, but how in the world does it look at a message that has never been seen before, and decide to delete it, quarantine it, or let you see it?

3 A Million Names

Machine learning is so flexible that it has evolved into hundreds of areas, each developing its own names, algorithms, and traditions. Here are a few of the more common names involved:

- artificial intelligence (AI);
- big data;
- bioinformatics;
- compressed sensing;
- computational neuroscience;
- computer vision;
- data analytics;
- data assimilation;
- data driven science;
- data science;
- deep learning;
- dimensionality reduction;
- expert systems;
- exploratory data analysis;
- facial recognition;
- image analysis;
- information technology (IT);
- intelligent retrieval;
- knowledge engineering;
- machine learning (ML);
- machine translation;
- machine vision;

- mathematical neuroscience;
- mobile computing;
- natural language processing (NLP);
- neural networks;
- predictive data analytics;
- robotics;
- speech recognition (Alexa, Google Home, Siri);

4 Traditional origins, modern development

Machine learning borrows and adapts ideas and techniques from traditional disciplines, particularly:

- computer science;
- mathematics;
- probability;
- statistics;

Thus, without realizing it, you may already be familiar with many ideas and techniques that are prominently used in machine learning.

The most notable new feature of machine learning is that these traditional methods are chosen and adapted in order to be applied to problems associated with really big sets of data. In many cases, a standard textbook approach learned in traditional classes breaks down when the problem size becomes extreme. Then it may be necessary to switch to a method that can only get an approximate answer, that sneaks up on the solution using iteration, or that uses an approach that is "low order" but unlikely to break down because of problem size.

Some of these algorithms use randomness to do their work, or can operate in parallel, exploring many options at the same time, or rely on the power of modern GPU systems, or special sparse techniques for reducing the effective size of a massive set of input data.

For applications like self driving cars, there is a very high premium on quick and accurate responses as new data is received (someone jumping in front of the car!).

5 Machine Learning Careers

Universities are desperate to hire new faculty with a machine learning background; Every department wants to offer courses in machine learning. Departments fight over whether machine learning belongs exclusively to their territory. Universities are creating new degree programs and even regrouping some departments into new institutes or schools to focus on machine learning.

- For over thirty years, the school of library science at **the University of Pittsburgh** has refused to die along with libraries; First it became the school of library and information science; now it has joined with the Computer Science department and others to form the new gigantic *School of Computing and Information.*
- **Virginia Tech** formed an interdisciplinary program in computational modeling and data analytics, involving Mathematics, Statistics, Computer Science, and Computer Engineering. The individual departments complained that this program was sucking up all their best students. Now that program has become part of a new *Academy of Data Science.*
- **Florida State University** was an early contestant, with an interdisciplinary School of Computational Science. This became a Department of Scientific Computing. Now that department is part of an interdisciplinary *Data Science* program involving the departments of Mathematics, Computer Science, Scientific Computing, Statistics and Philosophy(!)

Why are universities trampling over the objections of individual departments to create these new programs? Students are demanding them! What do students and their parents know about machine learning? They believe that degrees with a machine learning aspect guarantee good job offers from companies across the spectrum of technology, finance, industry, and government labs.

This doesn't make math, computer science, probability or statistics any less valuable as a field of study. But it does mean that a student would be wise to pick up some skills and class experience as preparation for a career that may include machine learning.

Such valuable skills and areas of expertise include:

- **Python:** numpy, scipy, matplotlib, PIL, pandas, seaborn, keras, scikit-learn, tensorflow, Jupyter
- **Programming:** R, Excel, HTML/JavaScript, C/C++
- **Statistics:** Descriptive statistics, inferential statistics, design of experiments;
- **Mathematics:** Algebra, Calculus, Linear Algebra, Optimization
- **Data Wrangling:** Python(pandas), Hadoop/Spark/MongoDB, SQL
- **Visualization:** Python(matplotlib,seaborn), R(ggplot), d3.js

Udacity provides a nice checklist of the skills a data analyst should have. Download this 18 page guide at:

https://www.udacity.com/blog/data-analyst-skills-checklist-eguide

6 Machine Learning Algorithms

Machine learning is commonly divided into three subfields, known somewhat mysteriously as *supervised learning*, *unsupervised learning*, and *reinforcement learning.* Of course, there are always some oddball topics that could go into a fourth *Other* category, but we won't worry about those just yet. These vague topic areas become a little more clear when we list some of the underlying algorithms that belong to each one:

- **Supervised Learning:**
  - Decision trees;
  - Naive Bayes;
  - Least squares regression;
  - Logistic regression;
  - Neural Networks;
  - Support Vector Machines;
  - Ensemble methods;
- **Unsupervised Learning:**
  - Clustering;
  - Principal Component Analysis;
  - Singular Value Decomposition;
  - Independent Component Analysis;
- **Reinforcement Learning:**
  - Q Learning;
  - T Learning;
  - Adversarial Learning;
  - Genetic Algorithms;

7 Supervised Learning

In supervised learning, we are trying to predict the results of new data, given many old results. For instance, a realtor must decide on a reasonable selling price for a house that has just come on the market, and bases this decision on the records of past house sales in this area. A house price clearly depends on many factors such as age, number of rooms, number of bathrooms, number of floors, size of the lot, age, previous selling price and so on. For the data, we have all these values, and the selling price. For the new house, we have only the data. We want to make a sensible estimate of the selling price for this house, by finding an underlying pattern in the old data.

In machine learning, items such as age and number of rooms are called *features*, and the selling price is the *target*. Mathematically, we might think of features as `x` quantities, so that the age of a house might be called x_1, the number of rooms x_2 and so on. If we think of the target value as a `y`, we are seeking a formula $y = f(x_1, x_2, ..., x_n)$. If we can determine a function $f()$ that is a good predictor, we can simply plug in the data values for our new house and get the selling price.

Without such a formula, a realtor might simply suggest an asking price that seems reasonable. Considering a home with 2,000 square feet of living space, 9 rooms, 3 bedrooms and 2 bathrooms, that is 50 years old, has a 0.45 acre lot, and a tax assessment of $2,500. Suppose the realtor thinks that $200,000 is a reasonable asking price. To see that something might be off with this price, we can look at the corresponding asking prices of a number of neighboring homes, and search for a relationship in terms of the corresponding data.

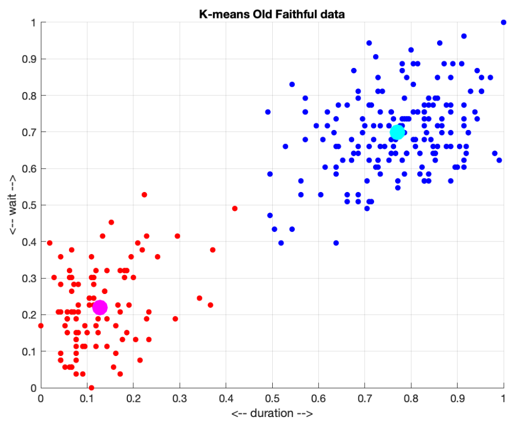
Surprisingly, a simple linear model turns out to be

```
predicted asking price in dollars =
  40631.3
+ 37.5894 * living space in square feet
- 3365.47 * number of rooms
+ 2014.06 * number of bedrooms
+ 1760.19 * number of bathrooms
- 436.631 * age in years
+ 5281.43 * lot size in acres
+ 19.3207 * annual taxes in dollars.
```

For each of the 50 houses, we can plot the actual asking price versus the predicted asking price. If the model were perfect, the dots would form a straight line. Here's what we see:

which means that our model is reasonably accurate, but not perfect. According to our model, however, the realtor's suggesting asking price for the new listing is way off. Our model suggests instead an asking price of $123,930, a big difference, but one that we can at least justify with our formula.

8 Unsupervised Learning



In unsupervised learning, we are given a large amount of data and asked to search for patterns in it. Unlike the supervised case, we don't already know the answer for some of the data. We may not know what kinds of patterns to expect, or how many separate patterns the data might divide into.

It is commonly thought that the Old Faithful geyser in Yellowstone National Park simply erupts on a regular cycle. However, the length of each eruption, and the waiting time before the next one, seems to vary in a more complicated way. By observing these times, and plotting pairs of eruption duration `(x)` versus subsequent waiting time `(y)`, a pattern does seem to emerge. If we choose to accept the division of the data into two groups, we can then average each set of data, coming up with two representative behaviors that the geyser might exhibit. We can immediately see that the data suggests that a short eruption will be followed by a short wait, as though the geyser doesn't need so long to recharge.

What is a mystery, and remains to explain, is why the geyser seems to have two main modes of behavior. This question is raised by the machine learning approach, but probably must be answered by scientific study of the geothermal structure of the geyser.

9 Reinforcement Learning

In reinforcement learning, the program seeks to achieve some goal, or maximize some reward. It must do this by carrying out a sequence of choices. Generally, the number and type of choices are so varied and large, and it is impractical to try to collect a sufficiently dense set of example data in advance. Instead, the program must stumble through many experiments, observing the current status, estimating how close it is to the goal, choosing an option it thinks will improve the status, until it finishes with some reward value. If the reward was good, then every choice along the path will get an increased likelihood of being chosen on the next trial. By carrying out this process repeatedly, the program may converge on an efficient path to an optimal reward.

`AlphaZero` and `AlphaGo` are examples of reinforcement learning programs that have beaten masters at chess and Go. As part of their training, both programs played millions of games against copies of themselves. One drawback to these kind of programs is that it is difficult to summarize the expertise of the program as a series of rules or steps. In other words, a program can play chess at the master level, but it may be impossible for us to actually learn from it how to play master chess ourselves.

For some background, see:

https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go

and

https://deepmind.com/research/case-studies/alphago-the-story-so-far