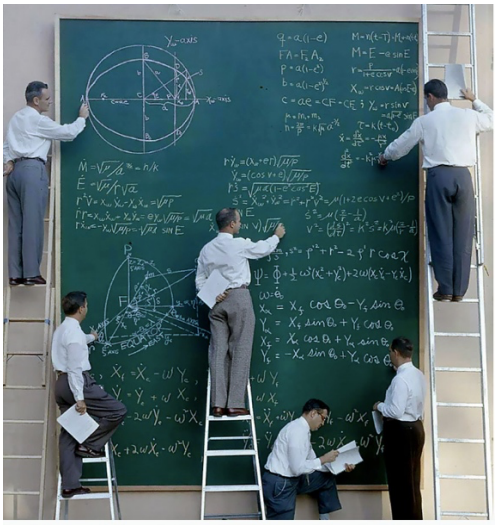


**NASA Rocket Scientists in 1960.
High School Mathematics is training students for jobs in the NASA of the 1960's**

As Steven Levitt and Jo Boaler state in their LA Times editorial from October 2019, the current high school math curriculum is focused on algebra, trigonometry and calculus which is the mathematics of rocket science. It's a curriculum put in place more than 60 years ago to enable the US to win the space race of the 1960s. They point out that it's 2022, the space race is over (and we won) and data analysis, not rocket science, should be what drives the material covered in the high school mathematics curriculum.

This has ignited a new math war. People such as Tucker Carson are weighing in. The main thrust of the argument is that data science is not rigorous enough. People can do data science without being exposed to mathematical proofs, rigor, or a lot of complicated mathematical notation. They contend that this *dumbs down* mathematics. Not only that, but they also assert it doesn't account for the interests of students who wish to study more advanced math while in high school. I believe there is a way to put this argument to bed.

Linear Algebra and the Study of the Inner Workings of Machine Learning Algorithms

I propose a *second course* in data science which would include linear algebra and the study of the inner workings of machine learning algorithms. I suspect that the primary objection people have with data science as an alternative to traditional high school math is about removing rigor and mathematical abstractions from the study of math more than anything else.

Before I go further, I want to make it clear that this would be the *second course* in a data science sequence. The beauty of data science is that it can be successfully taught to people who don't have a well-developed understanding of the math or inner workings of the algorithms it uses. As students build data science projects in an intro course, there is a good chance their

curiosity will grow. They will want to find out what is under the hood. One goal of the second course would be to satisfy this curiosity.

Linear Algebra in Data Science

One of the main tools used in data analysis is dimension reduction. A common tool for doing this is called Principal Component Analysis (PCA). PCA achieves dimension reduction by projecting the data onto an orthogonal basis determined by the principal directions of the correlation matrix of the data matrix. Deeply embedded in that dizzying mouthful of mathematical vocabulary are a handful of important, but manageable, theorems from linear algebra. Subjects covered in linear algebra include linear transformations, vector spaces, orthogonality, eigenvalues, eigenvectors, matrix decomposition, change of basis, high dimensional mathematics, and inner products. I don't expect the reader to be familiar with any of this. I'm simply pointing out that the mathematics of data science is as rich as any given branch of mathematics. An advantage the math of data science has over a typical high school math course is applications to interesting real-life problems can be readily demonstrated. Below is an example of an *unmathy* problem that applies matrix decomposition, linear transformations, eigenvectors, eigenvalues, high dimensional mathematics and change of basis. It's all done using principal component analysis. In a *Second Course in Data Science*, the math supporting PCA would be covered alongside the algorithms used.

PCA and the Wizard of Oz

Below I provide an example of PCA in action. In this case it's used to solve a problem in natural language processing. Since this article is for the general reader, I don't go into the linear algebra.

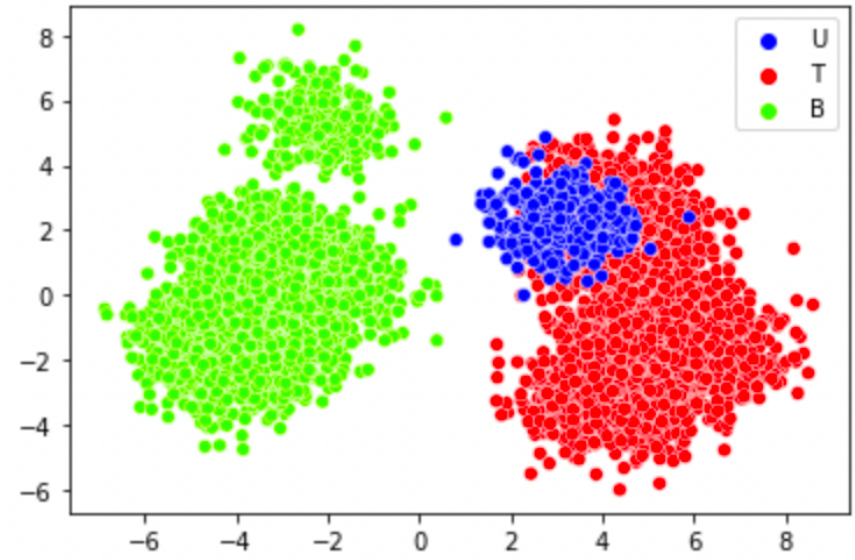
From 1900 to 1938, close to 30 novels were written that took place in the land of OZ. These stories were the Harry Potter books of the early 20th century. Frank L Baum was the original author. He wrote the first 14 stories. Starting with the 15th story, a second author came into play. In 1919, Baum became very ill and he chose Ruth Plumy Thompson to help him finish his 15th book. At the time, there was a question as who actually wrote the book. Did they work together? Did she write it alone? Baum died before the book was published. Thompson went on to write 10 more Oz books. Can natural language processing be used to suss out who wrote the 15th book?

In 2003, Jose Nilo G Binongo authored "*Who Wrote the 15th Book of Oz?*" I read the paper and via the python packages Spacy and Sklearn, I replicated his method and came up

with results that were in line with his. I performed the replication to ensure that I fully understood his method.

The books are freely available at the Gutenberg Project. They are formatted as .txt files. I took 100 random samples of 1000 words each from each book and extracted ***stop*** words from each sample. Examples of *stop* words are *then, once, too, very, about, against, etc.* Although these words are necessary, they don't convey much meaning. Therefore, authors probably use these words unconsciously. The thinking is that this leads to individualized usage patterns.

I created vectors with 294 entries -- one entry for each of the *stop* words. Counts were calculated for the number of times a word was present in a random sample. The final result was a dataset with 2400 vectors with 294 dimensions or coordinates. PCA reduced this high dimensional data down to a dataset with 2400 two dimensional vectors, from which I created the plot below.



'U' (blue) stands for *unknown author*. 'T' (red) stands for *Thompson*. 'B' (green) stands for *Baum*. Note that the samples for *unknown* overlap the samples for *Thompson*. This is strong evidence indicating that the 15th book was written by *Thompson.*

The Demystification of the Algorithms Used in Data Science

Andrew Ng is a well-known AI professor at Stamford. He equates AI and machine learning (ML) with electricity. He sees AI and ML as approaching the prevalence of electricity in every-day life. With that in mind, it seems that a general understanding of ML and AI are as important as knowledge of biology, chemistry and physics, each of which are taken by every

college bound high school student. Solid understanding requires demystifying ML and AI by having students learn the inner workings of the algorithms involved. Achieving this understanding is an intellectual challenge on par with high school calculus. In this *second course*, the inner workings of the main algorithms of data science would be taught in detail. Python code for the inner workings of *k-nearest neighbors, linear regression, logistic regression random forest, gradient descent etc.* are freely available online. Because of this, a project-based approach to assess student knowledge of the inner workings of these algorithms would not be reliable. An assessment method came to me as I was walking through the algorithms to ensure I could understand them. I found myself adding comments to the code as a comprehension aid. A lightbulb went off. A quiz/exam/assessment could take the following form:

1) Hand out uncommented versions of a few algorithms.
2) Instruct students to fill in comments in a few designated regions of the code.
3) The grade would be based on student understanding conveyed in the comments.

And Finally - Matrix Operations and Neural Networks

Due to packages such as Keras, students can build neural nets without having to be exposed to the matrix operations that take place under the hood. When the main goal is to create a net to solve a given problem, this is as it should be. But, if the goal is for students to build a solid understanding of what goes on under the hood, it's a must to have them construct small neural nets directly from matrices. Doing this would also add mathematical rigor to a *Second Data Science Course.* Students would be exposed to matrices as linear transformations. They would see the connections between graphical processing units (the CPUs of neural nets and deep learning). GPUs were first developed for computer graphics. They are designed to perform linear transformations literally in the blink of an eye. That is how motion is simulated. High speed computer graphics are mediated by high-speed linear transformations. The math of neural nets is also mediated by high-speed linear transformations.

Another term for linear transformation is matrix multiplication. As an exercise in rigor, students would learn how to decipher the kind of notation seen below. The notation is a formal recipe for performing a linear transformation, i.e., matrix multiplication.

If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is an $n \times p$ matrix,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

the *matrix product* $\mathbf{C} = \mathbf{AB}$ (denoted without multiplication signs or dots) is defined to be the $m \times p$ matrix

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{pmatrix}$$

such that

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj},$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, p$.

This article represents thoughts I have that amount to a very rough sketch of a *Second Class in Data Science* for high school. I believe these thoughts address the concerns of people who feel that data science does not offer enough mathematical rigor to be an alternative to the current high school math sequence. I hope the ideas in this article are found to be of value. As for me, I'm off to see the Wizard.