MATHEMATICS TEACHER
Learning & Teaching
PK 12

Mathematics Teacher: Learning and Teaching PK-12, is NCTM's newest journal that reflects the current practices of mathematics education, as well as maintains a knowledge base of practice and policy in looking at the future of the field. Content is aimed at preschool to 12th grade with peer-reviewed and invited articles. *MTLT* is published monthly.

ARTICLE TITLE:

Building a Digit Classifier with MNIST

AUTHOR NAMES:

Williams, Jedediyah

DIGITAL OBJECT IDENTIFIER: 10.5951/MTLT.2023.0187

VOLUME: 117

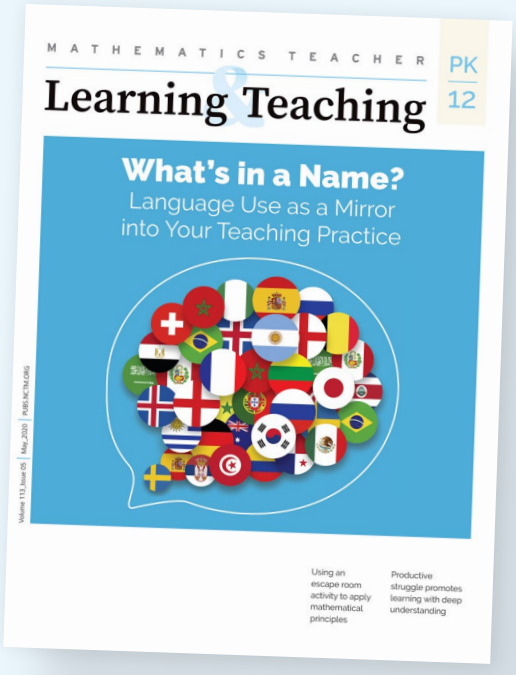
ISSUE NUMBER: 2

Mission Statement

The National Council of Teachers of Mathematics advocates for high-quality mathematics teaching and learning for each and every student.

Approved by the NCTM Board of Directors on July 15, 2017.

CONTACT: mtlt@nctm.org



NCTM NATIONAL COUNCIL OF TEACHERS OF MATHEMATICS

Building a Digit Classifier with MNIST

Students use feature engineering to build a classifier that can accurately recognize digits from images.

Jedediyah Williams

Email filters classify new messages as either spam or not spam based on word frequency, syntax, and metadata. Species of iris flowers can be classified by proportions of their petal lengths (Jones, 2021). Everything from colleges (O'Neil 2016, Chapter 3) to job applications (Wachter-Boettcher, 2018) and social media content (Krolik & Hill, 2019) is quantified and classified.

A *classifier* is an algorithm that maps input data into categories based on distinguishing characteristics, or *features*. Features can be raw data or attributes derived from that data. *Feature engineering* is the process of identifying or extracting features that can be used to capture relevant patterns in data.

The project I introduce below uses feature engineering with a dataset called the Modified National Institute of Standards and Technology dataset (MNIST). MNIST is a modified version of a NIST dataset (Deng, 2012) that contains 70,000 grayscale images of handwritten digits (Figure 1), and the goal with this dataset is to build a classifier that can accurately recognize digits from those images. Image processing algorithms for character recognition are widely applicable (Garris, 2019), from efficiently reading postal codes to apps like Photomath and self-driving cars that read street signs.

Each MNIST image is 28×28 pixels, and those 784 pixels take on values from 0 to 255, where 0

corresponds to black and 255 corresponds to white, with grays in between (Figure 2). Note that background pixels with no writing in them have values of 0.

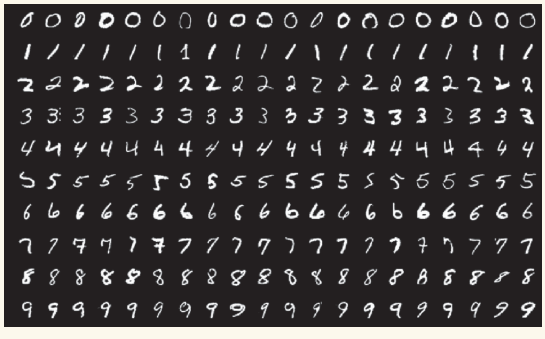
Each value corresponds to the brightness of one of the 784 pixels, from 0 to 255. The values here are aligned to their pixel location to simultaneously provide a sense of the image and the underlying data.

MNIST is a classic dataset in machine learning courses where students learn about algorithms that can identify and assess features. The project described here is not a machine learning project. Instead, students are invited to explore MNIST and create features and algorithms themselves. This is not simply a pedagogical choice aimed at developing understanding of fundamental concepts in modern mathematics applications. There are any number of scenarios, particularly when data are sparse or chaotic, where machine learning algorithms are not capable of providing feasible solutions (Birhane, 2021; Broussard, 2019; Raji et al., 2022) and expert human insight is required.

While the ultimate goal of this project is to algorithmically distinguish all ten digits from one another, let's explore a preliminary stage of distinguishing only between '0's and '1's. From there, the challenge is to incrementally extend solutions to distinguish more digits and improve the accuracy of the classifiers.

I introduce this digit classifier project to my students using a spreadsheet. However, this project is platform-independent and can be approached in a variety of ways and completed to varying extents. There is not a single solution, and many approaches will be successful. All of the data for this project, including MNIST images, sample spreadsheets, and example solutions, are available (link online).

Figure 1 Samples from MNIST of the First 20 Images of Each Digit

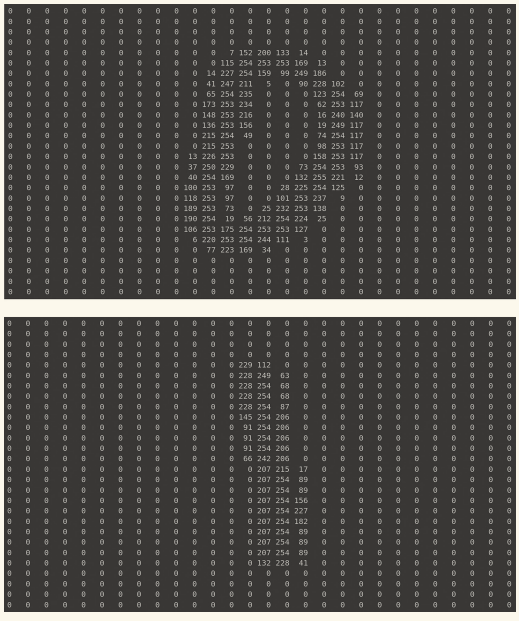


OVERVIEW OF PROJECT STAGES

This project has been used in Algebra 1, Precalculus, and Advanced Quantitative Reasoning classes in units on mathematical modeling and statistics. We follow three main stages, which reflect a common workflow in data modeling:

1. Explore and familiarize with the MNIST image data.
2. Feature engineering:
   a. Conceptualize features that could potentially distinguish '0's from '1's.
   b. Formalize those features into explicit formulas.

Figure 2 Sample of a '0' and a '1' from the MNIST Dataset



Note. The values here are aligned to their pixel location to give simultaneously a sense of the image and the underlying data.

Jedediyah Williams, jedediyah@gmail.com, teaches at Belmont High School in Massachusetts. Jed is concerned with ethical and technical pitfalls of data and AI technologies, including those being introduced into classrooms.

doi:10.5951/MTLT.2023.0187

c. Assess the distinguishing capabilities of those features.

3. Build a classifier using distinguishing features.

In Stage 1, each student creates their own copy of the data to work with, including at least fifty '0's and fifty '1's. Students explore the image data in multiple representations and begin to develop intuition about the distinguishing characteristics, in the context of data, between '0's and '1's. This stage of the project is excellent for introducing or practicing skills of exploratory data analysis.

In Stage 2, students are provided with opportunities to work both independently and collaboratively to find and build features that can distinguish between '0's and '1's. These features are any characteristic or combination of characteristics that can be leveraged to differentiate '0's and '1's based on the data that represent them.

In Stage 3, students use their features to build their own classifier, assess the success rate of that classifier, and validate their results on new data.

The modeling tasks involved with classifying digits are steeped in the Standards for Mathematical Practice (SMP) and the Standards for Mathematical Content (National Governor's Association Center for Best Practices & Council of Chief State School Officers [NGA Center & CCSSO], 2010). Throughout the project, students will apply their mathematical knowledge to build models (SMP 4). They will face choices and challenges regarding numerical platforms and which tools to utilize (SMP 5). In looking for features, students will search for patterns, not just in the structure of equation syntax, but in the structure of data (SMP 7), and they will need to reason abstractly and quantitatively (SMP 2) about how to formalize their conceptualizations of features that can capture the characteristics of a '0' or a '1' in the context of data. In determining which features should be leveraged to build a successful classifier, students will construct arguments and debate the efficacy of particular approaches by breaking problems down and analyzing individual features (SMP 3), while continuously reflecting and making sense of each step along the way toward their solutions (SMP 1).

This project builds upon middle school mathematics standards related to statistics as well as high school standards across multiple domains, including the following:

- HS.N-Q Reason quantitatively and use units to solve problems.
- HS.F-BF Building functions.
- HS.A-CED Create equations that describe numbers or relationships.
- HS.S-ID Interpreting categorical and quantitative data.
- HS.S-IC Making Inferences and justifying conclusions.

For students who opt to work with a numerical computing platform, they will also develop within the following:

- NS.A-REI Reasoning with equations and inequalities.
- HS.N-VM Vector and matrix quantities.

The process of building a model from data is often messy and iterative. Much of the familiarity garnered will come from failed attempts to determine useful features. Some approaches will be successful, but they will iteratively become even more successful. The examples below are drawn from student work and highlight some of those successes.

It is noteworthy that a full solution (for distinguishing '0' from '1') can be written in a single formula and takes only a few seconds to write, yet we spend three or more class periods working on this project. That time is filled with data exploration, mathematical reasoning, and communicating about algorithmic ideas.

Figure 3 Partial View of MNIST Data

	A	B	C	D	E	F	G	H	I	J	K	L
1	Digit	Pixel values --->										
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0

FAMILIARIZING WITH IMAGE DATA

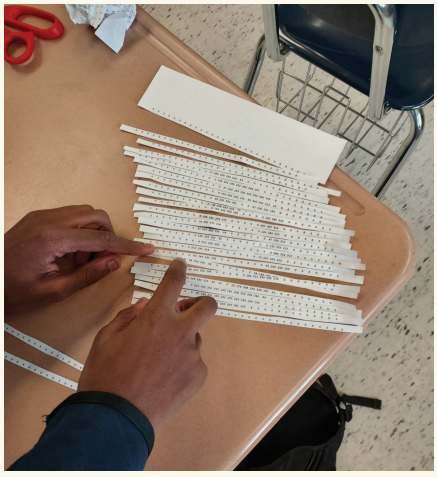
The MNIST image data are represented by arrays of pixel values where each row corresponds to a single image. Figure 3 shows a small portion of a spreadsheet of some of that data. Each row represents a single image. The first column is the digit label, and the next 784 columns are the pixel values of that digit. The first digit in row 2 is a '0', and its pixel values populate cells B2 through ADE2. The second digit in row 3 is also a '0', and its pixel values populate cells B3 through ADE3. The spreadsheet we start with contains 50 rows of '0's and 50 rows of '1's.

It can be confusing to look at an "image" as a row of 784 numbers. Figure 2 depicts digit samples where pixel values are in their corresponding locations. This hybrid view can support students' understanding and build connections between the image displayed on a screen and the data stored in a computer's memory. Both representations are useful when thinking about how to distinguish digits from one another.

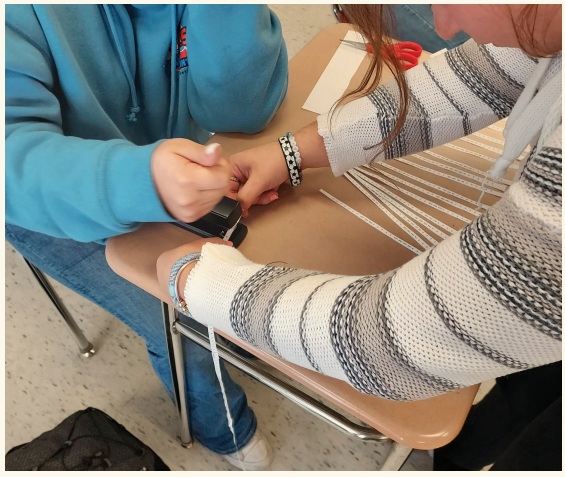
To further reinforce the connections between images and the spreadsheet of pixel values, we print examples in the form of Figure 2, so that students

Figure 4 (a) Student Cuts Out 28 Rows of Pixel Values Representing the Image of a Zero, (b) Students Work to Join 28 Rows into a Single Row of 784 Values, and (c) Students Post Their "Images" on a Whiteboard

(a)



(b)



(c)

can physically cut out the 28 rows of 28-pixel values and transform them into one long row of 784 values. Figure 4 depicts various stages of this exercise. The result is a physical representation of our spreadsheet where each row corresponds to a single digit.

Students explore the data in the spreadsheet, or on the board, and look for patterns. We do this work together with frequent check-ins to share ideas. Familiarizing with data together offers opportunities for students to develop shared conceptions, building collaboratively, while identifying landmarks in the landscape of the data. These conceptions will develop into the features that can be used to classify digits.

The following is an example interaction from a group when viewing the data early on:

Student 1: I forgot that some people put "hats" on their '1's… and add a bottom piece.
Student 2: It's mostly zeros!
Student 3: So most of the image has no data.
Dr. W: Could you say more about there being "no data"?
Student 3: I mean, like, it has data, but the value is nothing.
Dr. W: Totally! Wait, what does that mean, though?
Student 4: The zeros are the background where there's no writing. You only see [non-zero] numbers where there is writing.

…

Student 2: When you slide to the right in the spreadsheet, the numbers are grouped together. Like there are chunks of numbers. It has lots of zeros and then not zeros and then zeros.
Student 1: Because it's like reading a book. It goes line by line across the page, so you see where there was a number in the picture, and then there wasn't a number.
Student 5: When you slide back and forth in the spreadsheet you can kind of see the line where the '0's stop and the '1's start. You can tell above here it's '0's and below that it's '1's.
Student 1: If you look in the middle of the data, around the column MO, you can see where the '1's go straight up and down and the '0's go kind of around the '1's.

At the beginning stages, students will be familiarizing simply with the format of the data. As they get to know the data, they will begin to uncover patterns, particularly ones that will be useful for distinguishing digits. Student 1 above was noticing a distinction between '0's and '1's that would later develop into Feature 2, realizing that '1's tend to go through central pixels, while '0's tend to go around them.

FEATURE ENGINEERING

After familiarizing ourselves with our datasets of '0's and '1's, we begin to conceptualize, formalize, and analyze features. I ask, "What are the distinguishing features of '0's and '1's?" I visibly record students' ideas. Some of those students' conceptualizations include the following:

1. If you add up all the pixels for '0's it should be more than for '1's.
2. '0's are "hollow" so the middle pixels should be zeros.
3. '0's are wider, so they have more [non-zero] pixels outside of the middle [column].
4. Across the middle row of a '0', it goes zeros, not zeros, zeros, not zeros, zeros.
5. '1's should have more pixels that are zero.
6. If you add up the pixels in just the middle row, it should be less for '1' than for '0'.
7. The ratio of the sum of rows 5–8 to the sum of rows 11–15 should be lower for '0'.
8. Straight down the middle [column], '1's should have more non-zero pixels.

Some of the above features are easier to formalize than others. Features 1 and 2 are developed in the next sections.

Feature 1: Sum of All Pixels

Multiple students proposed Feature 1. One student elaborated that their thinking behind this feature was that "writing a '0' covers more pixels than a '1' does, and since the pixels with no writing are [valued at] zero, adding up all of the pixels in an image should give a higher value for '0's than for '1's". This feature is nicely supported in a spreadsheet, as we can write a formula to sum the pixels of the first image in row 2:

= SUM(B2:ADE2)

This formula calculates the sum of all values in cells from B2 to ADE2, which represent the 784 pixels of the first image in row 2. Within the spreadsheet, this formula can be placed in an empty column in row 2, and

the remaining rows can be filled to calculate this feature for each image in each row.

Figure 5 is a visualization of the values of Feature 1 for the one hundred images in our spreadsheet. I ask students to interpret and share descriptions of what they see in the visualization. I then ask, "What is a value for this feature that would make you confident the image is a '0'? What about '1'? What value would make you unsure?" A natural question develops as to where to place the threshold. "What feature value should be chosen above which we classify an image as a '0'?" Just one feature can facilitate an enormous amount of discussion with descriptive statistics.

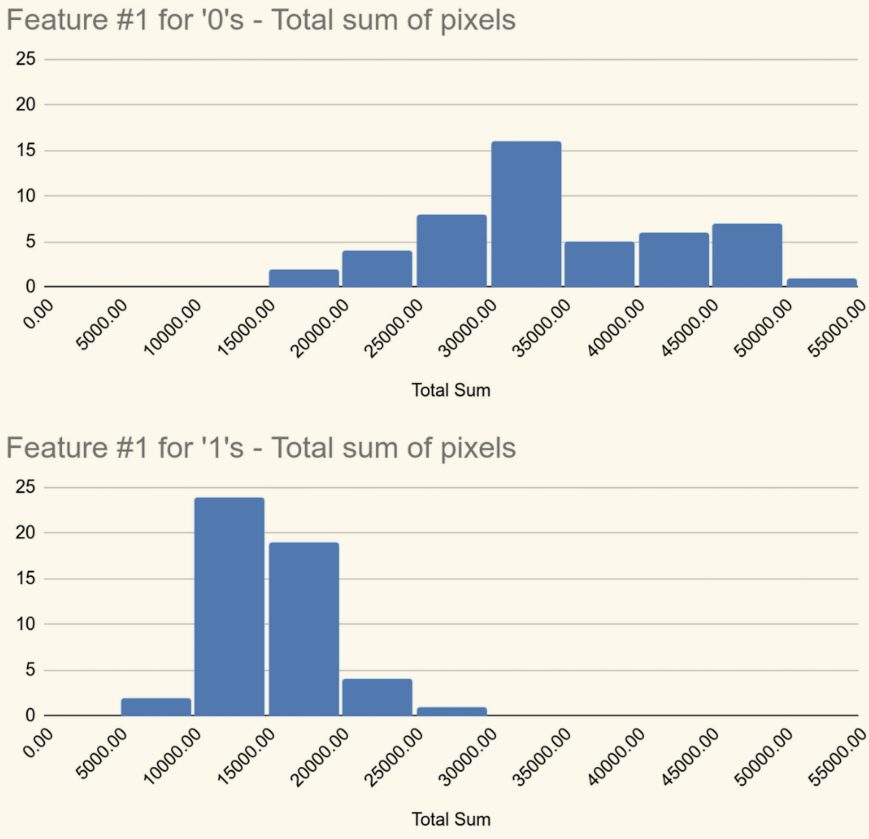
With this particular subset of data, my students tend to identify 20,000 and 25,000 as interesting values and debate where to place the threshold, typically deciding on a value somewhere in the middle. Inspecting the data directly or using a scatter plot, they discover the highest sum for a '1', excluding an outlier, is 22,590. Interestingly, without being explicitly prompted, they seem to understand that there is an underlying optimization problem where we are attempting to choose a threshold sum that splits the data such that the number of correctly classified digits is maximized.

Feature 2: Sum of Middle Pixels

The student thinking behind Feature 2 was that "the '1's should pass through the middle pixels, but '0's are hollow so they should go around the middle pixels." The concept behind this feature has been formalized by students in a variety of ways:

1. Identify a single pixel in the middle of the image data that appears to have a value of zero for '0's and a non-zero value for '1's (e.g., the pixel in column NQ).
2. Choose a small group of pixels that span multiple columns and rows in the middle of the image (e.g., columns MN, MO, NP, and NQ). These four pixels form a square roughly in the middle of the 28×28 image (rows 12 and 13, columns 15 and 16).

Figure 5 Histograms of Feature 1, Sum of Pixel Values



3. Choose a set of pixels that span multiple columns in the same row (e.g., columns NN, NO, NP, NQ). The approach of spanning more than one column is an attempt at a more robust feature than looking at just a single pixel.

All of the successful approaches above are conceptions that were engineered into features by inspecting the data for regions of pixels that seemed to follow the idea that '0's tend to go around central pixels while '1's tend to go through them.

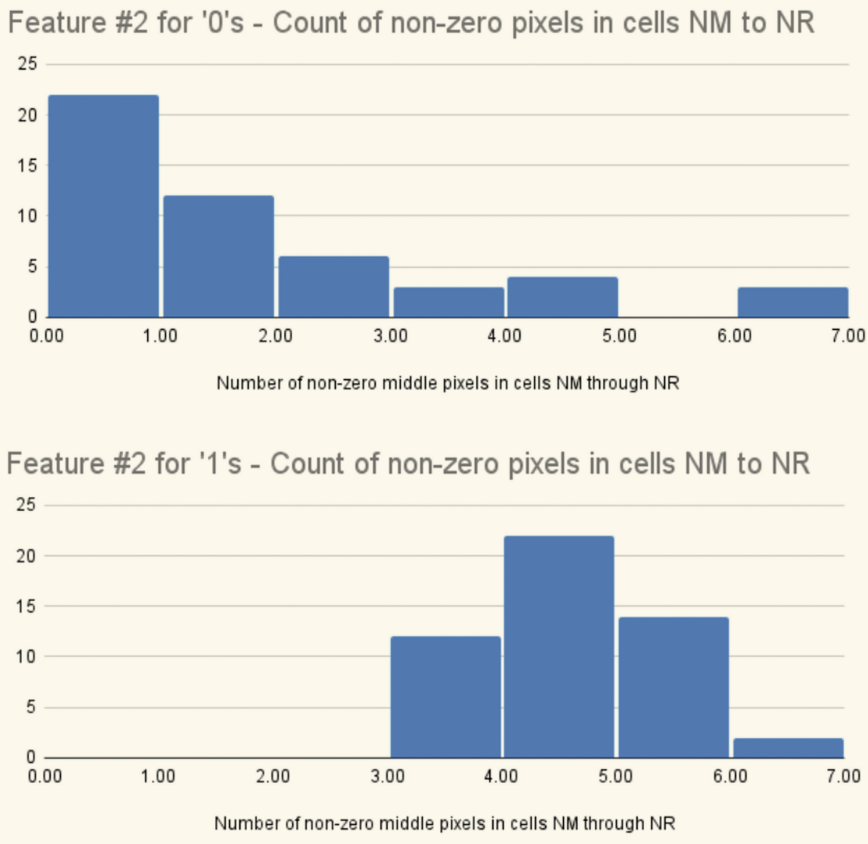
Here, we will consider a version of this feature with six adjacent pixels across a middle row of the image and count how many of those six pixels have values greater than zero. We suspect that more of these pixels will have non-zero values for '1's than for '0's. This feature can be calculated for the first image in row 2 with the following formula:

= COUNTIF(NM2:NR2,">0")

This formula counts how many of the cells in the range from NM2 to NR2 (six central pixels) have values greater than zero. As we did with our formula for Feature 1, this formula can be entered into an empty column in row 2 and then filled for all rows to calculate the feature values for every image.

Figure 6 contains histograms of this feature for all fifty '0's and all fifty '1's in the training set. The centers of the distributions are noticeably separated, but there is some overlap. I again ask students to interpret the feature visualizations and to consider values that would lead to confident classifications of '0' or '1', or for inconclusive values. One student observed that "if a digit has more than three, or three, pixels in the middle that aren't zero, if you call that a '1' then you're going to be right most of the time." Another student added that "if the number of non-zero pixels is less than three, then it is definitely a '0'." This type of feature, which is somewhat but not perfectly distinguishing, is an excellent candidate to be combined with other features.

Figure 6 Histograms of Feature 2: Number of Non-Zero Pixels in Six Central Cells



When you move from classifying only '0's and '1's to classifying a larger subset of the digits, some features will be useful for distinguishing between pairs or small groups of digits, and some will be useful for distinguishing one digit amongst them all. You might engineer features that are combinations of other features; for example, the ratio of the sum of the top half of the pixels to the sum of the bottom half of the pixels. Wonderful opportunities for collaboration arise when students share their ideas and create new features together to incrementally improve their features and classifiers.

BUILDING A CLASSIFIER

The features above capture some sense of "zero-ness" and "one-ness," which we can leverage to make a prediction about the category of a given digit image. In this section, we will combine our features into a single classifier and then validate that classifier on new data. When starting this stage with students, I ask for volunteers to share one or two features and then demonstrate how to build a classifier using the sample features.

Combining Features Into a Classifier

Feature 1 is the sum of all pixels in an image (Figure 5), and Feature 2 is the count of pixels, from six central pixels, that are non-zero (Figure 6). Each of these features is fairly distinguishing, but combining features will provide even better results than classifying with any one feature alone.

Our classifier is based on student observations of our two features:

1. For Feature 1, the highest value for any '1' is 22,590, excluding an outlier.
2. For Feature 2, all '1's have a value of 3 or above.

We can write both of these observations as inequalities and write a condition based on those inequalities: given an image, if the sum of its pixels is less than or equal to 22,590 *and* the number of non-zero pixels from its six central pixels is greater or equal to 3, then classify the image as a '1'; otherwise, classify it as a '0'. We might write this into a formula as follows:

```
= IF(AND(SUM(B2:ADE2) <= 22590,
  COUNTIF(NM2:NR2) >= 3),
  1, 0)
```

For our data subset of 100 images, this classifier correctly classifies 99 of the images.

Model Validation

Prediction accuracy of 99% is exciting, but this measure of accuracy is based on the data we used to build our classifier and is therefore susceptible to overfitting, a significant problem in data modeling (Kapoor & Narayanan, 2022). We should not infer too much from how well our classifier does on the data we used to build it (James et al., 2013). Instead, we should concern ourselves with how well our classifier performs on data it has never before seen.

Data splitting—where data is separated into a training set for building models and a testing set for model validation—is a powerful tool. We worked with only 100 images in our training set, and MNIST has tens of thousands of images. For validation, we can use the next 100 images of '0's and '1's. Applying our classifier to those images, we find a remarkable success rate of 100%.

REFLECTION

When building data-based models that could be used in real applications, it is important to reflect on the potential consequences of their use. What are the possible applications of digit recognition? What are possible consequences of misclassifying a digit in those applications? What happens if, for example, a postal code is misread on a shipping address, or a math app misreads a homework problem, or a self-driving car mistakes a 35 MPH sign for an 85 MPH sign? Such considerations are rarely explicitly connected to mathematics content (Akgun & Greenhow, 2022), but they are deeply connected with mathematics and data-based applications and are therefore a critical component of education in data sciences (Lee et al., 2021).

There is, unfortunately, no shortage of examples of harmful consequences of math-based technologies (D'Ignazio & Klein, 2023; Eubanks, 2019; Noble, 2018; O'Neil, 2016), but when we give students real problems to work on, we are providing opportunities to reflect on the real impact they can have (Williams, 2021). In the words of Ruha Benjamin, "Yes, train these young people to get these skills, but integrate into that not only the technical capacity but the critical capacity to question what they're doing and what's happening" (Benjamin, 2019, 58:48).

CONCLUSIONS

I introduced the MNIST dataset with a digit classification project that involves identifying features in image data to distinguish between images of '0's and '1's. Then, I demonstrated an example classifier, which achieved high accuracy, but this is only the beginning! It is left to you (and maybe your students) to find features that can distinguish even more digits. As more digits are included in your modeling, it becomes increasingly challenging to distinguish them manually, and you might even resort to machine learning.

When building mathematical models with data, model validation methods like data splitting are important for verifying that our models are working correctly beyond the data on which they were built. Even then, if we work with data that represents cultural conventions, it is likely that our models will require updating or rebuilding to reflect the changes not represented in the original data. Streaming entertainment like music and movies are constantly being updated to reflect the rotation of new and old content and may adjust recommendations or offer content corresponding to societal changes.

Reflecting on the possible applications of our models provides opportunities to think about the utility of mathematics as well as the consequences that can result from doing mathematics.

REFERENCES

Akgun, S., & Greenhow, C. (2022). Artificial intelligence in education: Addressing ethical challenges in K-12 settings. *AI Ethics, 2*(3), 431–440. https://doi.org/10.1007/s43681-021-00096-7

Benjamin, R. (2019, October 29). *The new Jim code? Race, carceral technoscience, and liberatory imagination*. Othering & Belonging Institute. https://www.youtube.com/watch?t=3510&v=JahO1-saibU&feature=youtu.be

Birhane, A. (2021). The impossibility of automating ambiguity. *Artificial Life, 27*(1), 44–61. https://doi.org/10.1162/artl_a_00336

Broussard, M. (2019). *Artificial unintelligence: How computers misunderstand the world*. MIT Press.

Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research [Best of the Web]. *IEEE Signal Processing Magazine, 29*(6), 141–142. https://doi.org/10.1109/MSP.2012.2211477

D'Ignazio, C., & Klein, L. F. (2023). *Data feminism*. MIT Press.

Eubanks, V. (2019). *Automating inequality: How high-tech tools profile, police, and punish the poor*. Picador.

Garris, M. (2019, March 26). *Michael Garris: The story of the MNIST dataset*. YouTube. https://www.youtube.com/watch?v=oKzNUGz21JM

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. Springer New York.

Jones, J. (2021). Integrating machine learning in secondary geometry. *Mathematics Teacher: Learning and Teaching PK-12, 114*(4), 325–329. https://doi.org/10.5951/MTLT.2020.0187

Kapoor, S., & Narayanan, A. (2022). *Leakage and the reproducibility crisis in ML-based science*. https://reproducible.cs.princeton.edu/#rep-failures

Krolik, A., & Hill, K. (2019, October 11). How photos of your kids are powering surveillance technology. *The New York Times*. https://www.nytimes.com/interactive/2019/10/11/technology/flickr-facial-recognition.html

Lee, V., Hoda Wilkerson, M., & Lanouette, K. (2021). A call for a humanistic stance toward K–12 data science education. *Educational Research, 50*(9), 585–687. https://doi.org/10.3102/0013189X211048810

National Governors Association Center for Best Practices & Council of Chief State School Officers. (2010). *Common core state standards for mathematics*. http://corestandards.org

Noble, S. U. (2018). *Algorithms of oppression: How search engines reinforce racism*. NYU Press.

O'Neil, C. (2016). *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown.

Raji, I. D., Kumar, I. E., Horowitz, A., & Selbst, A. (2022, June). The fallacy of AI functionality. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (pp. 959–972). https://doi.org/10.1145/3531146.3533158

Wachter-Boettcher, S. (2018). *Technically wrong: Sexist apps, biased algorithms, and other threats of toxic tech*. Norton.

Williams, J. (2021). *Data science and consequences in mathematical modeling.* https://jedediyah.github.io/nctm2021/

