Playing with the Data: What Legal Scholars Should Learn About Machine Learning

David Lehr[†*] *& Paul Ohm*[**]

TABLE OF CONTENTS

* Research Fellow, Georgetown University Law Center; J.D. Candidate, Yale Law School, 2020.

** Professor of Law, Georgetown University Law Center. We are extremely grateful for helpful comments from Kiel Brennan-Marquez, Jonathan Frankle, Clare Garvie, Micha Gorelick, James Grimmelmann, Harrison Rudolph, Andrew Selbst, and Michael Skirpan. We also thank participants at the University of Pennsylvania Law School's Fifth Annual Roundtable on Computer Science and Law for their feedback. Any errors are our own. This work has been generously supported by the AXA Award on Big Data, Privacy, and Discrimination, from the AXA Research Fund.

INTRODUCTION

Legal scholars have begun to focus intently on machine learning — the name for a large family of techniques used for sophisticated new forms of data analysis that are becoming key tools of prediction and decision-making. We think this burgeoning scholarship has tended to treat machine learning too much as a monolith and an abstraction, largely ignoring some of its most consequential stages. As a result, many potential harms and benefits of automated decision-making have not yet been articulated, and policy solutions for addressing those impacts remain underdeveloped.

To fill these gaps in legal scholarship, in this Article we provide a rich breakdown of the process of machine learning. We divide this process roughly into eight steps: problem definition, data collection, data cleaning, summary statistics review, data partitioning, model selection, model training, and model deployment. Far from a straight linear path, most machine learning dances back and forth across these steps, whirling through successive passes of model building and refinement.

Simplifying this mapping, we contend that legal scholars should think of machine learning as consisting of two distinct workflows: "playing with the data," which comprises the first seven steps of our breakdown, and "the running model," which describes a machine-learning algorithm deployed and making decisions in the real world. Our core claim is that almost all of the significant legal scholarship to date has focused on the implications of the running model — the predictive policing algorithm directing the deployment of officers,[1] the face recognition system identifying suspects,[2] or the autonomous automobile navigating a turn[3] — and has neglected most of the possibilities and pitfalls of playing with the data. Particularly in the

[1] *See, e.g.*, Andrew Guthrie Ferguson, *Big Data and Predictive Reasonable Suspicion*, 163 U. PA. L. REV. 327 (2015); Elizabeth E. Joh, *The New Surveillance Discretion: Automated Suspicion, Big Data, and Policing*, 10 HARV. L. & POL'Y REV. 15 (2016); Michael L. Rich, *Machine Learning, Automated Suspicion Algorithms, and the Fourth Amendment*, 164 U. PA. L. REV. 871 (2016).

[2] *See, e.g.*, CLARE GARVIE ET AL., THE PERPETUAL LINE-UP: UNREGULATED POLICE FACE RECOGNITION IN AMERICA (2016); Douglas A. Fretty, *Face-Recognition Surveillance: A Moment of Truth for Fourth Amendment Rights in Public Places*, 16 VA. J.L. & TECH. 430 (2011).

[3] *See, e.g.*, Bryant Walker Smith, *Proximity-Driven Liability*, 102 GEO. L.J. 1777 (2014) (explaining the increase of vehicle automation in the next decade); Harry Surden & Mary-Anne Williams, *Technological Opacity, Predictability, and Self-Driving Cars*, 38 CARDOZO L. REV. 121 (2016) (explaining self-driving cars that are controlled by computers).

fields of criminal justice and criminal procedure, machine-learning systems are seen as inscrutable black boxes by scholars focused on the Fourth Amendment.

Black boxes also sit at the heart of important work by Frank Pasquale and Danielle Citron who, together[4] and separately,[5] have authored important articles on the rise of automated decision-making in many contexts, such as the delivery of government benefits and credit scoring. As important as we find this work, we think it can be strengthened by giving more attention to machine learning's playing-with-the-data stages.

A few notable and important articles pay some attention to playing with the data. Most significantly, an article by Solon Barocas and Andrew Selbst on bias in employment decision-making focuses astutely on problems that creep in during data collection.[6] The article does tend to neglect many of the other stages of playing with the data, but we recognize this as a side effect of the topic they are studying — the problem of bias. Bias seems to emerge in data-related stages most directly and perhaps even exclusively.

By not paying attention to other stages of machine learning, scholars have overlooked the fact that the two workflows of machine learning give rise to very different issues. The potential harms and benefits that can creep in while playing with the data differ from those of the running model. For example, Barocas and Selbst documented the "garbage in, garbage out" problem, which can make machine-learning models discriminatory,[7] but, from the vantage point of the running model, this "garbage" is a static, unavoidable feature of the data. Only one who is attentive to the many ways in which data can be selected and shaped — say, during data cleaning or model training — will characterize fully the source of the stink. Similarly, a benefit of choosing certain machine-learning algorithms is the ability to place weight on particular types of errors over others — for example, to

[4] *See, e.g.,* Danielle Keats Citron & Frank Pasquale, *The Scored Society: Due Process for Automated Predictions*, 89 WASH. L. REV. 1 (2014) (defining black boxes, which convert inputs to outputs without revealing how they do so).

[5] *See, e.g.*, FRANK PASQUALE, THE BLACK BOX SOCIETY: THE SECRET ALGORITHMS THAT CONTROL MONEY AND INFORMATION (2015); Danielle Keats Citron, *Technological Due Process*, 85 WASH. U. L. REV. 1249 (2008).

[6] *See generally* Solon Barocas & Andrew D. Selbst, *Big Data's Disparate Impact*, 104 CALIF. L. REV. 671, 677-87 (2016) (discussing how data mining may reflect discrimination of society).

[7] *Id.* at 680-87.

favor false negatives over false positives in criminal justice contexts — but this choice is one that must be made when playing with the data.[8]

Another reason legal scholars in particular need to focus on playing with the data is that combatting harms at the running-model stage is often too little too late. Because playing with the data occurs earlier in time and entails much more human involvement than the running model, this phase provides more opportunities and behavioral levers for policy prescriptions. As many have documented, a running model is often viewed as an inscrutable black box,[9] but there are opportunities for auditing (record-keeping requirements, keystroke loggers, etc.) and mandated interpretability during playing with the data. We can ban certain approaches — say, deep learning techniques like convolutional neural nets, if our concern is inscrutability — during playing with the data, but, with a running model, all we can do is rue the choice that has already been made. These possibilities may be neither necessary nor sufficient to address the potential harms of machine learning, but they are likely to be missed by those with a single-minded focus on the running model.

Greater attention to playing with the data can also advance contemporary debates about machine learning. Regulation skeptics and industry members often rely on descriptions of machine learning as "more art than science,"[10] but we think this inappropriately assumes that black-box algorithms have black-box workflows; as we show, the steps of playing with the data are actually quite articulable. Additionally, many commentators have argued that we must preserve a "human in the loop" of machine learning,[11] but most of them are referring to the running model as the relevant loop. We think there are

---

[8] *See infra* Parts II.G–H.

[9] *See, e.g.*, PASQUALE, *supra* note 5; Leo Breiman, *Statistical Modeling: The Two Cultures*, 16 STAT. SCI. 199 (2001) [hereinafter Breiman, *Statistical Modeling*].

[10] Abhishek Mehta & Eliud Polanco, *Breaking Bad Data & Solving for AML*, GLOBAL RISK INST. (June 6, 2017), http://globalriskinstitute.org/wp-content/uploads/2017/06/Breaking-Bad-Data-Solving-for-AML-FINAL-UPDATE.pdf [https://perma.cc/M9DS-3F6G] ("Tuning AML detection models to achieve optimum precision is more art than science."); IBM ML Hub, *The 3 Kinds of Context: Machine Learning and the Art of the Frame*, INSIDE MACHINE LEARNING (Apr. 26, 2017), https://medium.com/inside-machine-learning/the-3-kinds-of-context-3c4065e26749 [https://perma.cc/G9JW-XUSK] ("As much as thinking about these three context has helped us, it's also reinforced the fact that machine learning is often more art than science.").

[11] Adrian Bridgwater, *Machine Learning Needs a Human-in-the-Loop*, FORBES (Mar. 7, 2016, 1:00 PM), https://www.forbes.com/sites/adrianbridgwater/2016/03/07/machine-learning-needs-a-human-in-the-loop [https://perma.cc/MZV2-DP4S]; *see* Jatinder Singh et al., *Responsibility & Machine Learning: Part of a Process*, (Oct. 27, 2016), https://ssrn.com/abstract=2860048 [https://perma.cc/9ZAE-ECHD].

different — perhaps more imperative — reasons to maintain humans in the underappreciated playing-with-the-data loop as well.

We are not saying that scholars ought to neglect the running model. The best assessments of the promises, perils, and prescriptions for automated decision-making will consider both phases of machine learning. However, widening the view to encompass earlier stages will be crucial for solving some seemingly intractable problems of our increasingly automated world.

Our Article proceeds in three parts. Part I surveys the burgeoning literature at the intersection of law and machine learning, highlighting the relatively limited conception of machine learning these articles seem to adopt. Part II provides a detailed explication of the stages of machine learning. Here, we highlight aspects of machine learning that have yet to figure prominently in legal scholarship. Finally, Part III builds on this primer, demonstrating how a more complete understanding of machine learning will help us diagnose harms we have not yet recognized, as well as benefits and prescriptions we have not yet tried to deploy.

I. THE STATE OF THE SCHOLARSHIP

A. Machine Learning, Automated Suspicion Algorithms, and the Fourth Amendment

Perhaps the largest swath of legal scholarship evaluating machine learning, measured both by authors and scholarly works, focuses on the impact of machine learning on policing and criminal procedure. Much of this writing focuses on what is known as "predictive policing." Scholars focus on the use by police departments of systems that predict crime "hot spots" — where and when crime is most likely to occur — and which individuals may commit crimes.[12] Often, these articles assess such systems as a matter of constitutional law. Does algorithmic decision-making of this sort comply with the Fourth Amendment's probable cause and particularity requirements?

A fine and representative example is 2016's *Machine Learning, Automated Suspicion Algorithms, and the Fourth Amendment* by the late

[12] *See, e.g.*, Ferguson, *supra* note 1; Andrew Guthrie Ferguson, *Policing Predictive Policing*, 94 WASH. U. L. REV. 1113 (forthcoming 2017); Andrew Guthrie Ferguson, *Predictive Policing and Reasonable Suspicion*, 62 EMORY L.J. 259 (2012); Elizabeth E. Joh, *Policing by the Numbers: Big Data and the Fourth Amendment*, 89 WASH. L. REV. 35 (2014); Joh, *supra* note 1; Rich, *supra* note 1.

Michael Rich.[13] Rich focuses on the use of machine learning "to predict individual criminality."[14] He argues that these techniques intrude into what had "remained the sole province of human actors,"[15] before now not implicated by the rise of technologies such as fingerprint and DNA identification.

Rich's thesis is that automated suspicion algorithms are insufficient to generate individualized suspicion on their own, and that their incorporation into human totality-of-the-circumstances analyses is feasible but reveals inadequacies in Fourth Amendment doctrine. Rich argues that an officer ought not be allowed to base arrest or search decisions entirely on a machine-learning system's predictions alone, because algorithms, unlike humans, cannot engage in a totality-of-the-circumstances analysis.[16]

With this established, the question then becomes how algorithms could be incorporated into a human-based totality-of-the-circumstances analysis. Rich attempts to analogize algorithmic predictions to police profiles and informants, but finds these analogies to be poor; important dissimilarities between how humans and algorithms process information make the latter unlikely to be afforded the same evidentiary weight as the former.[17] Rich finds a more tenable analogy between automated suspicion algorithms and drug dogs, a type of organic black box. But this analogy does not completely bless the use of algorithms in policing. Rich contends that there are inadequacies in the Court's current doctrine on drug dogs, particularly regarding if and how field performance is analyzed, as well as if and how prior odds form part of this analysis; these inadequacies will likely be amplified when the dogs become machines.[18]

There is much to recommend in this carefully reasoned article. The analogies Rich draws are compelling and richly elaborated. As an early work in this vein, it has attracted well-deserved praise and will end up part of the early canon. But even it suffers in the light of our playing-with-the-data thesis. This article is focused almost exclusively on the running model. Although it contemplates that errors can result from mistaken programming or bad data, these points escape detailed treatment.

---

[13] *See* Rich, *supra* note 1.

[14] *Id.* at 875.

[15] *Id.* at 877.

[16] *Id.* at 893-901.

[17] *Id.* at 901-11.

[18] *See id.* at 911-23.

Perhaps a circumstantial way to measure the extent to which an author fails to consider playing with the data is what we call the "subject test." When describing machine-learning algorithms, what nouns does the author choose to describe the force behind the decision-making? In Rich's articles, these nouns are almost always inanimate, such as "system"[19] or "model"[20] or "algorithm." We would prefer articles that referenced human programmers, statisticians, or data scientists, as at least indirect proof that the authors are conceptualizing these processes as human processes.

We are not quibbling about merely stylistic writing choices here. Rich relies directly on the inhuman nature of machine learning in some of his most effective passages. He asks whether courts are likely to compare the outputs of machine learning to information obtained from other officers or informants.[21] In both cases, machine-learning algorithms fall short of their human analogues precisely because they supposedly differ in important and fundamental ways from humans. Informants can update their suspicion when they learn new information, while a "database cannot contain all the facts that are relevant in every case."[22] Rich acknowledges that our faith in human officers to take in additional information is probably overly optimistic, hampered as humans are by cognitive biases and limited by legal constraints on what types of facts we should consider.[23] Still, he thinks the fiction of a human with the potential to learn is vital to the Constitution's requirement of individualized suspicion and is lacking with an algorithm.

This is an unusually constrained image of machine learning. It supposes not only that a machine-learning algorithm is an inhuman machine not controlled by human data scientists but also that these systems are static and incapable of bringing in new forms of inculpatory information. For example, Rich points with approval to an officer who is told by an algorithm about a suspected drug dealer on a corner who does not flee when the officer approaches and who appears to be handing out leaflets for a church event.[24] It is unclear why the algorithm could not be retrained to incorporate both of these obviously important facts.

---

[19] *Id.* at 875.

[20] *Id.* at 881.

[21] *See id.* at 901-11.

[22] *Id.* at 898.

[23] *See id.* at 899-900.

[24] *Id.* at 898.

This view of machine learning leads Rich to find the best possibility in comparing machine-learning algorithms to drug-sniffing dogs. Both machine-learning algorithms and drug-sniffing dogs "spit out predictions, but whose inner workings are unknown and perhaps incomprehensible to humans."[25] Interestingly, these features often differentiate machine learning from other algorithm-based investigative techniques such as "DNA matching and blood-alcohol-level testing."[26]

Once again, we find this analysis to focus too much on a "naturalized" view of machine learning, one that neglects the intricate processes of machine learning. The inner workings of a random forest algorithm are not nearly as inscrutable as a canine's olfactory system. More importantly, the random forest can easily be redesigned, while the dog's nose cannot.

Rich's article is not unusual in the amount to which it treats machine learning as a fully formed black box. In fact, we find that Fourth Amendment scholarship often falls prey to the misimpression that machine-learning systems spring into being fully formed and are impenetrable black boxes. This type of error is also present in some work of Andrew Ferguson, who has written several sophisticated articles about the intersection of Fourth Amendment law and machine learning. In *Big Data and Predictive Reasonable Suspicion*, for instance, his analyses are premised on the proliferation of "big data" policing techniques, but the reader is left to find out in a footnote that this phrase encompasses machine learning, and there is no further explanation of machine learning or how it factors into the policing practices he considers.[27] The exact same error can be found in Elizabeth Joh's *The New Surveillance Discretion*, which points to the same description from the same author as Ferguson's article to constitute the entire discussion of machine learning.[28]

Why might Fourth Amendment scholars fall prey to this more than scholars in other disciplines? We are not sure, but there are two possibilities. First, none of the Fourth Amendment scholars we have listed are, formally, technically trained. This is not to say that they are making errors about what they have reported. But perhaps the distance between these scholars and their subject lends itself to the playing-with-the-data error. To them, these technologies might appear

[25] *Id.* at 911.

[26] *Id.* at 912.

[27] Ferguson, *supra* note 1, at 350 n.122.

[28] Joh, *supra* note 1, at 16 n.6.

to be black boxes more than they would to those with more direct training.

Second, the path of legal scholarship might in interesting ways mirror the division of labor in the part of the legal profession being studied. It is highly unlikely that any police department in the country, save perhaps for the biggest departments such as the LAPD or NYPD or FBI, is developing machine-learning systems in-house. Instead, the best-documented examples involve private vendors who develop solutions that they then market to police departments in arms-length contract negotiations.[29] These agreements are shrouded both by the terms of the contracts themselves as well as a background constellation of intellectual property law, particularly trade secrecy.[30]

In other words, because police departments may buy running models off the shelf as inviolate black boxes, legal scholars may simply be inheriting from police officials the view that these running models cannot be scrutinized. Their lids are fixed tight.

B. The Scored Society: Due Process for Automated Predictions

Another article worthy of mention is Danielle Citron and Frank Pasquale's *The Scored Society: Due Process for Automated Predictions*.[31] Published in 2014, this is a notably early attempt to voice concerns about automated decision-making. Particularly, Citron and Pasquale worry that individuals are increasingly being scored by algorithms in consequential ways without any "technological due process."[32]

The authors focus mostly on financial scoring — the algorithmic generation of credit and other risk scores that can bear heavily on individuals' access to financial resources, employment, housing, and more. These scoring systems can rely on standard, logic-based algorithms or on machine-learning algorithms. According to the authors, scoring systems, artificially intelligent or not, are opaque;

---

[29] *See* Aaron Shapiro, *Reform Predictive Policing*, 541 NATURE 458, 459 (2017) ("HunchLab applies machine-learning and artificial-intelligence algorithms to predict the spread of crime types."); *What PredPol Is and What PredPol Is NOT*, PREDPOL BLOG (Nov. 19, 2015), http://www.predpol.com/whatispredpol [https://perma.cc/F5G5-FYW9] ("Predictive Policing's forecasting technology includes high-level mathematics, machine learning, and proven theories of crime behavior, that take a forward-looking approach to crime prevention.").

[30] *See* Rebecca Wexler, *Life, Liberty, and Trade Secrets: Intellectual Property in the Criminal Justice System*, 70 STAN. L. REV. (forthcoming 2018), https://ssrn.com/abstract=2920883 [https://perma.cc/A53M-PFST].

[31] Citron & Pasquale, *supra* note 4.

[32] *Id.* at 8.

inaccurate, or arbitrary; and potentially discriminatory.[33] Because of these harms, the authors call for, among other solutions, public transparency of scoring processes and calculations, as well as licensing and auditing requirements for scoring systems.[34] It is only with these safeguards, they argue, that we can prevent a society where individuals' livelihoods are put in jeopardy by unaccountable algorithms.

This identification of problems in an increasingly automated society is an undoubtedly important contribution to legal scholarship. Unlike the Fourth Amendment scholars, who put forward an often unrealistic, yet simultaneously unexplained, conception of machine learning, Citron and Pasquale do not. Also, Citron and Pasquale do not fall into the trap of premising their analyses on fanciful algorithms, like a system, detached from any humans, that autonomously decides whom to search or arrest.

Nevertheless, we think Citron and Pasquale's analysis could be augmented with more attention to playing with the data. In particular, we highlight two improvements to this work that could have been made had the authors said more about the playing-with-the-data stages of machine learning.

First, Citron and Pasquale could have provided more detailed and, thus, useful prescriptions for achieving technological due process. In attempting to provide prescriptions simultaneously applicable to both conventional and artificially intelligent scoring systems, Citron and Pasquale drafted prescriptions that are too vague to be put directly into practice for the latter. Take, for instance, their call for public inspection of "all processes (whether driven by AI or other computing)" of score calculations.[35] Legislators seeking to curtail the risks of purely conventional scoring systems would likely find this language sufficiently informative to guide their efforts. There are only so many processes in a conventional scoring system, and those processes can be fairly intuitively understood, even by lay individuals. So, calls for revealing "all processes" of conventional systems could be easily translated into legislation mandating the disclosure of particular technical aspects of those systems. The same cannot be said for artificially intelligent systems. Legislators will have a much more difficult time comprehending what "all processes" means in a machine-learning context and, thus, exactly what processes they want

[33] *Id.* at 10-16.

[34] *Id.* at 18-30.

[35] *Id.* at 21.

to force into the open. Had Citron and Pasquale engaged with the unique processes of machine learning — many of the playing-with-the-data stages — they could have provided legislators with the more detailed guidance they will need when confronting artificial intelligence.

Second, Citron and Pasquale could have avoided providing prescriptions that may not be entirely achievable in machine-learning contexts. For example, they argue that, to facilitate testing of algorithms, Federal Trade Commission ("FTC") experts should have access to the "programmers' notes describing the variables, correlations, and inferences embedded in the scoring systems' algorithms."[36] Such a call seems to be in tension with the very nature of machine learning — one that takes the human element largely out of embedding correlations and inferences in an algorithm. Similarly, the authors call for FTC assessments to reveal to the public "the logics of predictive scoring systems."[37] Citron and Pasquale do not specify what they mean by the "logics" of a system, but if they are referring generally to reasons for why a system generates the scores it does, some forms of reasons are more extractable from machine learning than others.[38] These oversights could have been avoided had Citron and Pasquale paid greater attention to the technical details of machine learning.

Again, despite this article's focus on the running model and its consequent faults, it proves an extremely valuable contribution to the literature. It is cognizant of the potential for disparate impacts in scoring systems, and it also clearly voices a myriad of broader concerns about due process. We will later build on these concerns — particularly the concern over inaccurate, or arbitrary, scoring — to show how a greater understanding of the machine-learning workflows can enrich the discussion.

C. Big Data's Disparate Impact

Neither of the two bodies of literature surveyed above probes playing with the data. Rich's article, and Fourth Amendment scholarship generally, is premised on an assumed running model, with the passing recognition that things can go wrong when programming an algorithm. But human programmers do not otherwise come into play. Similarly, Citron and Pasquale's article assumes running credit-

[36] *Id.* at 25.

[37] *Id.* at 26.

[38] *See infra* Part III.B.

or risk-scoring models, does not differentiate machine-learning systems from conventional systems, and misses the opportunity to transform otherwise vague calls for transparency and accountability into actionable prescriptions. There is, however, one seminal article that does come close to exploring our stages of machine learning — *Big Data's Disparate Impact* by Solon Barocas and Andrew Selbst.[39] First made public in 2014, this article deserves great credit for being one of the earliest to provide a detailed legal analysis of data mining. Particularly, the authors take on Title VII, asking whether it can effectively curtail discrimination by algorithms used in employment contexts. Their answer is a resounding "no."

The first step of this argument is a discussion of how black-box algorithms can yield discriminatory predictions, and it is here that playing with the data is paid some notice. Namely, Barocas and Selbst describe how bias can be introduced, often unintentionally, during three stages of data mining: defining the output variable and labeling its constituent classes,[40] collecting and labeling the training data,[41] and selecting the input variables.[42]

In the first stage, programmers must specify an algorithm's output variable — what is to be estimated or predicted — and there is often ambiguity in this process. If programmers specify output variables in ways that make members of certain demographic groups more likely than others to have "advantageous" outcomes, discrimination can be introduced. Two additional problems can emerge in the second stage, collecting an algorithm's training data: relying on data that reflect existing human biases, also known as the "garbage-in- garbage-out"[43] problem, and sampling data in a non-representative manner from the population of interest.[44] Finally, programmers collect data with an eye towards eventually including certain variables as inputs to an algorithm — Barocas and Selbst's third stage.[45] In making this decision, programmers can introduce bias by selecting sets of variables

[39] Barocas & Selbst, *supra* note 6.

[40] *Id.* at 677-80.

[41] *Id.* at 680-87.

[42] *Id.* at 688-92.

[43] *Id.* at 683 (internal quotation marks omitted).

[44] *Id.* at 684-87.

[45] Barocas and Selbst refer to this decision as "feature selection," a term that we have deliberately eschewed here. When discussing machine learning, "feature selection" tends to more often refer to a semi-automated process of paring down the number of input variables while avoiding a reduction in accuracy. We adopt this definition of "feature selection" and describe it in greater detail in Part II.H.3. Barocas & Selbst, *supra* note 6, at 688.

that are more predictive for members of certain groups[46] or that serve effectively as proxies for membership in a group.[47]

After introducing these sources of algorithmic discrimination, Barocas and Selbst contend that disparate treatment doctrine will largely fail to constrain these harms; absent inclusion of variables indicating protected class membership, it will be difficult to find conscious discriminatory motives in decisions tied to the output of a black-box algorithm.[48] Disparate impact doctrine may, on its face, be better suited to combat algorithmic discrimination, but this too may fall short; many predictive algorithms will likely survive judicial scrutiny by passing a business necessity test and showing job-relatedness.[49] There are nuances to these arguments, but the takeaway is simple — Title VII is not sufficient.

There is clearly much to laud in this article. More attention has been paid to the gritty details of working with data than in any other piece of legal literature. Also, if we were concerned that Rich's article performed poorly on the "subject test," *Big Data's Disparate Impact* fares much better. Barocas and Selbst do not overlook the fact that "*analysts* will often face difficult choices"[50] and that analysts must exercise caution when making these choices. But even this exemplary article falls short of fully and specifically teasing out playing with the data.

For one, and most obviously, this article fails to consider all of the stages of machine learning between input variable selection and the deployment of the running model. In other words, if a machine-learning algorithm is considered a black box, the authors have not cracked open that box fully. They catalogue harms caused by faulty inputs, but they neglect how the stages of machine learning that occur "inside" the black box can provide opportunities to *remedy* those harms.

This criticism should not be taken at all as an indictment of this fine article. Its legal questions are narrow, and, accordingly, the article properly focuses on the narrow issue of how data mining can cause disparate impacts, not necessarily how it can correct for them. Furthermore, a follow-on piece published earlier this year addresses some of this deficiency. This article, *Accountable Algorithms* was co-

[46] *See id.* at 688-90 (describing introduction of bias in the job application process based on the applicant's college or university's reputation and "redlining" by financial institutions).

[47] *Id.* at 691.

[48] *Id.* at 694-701.

[49] *Id.* at 701-14.

[50] *Id.* at 681 (emphasis added).

authored by Barocas and a team from Princeton's Center for Information Technology Policy ("CITP"), and it focuses on a set of steps to ensure "procedural regularity" of algorithms.[51] Although the bulk of this article focuses on algorithms generally defined, the authors do briefly take up machine learning specifically in Part III. There, they build on *Big Data's Disparate Impact* to, in part, address some of the burgeoning technical literature showing how disparate impacts can be mitigated.

A more minor flaw of this article is that, despite it being a response to the proliferation of data mining, it is not truly focused on data mining. This article deserves the special attention it has received because it provides the single most detailed explication of *parts* of the machine-learning, or data-mining, process. Due to this acclaim, this article has become an essential primer on machine learning for many legal scholars. But, in many ways, the article is not focused on machine learning; the parts of the process to which it pays attention are not those unique to machine learning. Barocas and Selbst's concerns are primarily concerns about the *data* algorithms analyze, not the algorithms' internal math that accomplishes the analysis. Therefore, readers should be aware that, despite this article's excellence, it is not a thorough treatment of data mining. Again, this is a minor point. However, given this article's deserved prominence, it is incumbent upon other scholars to correct any flaws or shortcomings, regardless how slight.

D. Machine Learning According to Lawyers

With this kind of scholarship becoming increasingly common, a particular conception of machine learning is taking hold in law. What does this lawyer-generated conception look like? What does it say about both what machine learning is technically and what the corresponding harms and benefits of machine learning are?

---

[51] Joshua A. Kroll et al., *Accountable Algorithms*, 165 U. PA. L. REV. 633, 637-38, 656-57, 662-72 (2017). Given this article's recent publication and thorough nature, readers may wonder why we do not engage with it in more depth. Put simply, asking how *Accountable Algorithms* fares under the playing-with-the-data critique is a bit of a non-sequitur. This is because the article, by and large, is not about machine learning. It does take up machine learning specifically in Part III, as we discuss in the accompanying text, but the rest of the article surveys software systems that tend to be more traditionally developed — the work of programmers (not data scientists) who create decision-making systems based entirely on human logic. The authors offer considered analysis on that topic, and it is commendable that they did not fall into the trap of holding out their article to be one focused primarily on machine learning. But, because it is not, we do not subject it to our critique.

On the technical side, a few scholars have recognized that machine learning starts with problem definition and data collection. An analyst must translate an abstract, often ill-defined goal into a highly specified outcome variable to be predicted by the algorithm. Then, data must be collected for both this outcome variable and any input variables fed into the algorithm. Up to this point, the lawyers seem to have a generally sensible view of the machine-learning process; we will add much nuance to these initial stages, but they have not been ignored (at least by some).

After problem definition and data collection, though, the magic of machine learning seems to begin. Out of the ether apparently springs a fully formed "algorithm," or "model," ready to catch criminals, hire employees, or decide whom to loan money. Barocas and Selbst, in describing the "key steps in the overall data mining process,"[52] jump right from discussing how analysts choose an algorithm's input variables[53] to cataloguing how that algorithm discriminates;[54] the data *mining* process is reduced to merely a data process. Rich's automated suspicion algorithms are similarly "generated"[55] out of the blue. He acknowledges in one sentence that "generated" algorithms result from training,[56] but what this means goes undiscussed, as does selecting a particular kind of algorithm or model. Finally, Citron and Pasquale premise their entire article on artificially intelligent scoring algorithms without any real discussion of how those algorithms are built.

With such a singular focus on the running model and a failure to consider stages of machine learning after data management, scholars have been forced to adopt an overly narrow view of algorithms' potential harms and benefits. Inaccuracy and bias are paid much attention, and they can indeed be traced back in part to poor data and variable specifications. But they can also creep in during other stages of machine learning,[57] and many harms arise almost entirely during those other stages. In fact, some of the most viscerally unsettling harms of machine learning — its opacity and lack of explainability — are brought about when algorithms are chosen and developed, not when data are collected or variables are specified.[58]

---

[52] Barocas & Selbst, *supra* note 6, at 677.

[53] *Id.* at 688-90.

[54] *Id.* at 691-92.

[55] Rich, *supra* note 1, at 885.

[56] *Id.*

[57] *See infra* Parts III.A, III.C.

[58] *See infra* Part III.B.

If a key motivation for writing on machine learning is the need to offer prescriptions or regulations to counter algorithmic harms, then this incomplete picture of harms should give scholars pause. By letting themselves be distracted by data and the running model, they have missed many opportunities for regulation. Our Article fills these gaps.

II. THE STAGES OF MACHINE LEARNING

Machine learning is not a monolith. For readers coming to this Article from the budding legal literature on machine learning, this point must be emphasized. As we noted earlier, it has become all too common to use "machine learning," "artificial intelligence," "big data analytics," and "data mining" as catchall phrases for an ever-changing family of myriad algorithms that, in reality, differ from one another in important ways.

"Machine learning" is what drives Netflix's recommendations,[59] predictive policing,[60] and self-driving cars,[61] to name a few applications. However, each relies upon different kinds of algorithms. When such nuance is lost, its policy and legal implications fall by the wayside as well. Potential harms are missed, and prescriptions for mitigating those harms remain underdeveloped.

To start filling these gaps, we provide in this Part a rich breakdown of the machine-learning process, one that is widely applicable and fairly comprehensive. Of course, it is impossible to discuss in depth the myriad ways in which every possible machine-learning algorithm differs from every other; we do not hold out our description as doing so. Nor do we assert that all machine-learning projects fit neatly within our breakdown. As we noted before and as we will demonstrate throughout this Part, much machine learning dances back and forth across our steps instead of proceeding through them linearly. Some

---

[59] *See* Chris Alvino & Justin Basilico, *Learning a Personalized Homepage*, NETFLIX TECH. BLOG (Apr. 9, 2015), https://medium.com/netflix-techblog/learning-a-personalized-homepage-aa8ec670359a [https://perma.cc/X43V-R3S6]; Xavier Amatriain & Justin Basilico, *Netflix Recommendations: Beyond the 5 Stars (Part 1)*, NETFLIX TECH. BLOG (Apr. 6, 2012), https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429 [https://perma.cc/DHD5-F2Q2]; Xavier Amatriain & Justin Basilico, *Netflix Recommendations: Beyond the 5 Stars (Part 2)*, NETFLIX TECH. BLOG (June 20, 2012), https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-2-d9b96aa399f5 [https://perma.cc/3T9U-5E22].

[60] *See* Shapiro, *supra* note 29, at 459; PREDPOL BLOG, *supra* note 29.

[61] *See* Alexis C. Madrigal, *The Trick that Makes Google's Self-Driving Cars Work*, THE ATLANTIC (May 15, 2014), https://www.theatlantic.com/technology/archive/2014/05/all-the-world-a-track-the-trick-that-makes-googles-self-driving-cars-work/370871 [https://perma.cc/7LGE-LNFZ].

particularly advanced techniques even push our breakdown to its breaking point, eschewing or dramatically restructuring some of our steps. This is particularly true for some "deep learning" techniques, which originally took inspiration from the neural architecture of the human brain and which, roughly speaking, find complex ways of changing how input variables are represented and then nonlinearly put together to yield predictions or estimates.[62] Despite these limitations, our breakdown should prove extremely useful to legal scholars taking on a variety machine-learning systems. Critically, our breakdown enables a deeper technical understanding of machine learning's fundamental components. As a result, legal scholars will be better placed to spot potential areas of concern and to identify possible technical solutions.

We have aimed for a level of depth about midway between the extremes of "gentle introduction" and "deep dive." We err on the side of including too much rather than not enough detail, and we delay until Part III most of the direct applications to legal scholarship. The impatient reader might skip to Part III, reserving the stages in this Part for later reference.

We contend that there are eight key steps often underlying machine learning, which can each be put into one of two workflows: "playing with the data" and the "running model." Playing with the data encompasses the first seven steps — problem definition through model training — and is where much of the data scientist's real work comes in. The last step — deploying the model in the real world — constitutes the "running model" workflow. As we previewed in Part I, most legal scholarship is premised on, or improperly assumes, a running model in existence, paying little attention to playing with the data. There are a few exceptions; some authors, particularly Barocas and Selbst, pay due notice to the deleterious effects of discriminatory or inaccurate data, as well as poorly defined outcome variables. In other words, they cover reasonably well the first three steps of our workflow: problem definition, data collection, and data cleaning. (Accordingly, we will not devote as much space to these stages as to others.) But the remaining steps, which, crucially, uniquely define machine learning, have been largely ignored, until now.

A. Definitions and Terminology

Before delving into the stages of machine learning, let us briefly clarify, on a more intuitive level, what machine learning is and what

[62] *See generally* IAN GOODFELLOW ET AL., DEEP LEARNING 1-8 (2016).

key terminology we will use. Fundamentally, machine learning refers to an automated process of discovering correlations (sometimes alternatively referred to as relationships or patterns) between variables in a dataset, often to make predictions or estimates of some outcome.[63]

We will refer to machine-learning "algorithms," or "models," and we will use these two terms interchangeably throughout this Article. Both terms will carry slightly different meanings depending on the where in the machine-learning process they are being invoked. In their first instances — that is, at the start of a machine-learning process — "algorithms" and "models" refer to a set of mathematical steps for achieving the learning described above when exposed to historical or example data. This process is known as "training" the algorithm or model.[64] In all algorithms, notwithstanding their mathematical nuances, training occurs via the optimization of an objective, or loss, function.[65]

Let us break this down. First, what is an objective function? It is a mathematical expression of the algorithm's goal. Often, this is some expression of *accuracy* or *inaccuracy*.[66] Take, for instance, an algorithm that predicts whether a stock trade is the result of insider trading. The objective function might, in a simplified sense, represent the percentage of transactions the algorithm incorrectly identifies — legitimate trades it identifies as fraudulent and fraudulent trades it identifies as legitimate. With this established, what does it mean for training to occur via the optimization of this objective function? It means that the algorithm will learn to make predictions such that the

---

[63] KEVIN P. MURPHY, MACHINE LEARNING: A PROBABILISTIC PERSPECTIVE 1 (2012) ("[W]e define machine learning as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty (such as planning how to collect more data!).."). For more detail on the possible applications and purposes of machine learning, see *infra* Part II.B.

[64] *See* MURPHY, *supra* note 63, at 2.

[65] *See* ETHEM ALPAYDIN, INTRODUCTION TO MACHINE LEARNING 41-42 (2d ed. 2010) (discussing a loss function as a dimension of supervised machine-learning algorithms); RICHARD BERK, STATISTICAL LEARNING FROM A REGRESSION PERSPECTIVE 14 (1st ed. 2008) ("Usual practice is to minimize a loss function with [the disparities between the observed response variable values and the fitted response variable values] as inputs."); Suzanna Becker & Richard S. Zemel, *Unsupervised Learning with Global Objective Functions*, *in* THE HANDBOOK OF BRAIN THEORY AND NEURAL NETWORKS 997, 997 (Michael A. Arbib ed., 1st ed. 1995) ("The main problem in unsupervised learning research is to formulate a performance measure or cost function for the learning, to generate this internal supervisory signal. The cost function is also known as an objective function, since it sets the objective for the learning process.").

[66] *See* BERK, *supra* note 65; Becker & Zemel, *supra* note 65.

objective function is either minimized or maximized, depending on how it is specified. In the stock trading example, the objective function was defined in terms of inaccuracy, so it would be minimized to achieve the goal of accurate prediction. The algorithm would try multiple possible predictive rules, ultimately choosing those that result in the minimization of the objective function.

After training, "algorithm" and "model" take on slightly different meanings. There, they refer to the set of predictively useful correlations, sometimes referred to post-training as "rules,"[67] discovered during training.[68] Take now, for instance, an algorithm predicting whether loan applicants will default on a loan. During training, the algorithm may have discovered that individuals with over twenty thousand dollars in credit card debt, no job, and three to five children are likely to default. This rule could then be used by the trained algorithm when it is deployed to make predictions in the real-world; individuals with twenty-thousand dollars in credit card debt, no job, and three to five children would be predicted as defaulting on a loan. Thus, although we will use the terms "algorithm" and "model" throughout, an algorithm or model pre-training carries a slightly different meaning than an algorithm or model post-training.

In addition to clarifying our use of "model" and "algorithm," let us also clarify our use of terms relating to individuals involved in the machine-learning process. We will refer to them interchangeably as "analysts," "data scientists," "statisticians," and "developers." The reader should not take the use of one of these terms over the others to imply anything unique about that individual's qualifications or his or her role in a particular stage of machine learning.[69]

B. Problem Definition

Machine-learning algorithms predict or estimate *something*, and the first step of any analysis is to define what that something should be

---

[67] *See, e.g.*, Barocas & Selbst, *supra* note 6 (referring throughout to the predictive "rules" inferred by a trained algorithm).

[68] We do not, however, intend for the use of "model" in the post-training context to imply a set of rules or correlations with *causal* significance — ones that ostensibly represent causal relationships in nature's underlying data-generating process.

[69] We may also refer to those individuals in either the singular or the plural, but the reader should recognize that great variety exists in terms of the size and makeup of teams working on machine learning. Sometimes, machine learning may be the province of a sole data scientist in a company. Other times, hundreds of individuals may be involved. We do not claim to be referring in any given instance to a particular size or kind of team.

and how it should be measured. By and large, the algorithms receiving the most legal attention are *supervised* learning algorithms. Supervised algorithms are given a labeled outcome variable (alternatively called an output or response variable) representing the true values to be predicted on the basis of input data.[70] So, for supervised algorithms, defining the problem means defining the outcome variable.[71]

Outcome variables can take many forms. They can be binary indicators — True/False, Yes/No, etc. — and algorithms predicting binary outcomes are commonly referred to as "classification" algorithms.[72] Classification algorithms can also be "multiple classification" or "multi-label classification" algorithms, which predict indicators with more than two classes — Pedestrian/Cyclist/Tree, Red/Green/Blue, etc.[73] Moving away from classification algorithms, algorithms applied to "regression" problems predict a continuous quantitative outcome — one that can take any numeric value.[74] And conceptually related to both classification and regression algorithms are those that estimate ordinal outcomes — outcomes with multiple classes that, while not falling along a continuous number line, possess some inherent ordering, such as Cold/Warm/Hot or 1/2/3 or First/Second/Third.[75]

How do analysts specify an outcome variable? Fundamentally, doing so is an exercise in translating abstract goals into measurable outcomes. There are a couple of steps to this process.[76]

---

[70] *See* MURPHY, *supra* note 63, at 2-4.

[71] *See* Barocas & Selbst, *supra* note 6, at 677-80 (describing the process and difficulties of defining the outcome variable for supervised learning algorithms).

[72] MURPHY, *supra* note 63, at 2.

[73] *See id.* at 3.

[74] *Id.* at 8-9. Note that the use of the phrase "regression" here is not meant to imply anything about the math of a machine-learning algorithm applied to predict a continuous outcome variable. While there are some machine-learning methods that use a least squares-based objective, or loss, function and, thus, resemble a general linear model mathematically, others bear less of a resemblance. *See, e.g.*, BERK, *supra* note 65, at 301 ("Support vector machines (SVM) will seem somewhat far afield from the [regression-related] statistical learning procedures discussed to this point . . . [but] [s]upport vector machines now can have more the look and feel of regression.").

[75] MURPHY, *supra* note 63, at 2.

[76] We are assuming throughout this discussion that the data scientist in question has the practical ability to collect his or her own data set, creating de novo the outcome variable in the process. There may, of course, be instances in which this is not possible. Companies or other institutions may have pre-existing databases, and data to be fed into a machine-learning algorithm might, practically, have to be culled from these databases.

First, the abstract goal has to be translated into a decision about what, conceptually, to predict. A prison warden may, for instance, have as his overarching policy objective reducing inter-inmate altercations. Accordingly, he may decide that the predictive endeavor best suiting this goal is predicting whether each new inmate who walks through the door is likely to be involved in a violent altercation while in prison; if an inmate is predicted as likely to be so, he or she could be placed under increased supervision, thus reducing the risk of altercations.[77] Similarly, an autonomous vehicles developer may have as her goal creating a car that minimizes human casualties. Part of reaching this goal may be creating an algorithm within the vehicle that enables it to predict whether objects in its surroundings are, say, pedestrians, animals, or trees.[78] Thus, in both examples, a decision-maker has gone from an abstract goal to a predictive goal.

But it is in the next step where the rubber really hits the road. In this step, the decisionmaker must translate the predictive goal to a specified outcome variable. The ease with which this can be accomplished depends on how distant the predictive goal is from a fully specified outcome variable specification. Take the autonomous vehicle example. There, the predictive goal — determining whether an object is a pedestrian, animal, or tree — entirely dictates the form of the output variable; it will be a categorical variable with three classes, and each of the classes will be readily codeable by a human to create training data.

This is less the case in the prison example. There, the predictive goal is predicting individuals' potentials for committing acts of violence, but what kind of outcome variable could represent this concept? There are many possibilities: a binary variable indicating an individual's involvement in a violent altercation within the first year of incarceration, a continuous variable indicating how many times an individual is involved in a violent altercation during the entirety of his or her incarceration, a continuous variable indicating the average number of altercations in which a prisoner is involved per year of his

[77] *Cf.* Richard A. Berk et al., *Forecasting Domestic Violence: A Machine Learning Approach to Help Inform Arraignment Decisions*, 13 J. EMPIRICAL LEGAL STUD. 94, 94-96 (2016) (describing how, when building an algorithm to inform courts' decisions of whether to release those arrested for domestic violence between their arraignments and hearings, a motivating abstract goal is to mitigate the threat to public safety, as described in the Bail Reform Act of 1984, which the researchers translate into the predictive goal of forecasting whether those arrested will engage in future domestic violence if released).

[78] This is obviously an overly simplistic example of how object recognition occurs in autonomous vehicles, but it is instructive nonetheless.

or her incarceration, etc.[79] Even more difficult, imagine an employer is a decisionmaker whose predictive goal is identifying good employees.[80] What kind of outcome variable measures the "goodness" of an employee? Whether the employee is promoted? How long the employee stays with the company? How many overtime hours the employee works? Often the true value the abstract or predictive goal concerns — say, happiness or human flourishing or dignity or autonomy — cannot be measured at all, and the best analysts can do is find a reasonable proxy. Too often, data scientists and others engaged in what is now referred to as "evidence-based policymaking" lose sight of the intrinsic limit of targeting policy only on what we can measure.

A few different factors play into how data scientists make these tough choices. First, there is subject matter knowledge. Those steeped in a particular context may have institutional knowledge that would give them good reason to believe that, say, employee tenure is likely to be a better indicator of a quality employee than his or her promotion. Second, there are technical implications to be mindful of. As we will detail later in this Part, different algorithms are capable of working with different outcome variable forms and of producing different kinds of supplementary output for different outcome variable forms.[81] Therefore, if other concerns cause analysts to pursue a particular machine-learning algorithm or a particular kind of algorithmic output, they may be prompted to choose an appropriately specified outcome variable. Finally, and not to be discounted, are practical concerns surrounding resource limitations; it may simply be the case that certain outcome variable specifications are easier or less costly to measure. Of course, pursuing a particular outcome variable for the sake of convenience carries with it a greater risk of mismatch between the predictive goal and the variable's specification.

In addition to the above three factors, though, many legal scholars would rightfully hope that a fourth factor would form part of data scientists' deliberations — the potential for harms, particularly discrimination, to result from certain specifications. Indeed, many legal scholars, most notably Barocas and Selbst, have provided excellent explanations of how certain outcome variable specifications, including ones purportedly identifying good employees, can

---

[79] *Cf.* Berk et al., *supra* note 77, at 98-99 (describing how stakeholder input factored into deciding the form of the outcome variable representing future domestic violence, both in terms of specifying it as a three-class variable indicating future arrest after arraignment and measuring that arrest two years after arraignment).

[80] *See* Barocas & Selbst, *supra* note 6, at 679-80.

[81] *See infra* Part II.G.

exacerbate historical societal biases.[82] We will not reiterate the details of those points here, but the potential for these harms warrants being aware of how exactly data scientists attempt to tackle problem definition.

The above discussion has applied to supervised learning, but this is not the only kind of machine learning.[83] Unsupervised learning algorithms do not predict outcome variables labeled with ground truth. Instead, they group or cluster subjects together based, roughly speaking, on how similar their input data values are.[84] This clustering can be an end in itself or a stepping stone on the way to a supervised approach; the groups resulting from an unsupervised learning algorithm can serve as classes predicted by a supervised algorithm.[85] For an unsupervised algorithm, then, problem definition entails deciding on a particular mathematical measure of similarity.[86]

We will not provide further details on unsupervised learning for two reasons. First, doing so would require even more mathematical explanation. Second, and more importantly, learning algorithms other than supervised ones are generally of the least legal salience, at least for the time being. The literature to which we are responding has focused almost exclusively on supervised algorithms because it is these algorithms driving many legally consequential decisions — risk assessment,[87] predictive policing,[88] credit scoring,[89] employment

[82] Barocas & Selbst, *supra* note 6, at 681-84.

[83] In addition, there is yet another form of machine learning — reinforcement learning. MURPHY, *supra* note 63, at 2. Briefly, in reinforcement learning, an agent is placed in a model environment and figures out the optimal set of actions to take to reach a certain goal. ALPAYDIN, *supra* note 65, at 13-14. The agent and the environment in which it is placed can be physical, as when a robot learns to navigate a maze, or entirely mathematically modeled, as when a program learns how to master Go. *See* David Silver et al., *Mastering the Game of Go with Deep Neural Networks and Tree Search*, 529 NATURE 484, 484-87 (describing how Google DeepMind's AlphaGo algorithm is a combination of supervised learning and reinforcement learning). In either circumstance, critical to defining the problem is defining the "reward" that can accompany the agent's actions, as the agent will ultimately learn to take actions that maximize its reward. In the case of a maze-navigating robot, for instance, the reward might be distance to the end of the maze. Thus, for reinforcement learning, problem definition can be thought of as a combination of modeling the environment and specifying the reward based on the goal manifested in the environment.

[84] MURPHY, *supra* note 63, at 9-13.

[85] In such instances, the machine-learning systems are often referred to as "semi-supervised." *See, e.g.*, OLIVIER CHAPELLE ET AL., SEMI-SUPERVISED LEARNING 1-2 (Olivier Chapelle et al. eds., 2006).

[86] *See* MURPHY, *supra* note 63, at 877-78.

[87] *See, e.g.*, Berk et al., *supra* note 77.

[88] *See, e.g.*, Tong Wang et al., *Learning to Detect Patterns of Crime*, *in* MACHINE

processes,[90] etc. Therefore, supervised algorithms will also be our focus; although some of the points we make in the subsequent sections may be just as applicable to unsupervised and other forms of learning as they are to supervised learning, they should be read and understood foremost in the latter context.

C. Data Collection

Once data scientists have conceptualized the goal of the machine-learning system and reduced that goal to a specified outcome variable, the data themselves have to be assembled — both for the outcome variable and the myriad input variables. For many projects, this can be the most time-consuming stage, and it also holds enormous consequences; as commenters have noted previously, an algorithm is, at the end of the day, only as good as its data.[91]

Roughly speaking, there are two different approaches to data collection: gathering and merging data that have already been measured, and measuring data.[92] These approaches are not mutually

LEARNING AND KNOWLEDGE DISCOVERY IN DATABASES 515, 517-18 (Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen & Filip Železný eds., 2013) (describing their approach to identifying patterns of crime as a supervised learning method).

89 *See, e.g.*, Amir E. Khandani et al., *Consumer Credit-Risk Models via Machine-Learning Algorithms*, 34 J. BANKING & FIN. 2767, 2775-77 (2010) (describing their credit-risk model as a supervised algorithm, a classification tree).

90 *See, e.g.*, Qasem A. Al-Radaideh & Eman Al Nagi, *Using Data Mining Techniques to Build a Classification Model for Predicting Employees Performance*, 3 INT'L J. ADVANCED COMPUTER SCI. & APPLICATIONS 144, 145-46 (2012) (describing a supervised classification tree used for forecasting employee performance).

91 *See, e.g.*, Harry Surden, *Machine Learning and Law*, 89 WASH. L. REV. 87, 106 (2014) ("In general, machine learning algorithms are only as good as the data that they are given to analyze.").

92 There is another form of data collection, of sorts, that we do not take up in greater detail. Particularly, analysts can use techniques to derive from raw, often unstructured input information more useful information that can then serve as input variables. These techniques are extremely diverse and are referred to variously as "feature extraction," "feature generation," "feature engineering," "feature construction," "feature processing," or "feature transformation." *See* HUAN LIU & HIROSHI MOTODA, FEATURE EXTRACTION, CONSTRUCTION AND SELECTION: A DATA MINING PERSPECTIVE 3-5 (1998); AMAZON WEB SERVICES, AMAZON MACHINE LEARNING: DEVELOPER GUIDE 13-14 (2017), http://docs.aws.amazon.com/machine-learning/latest/dg/machinelearning-dg.pdf [https://perma.cc/49JC-P8E9]; Brad Severtson et al., *Feature Selection in the Team Data Science Process (TDSP)*, MICROSOFT AZURE (Mar. 24, 2017), https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-data-science-select-features [https://perma.cc/C9PN-YXAF]. What exactly these techniques entail differs depending on the context. But, taking natural language processing as an example, feature extraction might refer to extracting the frequencies of words or groupings of words in texts, two pieces of information that could

exclusive, and both can be used within the same project. Regardless of the approach, though, a few factors weigh on the data scientists' minds when evaluating the quality of collected data.[93]

For one, and at the most basic level, data scientists want to ensure that they have collected *enough* data. Although there is no technical bar to running machine-learning algorithms on small data sets, doing so is, in practice, pointless. Many machine-learning algorithms have been proven to perform at least as accurately, and often many times more so, than conventional techniques, but this property is asymptotic; it occurs only as the number of observations in a dataset grows towards infinity.[94] To reap the predictive benefits of machine learning, a sufficiently large number of observations is required. There is no hard and fast rule about, in practice, how many observations is enough. Having only a few hundred to a few thousand observations may often be insufficient. Beyond that, tens of thousands could be

then be used as inputs to a machine-learning algorithm. *See, e.g.*, *4.2. Feature Extraction*, SCIKIT-LEARN, http://scikitlearn.org/stable/modules/feature_extraction.html [https://perma.cc/62WB-5F5R] (last visited Sept. 16, 2017). Similarly, in face recognition systems, feature extraction could entail extracting grayscale values of pixels and surrounding regions from faces of photos. *See generally* M. Saquib Sarfraz et al., *Feature Extraction and Representation for Face Recognition*, *in* FACE RECOGNITION 1, 14 (Milos Oravec ed., 2010). In such applications, choices taken by an analyst in this kind of data collection can have enormous consequences on the quality of the end-product algorithm. We do not, however, provide a deeper dive into these choices because (1) there is too much variation across contexts to speak at the broad level of detail that would be required in this Article, and (2) by and large, the supervised machine-learning techniques on which this Article focuses do not frequently rely on these techniques that create new input features.

[93] We recognize that social sciences, particularly psychology, have ascribed various "validity" and "reliability" labels to many concepts of data quality we discuss throughout this section. *See generally* JUM C. NUNNALLY & IRA H. BERNSTEIN, PSYCHOMETRIC THEORY 83-114 (James R. Belser & Jane Vaicunas eds., 3d ed. 1994) (describing different major meanings of validity in the context of psychology). We avoid using those labels, though, for two reasons. First, there can be overlap or ambiguity in how different social scientists use particular terms, and we do not wish to wade into that murkiness. Second, many of the labels, such as "internal validity," are most meaningful in contexts where the statistical methods at issue attempt to yield causal inferences, which machine learning does not.

[94] *See, e.g.*, V. Koltchinskii & D. Panchenko, *Empirical Margin Distributions and Bounding the Generalization Error of Combined Classifiers*, 30 ANNALS STAT. 1, 2 (2002) (demonstrating this asymptotic convergence for combined classifiers, such as those that incorporate both the processes known as bagging and boosting); LEO BREIMAN, TECHNICAL REPORT 577: SOME INFINITY THEORY FOR PREDICTOR ENSEMBLES (2000), https://www.stat.berkeley.edu/~breiman/some_theory2000.pdf [https://perma.cc/8NPE-X974]. For reference, this paper deals with proving upper limits of combined classifiers' generalization error, which is the "expected value of the misclassification rate when averaged over future data." MURPHY, *supra* note 63, at 23. For more information on what an expected value is, see *infra* note 157.

sufficient for some applications, and other applications operate optimally on hundreds of thousands or even millions of observations.

In addition to collecting enough data, data scientists try to ensure that the variables for which data are collected indeed measure what they are supposedly measuring. This inquiry occurs at two levels. First, there is the question of whether the particular techniques that went into measuring a variable allow it to accurately and precisely measure what it is claimed the variable indicates *on its face*. For instance, if a variable in a violence risk assessment algorithm supposedly indicates whether an individual has ever been previously charged with gun-related violence, then it is reasonable to think that the likely source of the data — court records — faithfully recounts criminal charges. On the other hand, if a dating website is developing an advertising algorithm to which one input is a user's self-reported age, there is likely reason to believe that some users have lied about their ages.[95] In that case, the variable would not as validly measure what, on its face, it supposedly does.

The idea of "faithful measurement" also raises a deeper issue. Measurements must be faithful not just to what a variable ostensibly indicates on its face, but also to what underlying construct (also called a latent construct) the data scientist believes it represents. This goes to the reason for which a data scientist sought data for that variable in the first place. Take again the two examples in the preceding paragraph. When designing the hypothetical risk assessment algorithm, data scientists may have sought out the gun charges variable because it could indicate how prone someone is to commit violent acts. But it is obvious that charges for gun-related violence is a flawed indicator of tendency for violence; many individuals commit violence for which they are not arrested and charged, and others are charged for violent crimes they never committed. Thus, while this variable may be validly measured on its face, it likely is not when considering its underlying construct. The opposite could be true of the age variable in the dating website's advertising algorithm. Users may fudge their ages by a few years,[96] but obtaining users' *exact* ages is likely not the motivating reason for the website to include this variable. Rather, the website cares about what general age ranges users lie in; being twenty-eight as opposed to twenty-six is unlikely to have

[95] *See* Jeffrey T. Hancock et al., *The Truth About Lying in Online Dating Profiles*, PROC. SIGCHI CONF. HUM. FACTORS COMPUTING SYS., Apr. 28–May 3, 2007, at 449, 451-52 (2007) (analyzing frequency of deception in dating profiles).

[96] *Id.* at 451 (noting that subjects actual ages ranged from three years younger to nine years older than they reported, with an average deviation of 0.44 years).

a large bearing on one's responsiveness to certain ads, but being twenty-eight as opposed to being forty might. Because it is rarer for users to lie egregiously about their ages and, thus, put themselves in the improper "age bin" relevant for advertising purposes, this variable could be validly measured when considering its underlying construct, even if it is not when considering it on its face.

Finally, in addition to the size of the dataset and variables' measurement validities, data scientists are concerned with generalizability — the ability of an algorithm trained on a particular dataset to generate accurate predictions when deployed on different data.[97] To facilitate this, the collected data used for training purposes should be representative of the real-world data on which the algorithm will eventually be deployed. When collecting data entails going out into a population and measuring data anew, data scientists can attempt to achieve representative data by randomly sampling the individuals of the population for whom they measure data.[98] Of course, there will always be some risk that the makeup of the population at the time of data measurement will not be the same as the makeup of the population at the time of algorithm deployment. But, generally, random sampling from the population of interest ensures a reasonable likelihood of obtaining representative data. Unfortunately, true random sampling rarely occurs in practice. Take, for instance, a financial company developing a credit-scoring algorithm. If the outcome variable is some measure of whether an individual defaulted on a loan, then the training data will have to be composed of individuals who applied for *and were granted* a loan. But this is clearly not the population of interest — all individuals who apply for a loan. Thus, such an algorithm is likely to be less predictive on individuals similar to those who, in the past, applied for a loan but were rejected. This also has implications for the potential discriminatory effects of machine learning; if those who were rejected for loans in the past were

[97] *See, e.g.*, MURPHY, *supra* note 63, at 3 ("Our main goal is to make predictions on novel inputs, meaning ones that we have not seen before (this is called generalization), since predicting the response on the training set is easy (we can just look up the answer).") (emphasis omitted).

[98] *See generally* William Kruskal & Federick Mosteller, *Representative Sampling, III: The Current Statistical Literature*, 47 INT'L STAT. REV. 245 (1979) (describing current use of the term "representative sampling" in modern statistical literature); William Kruskal & Federick Mosteller, *Representative Sampling, IV: The History of the Concept in Statistics, 1895–1939*, 48 INT'L STAT. REV. 169 (1980) (describing history of representative sampling in modern statistics).

disproportionately minority applicants, then the resulting algorithm may perform worse on exactly those applicants.[99]

D. Data Cleaning

Collecting data with an eye towards quantity, validity, and generalizability is but one aspect of preparing the data. Datasets are rarely, if ever, free from missing and inaccurate values. Although these problems are not show-stoppers — analysts do not abandon a project merely because they have to wrestle with data — they can hugely affect the quality of predictions. The data cleaning stage aims to mitigate this threat.

For a variety of reasons, subjects within a dataset might be missing values for one or more variables. Perhaps a website user opts not to provide his or her sex when filling out a registration form. Perhaps someone tasked with digitally entering the data on a handwritten form cannot make out what was scribbled in one field. Perhaps a data scientist working in a program like Microsoft Excel accidentally deletes the contents of a cell. Mistakes happen in the messy world of data.

What can be done about these missing values? One option is simply to delete the subjects with missing values from the dataset; if an individual is missing a value for his or her age, simply exclude him or her entirely from analysis. As we stated earlier, though, having enough observations is critical to machine learning's success, so this solution may seem problematic.[100] How problematic it is depends on a couple of factors. If the dataset is sufficiently large, deleting some, perhaps even many, subjects may not impair the ability of the data to mostly cover all possible variability between subjects; deleting one hundred from a dataset of one million is unlikely to be damaging. Another consideration is whether the subject with the missing data has other data values that are not well represented in other subjects. Representative data are key to generalizability, and, if there are few subjects with certain characteristics, deleting even a relatively small number of them could reduce generalizability. If, for instance, a dataset of one million individuals contains only one thousand

[99] Many scholars have offered excellent explanations of this and other data-related sources of discriminatory impacts. *See* Barocas & Selbst, *supra* note 6, at 677-94; Kate Crawford, *Think Again: Big Data*, FOREIGN POL'Y (May 10, 2013), http://foreignpolicy.com/2013/05/10/think-again-big-data [https://perma.cc/8VH6-PKVP]. We do not wish to repeat all of their findings here and instead direct the reader towards this existing literature.

[100] *See supra* Part II.C.

individuals of a particular race, then deleting one hundred of them could jeopardize generalizability. As with many of the tradeoffs discussed throughout this Part, there is no strict mathematical guidance for making these choices; they are judgment calls, and analysts will have to bring subject matter knowledge to bear.

Another way of correcting missing values is to impute them — in other words, to, through an automated process, make educated guesses about what the missing values are likely to be. This can be accomplished in different ways. An analyst could employ a fairly rough guessing process — one that sets missing values of a given variable as, say, the median or mode value for that variable.[101] If, for instance, fifty-five out of one hundred individuals in a dataset are men, and one of the one hundred individuals is missing an observation for the sex variable, that missing value would be imputed as male. This approach is clearly likely to impute the wrong values for many individuals, but some algorithms allow for much more nuanced guessing processes. In those, the algorithm tries to identify other individuals in the dataset that are similar to those with missing values, and imputes values that tend to be held by the similar individuals.[102] Deciding to implement these methods entails a tradeoff. An analyst would likely be inserting some values that turn out to be wrong, and this, in turn, is likely to make the algorithm less accurate and less generalizable.

Outside of missing values, analysts also "clean" incorrect values. One immediate difficulty, though, is identifying incorrect values. Missing values tend to be hard to misinterpret — when viewing the data in tabular form, cells might be blank or filled in with an indicator like "NA." In contrast, incorrect values might appear legitimate.

---

[101] *See, e.g.*, LEO BREIMAN ET AL., PACKAGE 'RANDOMFOREST': BREIMAN AND CUTLER'S RANDOM FORESTS FOR CLASSIFICATION AND REGRESSION 10-11 (Oct. 7, 2015), https://cran.r-project.org/web/packages/randomForest/randomForest.pdf [https://perma.cc/JY9N-LYBY] (describing how the randomForest package for R contains the "na.roughfix" function, which allows for the imputation of missing values based on median or mode); DAVID MEYER ET AL., PACKAGE 'E1071': MISC FUNCTIONS OF THE DEPARTMENT OF STATISTICS, PROBABILITY THEORY GROUP (FORMERLY: E1071), TU WIEN 25 (Feb. 2, 2017), https://cran.r-project.org/web/packages/e1071/e1071.pdf [https://perma.cc/BD82-7UK7] (describing how the e1071 package for R, which implements a few machine-learning algorithms, such as support vector machines, contains the "impute" function, which allows for the imputation of missing values based on median or mode).

[102] *See* BREIMAN ET AL., *supra* note 101, at 23-24 (describing how the randomForest package for R contains the "rfImpute" function, which initially replaces missing values with the median or mode using "na.roughfix," and then runs the random forest algorithm on the data to generate proximity measures, which in turn update the previously roughly imputed values).

Absent some external hint that a value might be incorrect, such as knowledge a measurement tool was improperly used, analysts may not have much to go on. Values could, of course, be so extreme that they are most likely incorrect; a supposed ten-year-old individual in a dataset with information on credit card usage is certainly not ten years old, just as a company with a trillion dollars in revenue is undoubtedly fictional. Analysts must be on the lookout for these kinds of outliers, an endeavor that can be assisted by the techniques described in the next section.

If analysts do identify probably incorrect values, they face essentially the same choices as when encountering a missing value. Unless they can track down the original source of the data and obtain or measure the true value, they must either delete the corresponding subject or impute a more likely value. This decision would be governed by weighing the same factors described previously.

E. Summary Statistics Review

Where previous scholarship has paid attention to playing with the data, it has been at the last three steps: problem definition, data collection, and data cleaning. But much lies beyond that. Before any algorithm itself can be developed, an analyst reviews summary statistics of every variable — input and output — in the dataset.[103] With summary statistics, the analyst characterizes the values a given variable takes on. For quantitative variables, examining summary statistics might entail, for each variable, determining its minimum, twenty-fifth percentile, median, mean, seventy-fifth percentile, and maximum. For class variables, it might entail determining what percentage of cases are in each class.

Why does an analyst review summary statistics? One reason is to weed out outliers — values that fall very below or above markers like

[103] *See, e.g.*, BERK, *supra* note 65, at 337-38 ("All data explorations must start with an effort to get 'close' to the data. This requires a careful inspection of elementary descriptive statistics: means, standard deviations, histograms, cross-tabulations, scatterplots and the like."); IAN H. WITTEN ET AL., DATA MINING: PRACTICAL MACHINE LEARNING TOOLS AND TECHNIQUES 65 (2016) ("Simple tools that show histograms of the distribution of values of nominal attributes, and graphs of the values of numeric attributes . . . are very helpful. . . . [They] make it easy to identify outliers, which may very well represent errors in the data file . . . . Domain experts need to be consulted to explain anomalies, missing values, the significance of integers that represent categories rather than numeric quantities, and so on. Pairwise plots of one attribute against another, or each attribute against the class value, can be extremely revealing.").

the twenty-fifth and seventy-fifth percentiles, respectively. Outliers can be problematic for two reasons.

First, many (but, arguably, not all) machine-learning algorithms suffer from a phenomenon called "overfitting."[104] Outliers can result from noise, or randomness, which is present in all datasets. And, if a statistical method is particularly adept at finding correlations in data, it risks identifying as legitimate correlations due to randomness, including outliers, in the training data — randomness that will not be the same in the real-world data to which the algorithm is eventually applied.

A similar, but mathematically distinct, cause for concern over outliers is a lack of generalizability — an algorithm not performing as well on real-world data as it does on training and test data, but for a reason other than overfitting.[105] Namely, if certain variables take on *non-randomly* extremely high or low values in the training and test data, but not in real-world data, the rules an algorithm learns to make predictions in the former may fail on the latter.

Outside of searching for and eliminating outliers, there is another reason for analysts to examine summary statistics, particularly those of the outcome variable. As will be discussed in more detail in the next section, a key paradigm of machine-learning analysis is training an algorithm on one random sample of the entire dataset and evaluating — or testing — it on another.[106] Under this paradigm, a highly skewed outcome variable — one with relatively few high and/or low values — may necessitate sampling a larger percentage of the entire dataset for the training data and a lower percentage for the test data.

F. Data Partitioning

No professional analyst both develops a machine-learning algorithm and then evaluates it entirely on the same data.[107] The point of any predictive endeavor is to develop a tool that predicts accurately *in the real world*, and there are myriad reasons why the data an analyst collects may not resemble the real world. Data could have been

---

[104] *See, e.g.*, BERK, *supra* note 65, at 42; MURPHY, *supra* note 63, at 22; Pedro Domingos, *A Few Useful Things to Know About Machine Learning*, 55 COMM. ACM 78, 81-82 (2012).

[105] *See* Hancock, *supra* note 97, at 452.

[106] *See infra* Part II.F.

[107] *See, e.g.*, MURPHY, *supra* note 63, at 23 ("Unfortunately, when training the model, we don't have access to the test set (by assumption) . . . ."); Domingos, *supra* note 104, at 80 ("The most common mistake among machine learning beginners is to test on the training data and have the illusion of success.").

collected in a non-representative manner;[108] there could have been important changes in the population or system from which data were collected between when collection occurred and when an algorithm is deployed;[109] a sufficient degree of randomness could exist in the system that makes data collected at one point in time non-representative of the system at a later point.[110] For this reason, analysts have developed methods of gauging, with some unavoidable degree of uncertainty, how well an algorithm trained on one dataset will perform on another.

One key method is to randomly split, or partition, an entire dataset into two: a "training" dataset and a "test" dataset.[111] A machine-learning algorithm is trained and learns the optimal predictive rules on the former. Then, the algorithm's accuracy and other performance metrics are assessed by asking it to predict the outcomes of the subjects in the latter. In this way, an algorithm is forced to predict data it has not "seen" before, which analysts hope is a decent analogue to asking the algorithm to predict outcomes in the "real world."

Of course, this approach is not a panacea for the concerns mentioned earlier. If the cause for concern is randomness in a system, then the results of this method might be a decent proxy for real-world

---

[108] *See* PAUL J. LAVRAKAS, ENCYCLOPEDIA OF SURVEY RESEARCH METHODS 254-55 (2008) ("In order to avoid [threatening external validity], the researcher should make certain that the sampling design leads to the selection of a sample that is representative of the population."). *See generally* Walter A. Kukull & Mary Ganguli, *Generalizability: The Trees, the Forest, and the Low-Hanging Fruit*, 78 NEUROLOGY 1886 (2012) (discussing the concept of generalizability and how representative sampling factors into it, particularly in the context of clinical and epidemiological studies).

[109] *See* LAVRAKAS, *supra* note 108, at 255 ("In order for a survey to be characterized as externally valid, the results should be generalizable and essentially remain invariant across different points in time."). Recent scholarship has argued this mismatch could be a problem for algorithms informing judges' decisions to release individuals on bail; if the training data were collected before the implementation of bail reforms, but the algorithms were deployed after such implementation, they could predict individuals as riskier than they actually would be. *See, e.g.*, John Logan Koepke & David G. Robinson, Zombie Predictions and the Future of Bail Reform (Sep. 22, 2017) (unpublished manuscript), https://ssrn.com/abstract=3041622 [https://perma.cc/Z8AH-3HWC].

[110] *Cf.* LAVRAKAS, *supra* note 108, at 255 ("The major concern with this threat [of setting characteristics] to the external validity is that the findings of a particular survey research study may be influenced by some unique circumstances and conditions, and, if so, then the results are not generalizable to other survey research studies with different settings."). Randomness is one factor that could render the setting of a particular survey different than that of another survey at a different point in time.

[111] *See supra* note 107.

performance; just as the randomness in the real world will, by its very nature, be different than that of the training data, so will that of the test data. If, however, the cause for concern is non-representative sampling or changes in a system over time, then this approach may be less potent; the training and test data were, after all, derived from the same initial dataset collected at a particular point in time via a particular sampling method. Nonetheless, partitioning data into training and test sets is one of the best methods for estimating an algorithm's real-world performance.

When an analyst partitions her data, she faces a key question: How much data should be allocated to the training and test sets? Should, for example, seventy-five percent of subjects be designated the training set and twenty-five the test set? Or should there be a fifty-fifty split? If more data are given to the training set, an algorithm stands a better chance of learning predictively useful rules because it has a greater number and variety of subjects from which to learn. But, if the test dataset is too small, an analyst will get a poorer, less certain sense of how well the algorithm performs and how well its performance might generalize to data other than those on which it was trained.[112]

This decision can be guided by three considerations: the size of the entire starting dataset, the summary statistics of the outcome variable, and subject matter knowledge about the domain to which an algorithm is applied. Taking the first consideration, a larger dataset tends to weigh in favor of a more even split between training and test sets.[113] Obtaining a well-trained algorithm is key. And, if there are hundreds of thousands or millions of observations, it is unlikely (in most contexts) that, by allocating half to the test dataset, the training dataset would be deprived of examples from which the algorithm could learn. In other words, there is likely a great deal of redundancy in massive data sets. If, however, the dataset is substantially smaller, caution might be warranted; to ensure a well-trained algorithm, an analyst might consider randomly assigning a greater proportion of

[112] *See* WITTEN ET AL., *supra* note 103, at 164 ("There's a dilemma [in deciding how to partition data]: to find a good classifier, we want to use as much of the data as possible for training; to obtain a good error estimate, we want to use as much of it as possible for testing.").

[113] *Cf. id.* ("If lots of data are available, there is no problem: we take a large sample and use it for training; then another, independent large sample of different data and use it for testing."). Although this quote contemplates separately sampling the test data from the population of interest, instead of partitioning already sampled data, the thrust of the quote holds in the latter context; if there is a large amount of data, an analyst need not worry about ensuring a sufficient amount of training data at the expense of the size of the test data.

observations to the training set. There is no hard-and-fast mathematical rule dictating how data should be partitioned, but this somewhat qualitative accounting of dataset size is critical.[114]

Even if there are many observations in a dataset, though, the second consideration above — the outcome variable's summary statistics — could push an analyst to partition unevenly in favor of a larger training set. Particularly, this could result if certain outcomes are extremely rare.[115] Often, machine learning is applied to predict exactly these kinds of rare events, but evenly splitting a dataset into training and test data could risk few of these observations ending up in the training data. Imagine a dataset of five hundred thousand observations, of which only one percent, or five thousand, exhibit a particular outcome of interest. An analyst might want to ensure, by randomly sampling more than half of her data for training purposes, that her algorithm is capable of learning how to predict this rare outcome. Again, this is a judgment call with more than a bit of subjectivity.

Finally, an analyst can bring subject matter knowledge to bear. The balancing act performed when partitioning is weighing the algorithm's ability to learn predictive rules against the ability to assess the algorithm's performance on "unseen" data. If an analyst reasonably believes the system or domain she is researching to be relatively stable — meaning that causal processes in it do not change rapidly and dramatically — then she can be somewhat confident that the unseen, real-world data will be very similar to the data she collected.

---

[114] Although there is no rule applicable in every scenario, common splitting ratios for training to test data range from 70:30 to 80:20. *See, e.g.*, AMAZON WEB SERVICES, *supra* note 92, at 14 ("A common strategy is to take all available labeled data, and split it into training and evaluation subsets, usually with a ratio of 70-80 percent for training and 20-30 percent for evaluation."); SHAN SUTHAHARAN, MACHINE LEARNING MODELS AND ALGORITHMS FOR BIG DATA CLASSIFICATION: THINKING WITH EXAMPLES FOR EFFECTIVE LEARNING 196 (Ramesh Sharda & Stefan Voß eds., 2015) ("The commonly used ratio is 80:20 for training and test data sets, based on the Pareto principle.") (citation omitted). *Cf.* Michael Schrage, *AI Is Going to Change the 80/20 Rule*, HARV. BUS. REV. (Feb. 28, 2017), https://hbr.org/2017/02/ai-is-going-to-change-the-8020-rule [https://perma.cc/MMU3-RXUB] (discussing the Pareto principle, as well as how it may be changed by artificial intelligence).

[115] *Cf.* MURPHY, *supra* note 63, at 2 ("[E]ven when one has an apparently massive data set, the effective number of data points for certain cases of interest might be quite small. In fact, data across a variety of domains exhibits a property known as the long tail, which means that a few things . . . are very common, but most things are quite rare. . . . One consequence of the long tail is that understanding or predicting the behavior of most items requires learning from small amounts of data, even if the total amount of data is large.") (citation omitted) (emphasis omitted).

Therefore, she may not need to rely as heavily on assessing the algorithm's performance in the test data, allowing her to partition in favor of the training set. If, on the other hand, the analyst believes her system of interest to be relatively dynamic, a well-trained algorithm will be of little predictive value when deployed in the real world.[116] Therefore, she may lean more on assessing performance in the test data, partitioning less data to the training set, on the assumption or belief that some of the critical dynamism of the data will be captured in variation in the test data. This should, to some extent, simulate how the algorithm will perform in data notably different from those on which it was trained.

G. Model Selection

The problem has been defined. Data have been collected, cleaned, and partitioned. Summary statistics have been reviewed. But one thing is conspicuously absent — the algorithm. Nothing discussed so far has involved what is recognizable as a pre-training "algorithm." Now, however, an analyst turns to the algorithm by choosing a kind, or class, of model to implement.

Before we delve into how this choice is made, though, let us clarify how we are using both "choose" and "a kind of model," as we adopt a certain level of generality in regards to both. On the former, an analyst could choose an algorithm by simply implementing a pre-programmed algorithm distributed as a bundle of code in a piece of statistical software or a programming language.[117] Alternatively, she could

---

[116] *Cf.* BERK, *supra* note 65, at 336 ("[T]he forecasting tool developed [to predict which high school students are at risk of dropping out] would be applied to new cohorts of students that would likely differ from the training and test samples by more than random sampling error. One might well expect a gradual drift in the background of incoming freshmen and the mix of incentives to remain in school."). Taking this example, if it were the case that the initial dataset had been collected over multiple cohorts, meaning that some of this drift in background and incentives was captured across the training and test datasets, then the analyst might be motivated to partition a greater amount of data to the test dataset for the reason described in text.

[117] For examples of such packages available for R, a widely used statistical programming language, see BREIMAN ET AL., *supra* note 101 (detailing the implementation of the random forests algorithm); MEYER ET AL., *supra* note 101 (detailing the implementation of support vector machines); GREG RIDGEWAY, PACKAGE 'GBM': GENERALIZED BOOSTED REGRESSION MODELS (Mar. 21, 2017), https://cran.r-project.org/web/packages/gbm/gbm.pdf [https://perma.cc/UW79-3QTQ] (detailing the implementation of AdaBoost and stochastic gradient boosting). Similarly, the scikit-learn library for the Python programming language implements a variety of machine-learning algorithms. SCIKIT-LEARN, http://scikit-learn.org [https://perma.cc/8MDA-4EXF]. Implementing any of these "pre-programmed algorithms" will still require the

program her own algorithm that builds off of, or in some way modifies, the code and math of an existing kind of model. Our discussion encompasses both of these possibilities.

Take now "a kind of model." As mentioned in Part II, all machine-learning algorithms operate fundamentally by optimizing — that is, minimizing or maximizing — an objective function. We do not intend for "a kind of model" to refer to the specific mathematical expression in an objective function or to all mathematical details of how the algorithm optimizes that function is optimized. Rather, we intend models of the same "kind" to be those that share generally similar mechanisms for generating predictions.[118] For example, what makes random forests models all part of the same class is their construction of a "forest" of many classification or regression trees, followed by some sort of averaging of predictions across trees.[119] Different implementations of random forests algorithms can, say, optimize different objective functions[120] and construct different numbers of trees,[121] but, at base, they all are alike in their use of the forest. Similarly, despite the variety of ways to implement a neural network, we condense all such implementations into the "neural network" class of models — models that pass information between "layers" of digital neurons (also called nodes), which perform some mathematical processing.[122]

analyst to make critical choices about how they are "tuned" — the subject of the next section.

[118] We recognize that our standard for classes of models may be unsatisfactory, particularly for more technical readers. After all, algorithms' mechanisms for generating predictions can be "similar" or "dissimilar" along several dimensions. Furthermore, these dimensions operate at different levels of generality. For example, a random forests model could be deemed its own class because of how it uniquely employs classification and regression trees, but it could alternatively be lumped into a larger class of ensemble machine-learning techniques. Nonetheless, for the sake of simplicity and the lay reader, this is a useful heuristic.

[119] *See generally* Leo Breiman, *Random Forests*, 45 MACHINE LEARNING 5 (2001) (giving a description of random forest models and their construction) [hereinafter Breiman, *Random Forests*].

[120] *See* BERK, *supra* note 65, at 201 (describing how, when using random forests for classification, a "1-0 loss" function and the deviance of predicted probabilities are two possible loss functions).

[121] *See* BREIMAN ET AL., *supra* note 101, at 17-18 (stating that the "ntree" argument for the "randomForest" function allows the user to set the number of trees constructed).

[122] *See generally* MURPHY, *supra* note 63, at 565-72 (describing implementation in the neural network of models).

Selecting a model can be quite easy or quite complicated, depending on the context. Factoring into this selection are six considerations: the kind of output variable, the ability to implement an "asymmetric cost ratio," the ability to explain or offer reasons for the predictions, the potential for overfitting, the opportunities for tuning, and practical resource limitations. We will provide a brief overview of each, with an eye towards the choices an analyst has available.

Kind of Outcome Variable: The analyst's choices could be immediately and significantly narrowed by the type of outcome variable with which she is working. Many machine-learning algorithms are capable of predicting binary classes and continuous numeric variables.[123] But, although computer scientists and statisticians are continually refining existing algorithms, some algorithms have not yet been extended to estimate certain kinds of responses, or have been extended but in a way that does not preserve all of the desirable mathematical properties of the progenitor algorithm. For instance, variants of support vector machines, which was originally designed for binary classification, can be applied to multiple classification problems, but do not preserve some theoretical properties that analysts value.[124] Similarly, the original random forests algorithm does not support predicting ordinal responses,[125] and attempts to extend it have been unsatisfactory.[126] Therefore, if the outcome variable contains multiple classes or ordinal classes, an analyst may have fewer algorithms from which to choose.

[123] *See* Gary Ericson et al., *How to Choose Algorithms for Microsoft Azure Machine Learning*, MICROSOFT AZURE (Apr. 25, 2017), https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice [https://perma.cc/2C8L-FZSH] (showing overlap between, among other algorithms, decision forests, boosted decision trees, and neural networks in their abilities to engage in both classification and regression).

[124] *See generally* Ürün Doğan et al., *A Unified View on Multi-Class Support Vector Classification*, 17 J. MACHINE LEARNING RES. 1 (2016) (describing various extensions of support vector machines to multiple classification problems and their properties); Chih-Wei Hsu & Chih-Jen Lin, *Comparison of Methods for Multi-Class Support Vector Machines*, 13 IEEE TRANSACTIONS NEURAL NETWORKS 415 (2002) (offering an earlier analysis of issues similar to those taken up by Doğan et al.).

[125] Silka Janitza et al., *Random Forest for Ordinal Responses: Prediction and Variable Selection*, 96 COMPUTATIONAL STAT. & DATA ANALYSIS 57, 70 (2016) ("The use of the ordering in the levels of an ordinal response variable in tree construction is not supported by the classical [random forests] version of Breiman.") (citation omitted).

[126] *See, e.g.*, *id.* at 71 ("[O]rdinal regression trees are a reasonable alternative to classification trees if the response is ordinal. However, the differences were only small and their practical relevance is questionable.").

Opportunities for an "Asymmetric Cost Ratio": Another consideration, and one that interacts with the consideration of outcome variable type, is the ability to code in an "asymmetric cost ratio."[127] In any predictive endeavor, there are different kinds of errors, or incorrect predictions.[128] When estimating a binary outcome, there can be false positives (predicting something when that something is not the case) and false negatives (predicting *not* something when that something *is* the case). When estimating a continuous outcome, there can be overestimates and underestimates. Different kinds of errors are often viewed by stakeholders as having different normative valences; it is very rare for a stakeholder to view being wrong in one way as equally harmful as being wrong in the opposite way.[129] Thankfully, many,[130] but not all,[131] machine-learning algorithms permit an analyst to specify

[127] RICHARD A. BERK, STATISTICAL LEARNING FROM A REGRESSION PERSPECTIVE 253 (2d ed. 2016) ("[I]f for policy or subject matter reasons one needs to tune to approximate a target asymmetric cost ratio in a confusion table, model selection is in play once again.").

[128] *See generally* WITTEN ET AL., *supra* note 103, at 179-83 (describing the natures of different kinds of errors and how, in a general sense, machine-learning algorithms can be sensitive to asymmetric costs).

[129] *See, e.g.*, *id.* at 180 ("In truth, you'd be hard pressed to find an application in which the costs of different kinds of errors were the same."). For examples of machine-learning applications where consideration asymmetric costs was essential, see Berk et al., *supra* note 77, at 103-04 (describing how false negatives — predicting that arrestees will not commit future domestic violence when they actually do — were viewed by stakeholders as ten times more costly than false positives — predicting that arrestees will commit future domestic violence when they actually do not); Brian Kriegler & Richard A. Berk, *Small Area Estimation of the Homeless in Los Angeles: An Application of Cost-Sensitive Stochastic Gradient Boosting*, 4 ANNALS APPLIED STAT. 1234, 1241 (2010) (describing how stakeholders view underestimates — estimating that there are fewer homeless individuals in a given area than there actually are — as more costly than overestimates — estimating that there are more homeless individuals in a given area than there actually are).

[130] *See, e.g.*, BERK, *supra* note 65, at 210-13 (describing how to implement asymmetric costs with random forests); Kriegler & Berk, *supra* note 129 (implementing asymmetric costs with stochastic gradient boosting). *See generally* Zhi-Hua Zhou & Xu-Ying Liu, *Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem*, 18 IEEE TRANSACTIONS KNOWLEDGE & DATA ENGINEERING 63 (2006) (implementing asymmetric costs with neural networks).

[131] *See, e.g.*, BERK, *supra* note 65, at 325 ("A final difficulty with the software [implementing support vector machines in R] is that there is no direct way to address the relative costs of false negatives and false positives."). Since that book was written, researchers have proposed methods for creating truly cost-sensitive support vector machines. *See, e.g.*, Shuichi Katsumata & Akiko Takeda, *Robust Cost Sensitive Support Vector Machine*, 38 PROC. MACHINE LEARNING RES. 434 (2015); Hamed Masnadi-Shirazi et al., *Cost-Sensitive Support Vector Machines*, ARXIV:1212.0975V2 (Feb. 15, 2015), https://arxiv.org/pdf/1212.0975.pdf [https://perma.cc/A36E-G4ZH]. But these methods

an asymmetric cost ratio; the analyst imposes mathematical constraints on the algorithm that force it to make certain kinds of errors (say, false positives in the binary classification context or overestimates in the regression context) more often than others. For example, a cost ratio of 5:1 for false positives to false negatives would ensure that, for every false negative generated, there are five false positives. Thus, implementing asymmetric cost ratios is an extremely potent tool; it translates normative values of stakeholders into actionable math. Additionally, of the algorithms that can implement asymmetric costs, how exactly implementation occurs depends on the kind of output variable, with different means of implementation having different advantages for the explainability or interpretability of the predictions.[132]

Explainability: Increasingly, in the search for accountable and transparent algorithms, calls have been made for ways of understanding what is going on inside the black boxes.[133] It is often difficult to put into intuitive, understandable prose how exactly a machine-learning algorithm generates, for each subject, a prediction from all of the subject's input variable values.[134] But there are ways to make algorithms more explainable. For one, a simpler class of models can be chosen — one with a less complex optimization process. Recall, for instance, our discussion of random forests earlier in this section. That algorithmic class's general optimization process is the

have not yet been rigorously validated by other researchers and have not been incorporated into the code of widely used statistical software packages.

[132] *Cf.* BERK, *supra* note 65, at 266-71 (describing how, in stochastic gradient boosting, classification problems are treated as regression problems, estimating probabilities, with thresholds applied to convert probabilities to predicted classes). The consequence of this is that, if asymmetric costs are applied by changing these thresholds at the very end of the algorithm's mathematical processing, predictor importance and partial dependence plots will not reflect this asymmetric cost ratio. In other words, they will reveal input variables' importances and the functional forms of relationships between input variables and the output variable *assuming symmetric costs*.

[133] *See, e.g.*, Citron & Pasquale, *supra* note 4; Kroll et al., *supra* note 51; Rich, *supra* note 1; Andrew D. Selbst & Solon Barocas, Regulating Inscrutable Systems (June 7, 2017) (unpublished manuscript) (on file with authors). Additionally, the NYU School of Law hosted a conference earlier this year on obtaining explanations from machine-learning algorithms. *Algorithms and Explanations*, NYU LAW, http://www.law.nyu.edu/centers/ili/events/algorithms-and-explanations [https://perma.cc/UP4G-BDHG] (last visited Sept. 16, 2017).

[134] Perhaps a prime demonstration of this comes from Leo Breiman, the developer of random forests, who stated, "My biostatistician friends tell me, 'Doctors can interpret logistic regression.' There is no way they can interpret a black box containing fifty trees hooked together." Breiman, *Statistical Modeling*, *supra* note 9, at 209.

construction of a forest of classification or regression trees. An alternative could be to employ a neural network, which is generally considered to involve more complex, and less explainable, mathematical processes.[135] An interpretability-conscious analyst might elect a simpler kind of model, like random forests, with, of course, the understanding that a simpler model could often be a less accurate one. (Also, even the simpler model may not be interpretable as the same way as, say, a regression is.) Second, an analyst can generate graphical plots that indicate how important different input variables were to the predictions and how changes in the values of input variables tend to be translated into changes in the outcome variable.[136] But an analyst can currently generate these plots for only some machine-learning algorithms.[137]

Overfitting: Analysts are also concerned with overfitting when selecting a kind of model.[138] As mentioned when discussing summary statistics review, randomness can be a problem for techniques, like machine learning, that readily discover nuanced correlations in datasets. But some algorithms are less vulnerable to overfitting than others. Namely, some algorithms include a process called "bagging." Bagging algorithms, roughly speaking, make predictions on multiple, partially overlapping random samples of training data and then, for each subject, decide on a "final" prediction by averaging or otherwise reconciling the predictions that resulted for that subject in all of the

---

[135] *See, e.g.*, Viktoriya Krakovna & Finale Doshi-Velez, *Increasing the Interpretability of Recurrent Neural Networks Using Hidden Markov Models*, ARXIV:1606.05320V2, Sept. 30, 2016, at 46, https://arxiv.org/pdf/1606.05320.pdf [https://perma.cc/XJ6T-MYKD] ("[A]doption [of recurrent neural networks] has been slow in applications such as health care, where practitioners are reluctant to let an opaque expert system make crucial decisions."); Scott Wisdom et al., *Interpretable Recurrent Neural Networks Using Sequential Sparse Recovery*, ARXIV:1611.07252, Nov. 22, 2016, at 1, https://arxiv.org/pdf/1611.07252.pdf [https://perma.cc/TL24-M94R] ("Interpreting the learned features and outputs of machine learning models is problematic. This difficulty is especially significant for deep learning approaches [like neural networks], which are able to learn effective and useful function maps due to their high complexity.").

[136] These will be discussed in greater detail in Part III.B.

[137] *See* BERK, *supra* note 65, at 324 ("[I]t is not easy to learn from SVM how inputs are related to outputs. . . . There is also no direct help in determining predictor importance."); *id.* at 336 ("If one needs to examine response functions and evaluate predictor importance, an implementation of support vector machines may not have what is needed.").

[138] *See, e.g.*, *id.* at 42 ("A major difficulty in statistical learning is overfitting. Very flexible fitting procedures will tend to respond to idiosyncratic features of the data, producing results that do not generalize well to new data.").

samples in which it was contained.[139] As a result of this averaging or reconciliation, the algorithms are less likely to fit spurious relationships in training data.[140] So, if an analyst is operating in a highly dimensional context — one teeming with input variables — likely to give rise to overfitting in susceptible models, she might select an overfitting-resistant model.[141]

Opportunities for Tuning: Building machine-learning algorithms is not merely a point-and-click exercise; an analyst does not simply open up a computer program like Microsoft Excel, load in some data, and then find and click, say, the "Run Random Forests" button. Although models within a class may share a general optimization method, aspects of that method referred to as parameters can be controlled, or "tuned," by the user.[142] These parameters afford flexibility in several regards, which will be discussed in the next section.[143] But, as has been the theme of preceding paragraphs, algorithms differ. Some offer a greater diversity of tuning parameters, or a set of parameters that are more influential.[144] This could afford the analyst greater flexibility, which is yet another factor an analyst considers when selecting a kind of model.

Resource Limitations: While the factors laid out above have largely been mathematical ones, there are, of course, practical considerations. Machine-learning algorithms take processing power, time, and memory space to run, and more complex algorithms take more of

---

139 *See id.* at 169-74.

140 *See id.* at 169 ("Averaging over a collection of fitted values can help compensate for overfitting. That is, the averaging tends to cancel out results shaped by idiosyncratic features of the data. One can then obtain more stable fitted values and more honest assessments of how good the fit really is.").

141 Some statisticians even claim that certain bagging algorithms are not just resistant to overfitting, but immune from it. For example, Leo Breiman and Adele Cutler have claimed that random forests "does not overfit." Leo Breiman & Adele Cutler, RANDOM FORESTS, https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm (last visited Sept. 16, 2017). *Accord* Breiman, *Random Forests*, *supra* note 119, at 7 ("This result explains why random forests do not overfit as more trees are added, but produce a limiting value of the generalization error.").

142 *See* BERK, *supra* note 65, at 42 ("For all of the statistical learning procedures examined, there are choices to be made about 'tuning parameters.' . . . They are parameters, much like dials on a machine, that determine how a procedure functions.").

143 *See infra* Part II.H.1.

144 *See, e.g.*, BERK, *supra* note 65, at 308 ("[S]omewhat in contrast to random forests and stochastic gradient boosting, variation across a set of reasonable values for the tuning parameters can have a large impact on the results."); Ericson et al., *supra* note 123, at 5 ("The upside is that having many parameters typically indicates that an algorithm has greater flexibility. It can often achieve very good accuracy.").

these.[145] If, when deployed, the running model will generate predictions for a relatively small dataset on a single end-user's personal computer, then perhaps a more complex model could be selected. But the running models increasingly driving automation continuously generate predictions for millions of subjects.[146] In such scenarios, the developers of the algorithms perform internal cost-benefit analyses and select algorithms with an eye towards their suitability for the environments in which they will be deployed.

Finally, an astute reader may have noticed the conspicuous absence of a key word — accuracy — from this section. If there is one prime motivation for using machine learning, it is accuracy, and surely there must be some algorithms that are simply more accurate than others. Does this not factor into model selection? Although it is true that certain kinds of algorithms have a reputation for sometimes being more accurate than others,[147] in practice it is extremely difficult to predict at the outset.[148] Which class of model performs best is often a function of peculiarities of a given data set. Therefore, it is common for this model selection step to result not in the selection of just one kind of algorithm, but in the narrowing-down of an analyst's choices to a few candidates.[149] These candidates would then be taken through the next step — model training — to determine which performs optimally and, thus, should be deployed.

H. Model Training

With training data ready to go and a kind of model (or multiple candidates) selected, an analyst can now begin the learning part of machine learning. An algorithm is run on the training data set and, in

[145] *See* Ericson et al., *supra* note 123 (comparing machine-learning algorithms on the bases of their memory footprints and training times).

[146] For example, Amazon uses machine-learning systems "at incredible scale and speed" to generate video recommendations for its millions of customers. *SDE – Amazon Video Recommendations: Machine Learning and Distributed Systems*, AMAZON JOBS, https://www.amazon.jobs/en/jobs/546401 (last visited Sept. 16, 2017).

[147] *See* Ericson et al., *supra* note 123, at 8 (distinguishing some machine-learning algorithms as generally showing "excellent accuracy" and others as generally showing "good accuracy").

[148] *See* BERK, *supra* note 65, at 335 ("Random forests, boosting, and support vector machines can all perform well. It is not yet clear which perform better for which kinds of datasets, or even if the differences in performance are likely to matter a great deal in practice.").

[149] *See* WITTEN ET AL., *supra* note 103, at 172 ("We often need to compare two different learning schemes on the same problem to see which is the better one to use.").

the process, learns rules for predicting the outcome. "Model training" does not refer to a single, discrete instance of running an algorithm on training data. Rather, the process of training a model encompasses a few additional tasks: tuning, assessment, and feature selection. Training a machine-learning algorithm dances back and forth across these tasks, interspersing between them runs of the model on the training data.[150]

1. Tuning

Machine-learning algorithms possess tuning parameters — operational levers an analyst can pull to change the internal operations of the algorithms.[151] These can vary dramatically depending on the particular algorithm implemented, and this Article does not attempt to provide a comprehensive description of all tuning parameters. But, broadly speaking, machine-learning algorithms generally afford the user control over the following aspects of the learning process: the objective function;[152] the bias-variance tradeoff; the cost ratio; and assessment, or validation, methods.

As mentioned, all algorithms have a "goal," and this goal is defined in an objective function. This function is a mathematical expression of what should be optimized — either minimized or maximized — when generating predictive rules. Most frequently, the objective function can be fairly intuitively tied to conceptions of accuracy.[153] Broadly speaking, there are often multiple ways to mathematically represent accuracy, with different mathematical assumptions and tradeoffs

[150] *See* Pete Chapman et al., CRISP-DM 1.0, at 25 (2000), https://www.the-modeling-agency.com/crisp-dm.pdf [https://perma.cc/QAJ8-6HSP] ("According to the model assessment, revise parameter settings and tune them for the next run in the Build Model task. Iterate model building and assessment until you strongly believe that you have found the *best* model(s).") (emphasis in original).

[151] *See supra* note 142 and accompanying text.

[152] We recognize that many technical readers may be disinclined to view the objective function as a tuning parameter, and for good reason; the objective function plays a much more significant role, defining what is to be optimized, than do the other parameters. We have chosen to include it in this section simply to aid the flow of our discussion and because setting the objective function might, for a nontechnical reader, appear more closely linked conceptually to the setting of other tuning parameters.

[153] To name just two examples, the function could be a Gaussian loss function, representing the residuals between the predicted quantitative outcomes and the true outcomes. *See, e.g.*, BERK, *supra* note 65, at 270 (noting the Gaussian loss function as one possibility for stochastic gradient boosting). It could also be the homogeneity of outcomes in a cluster, or node, of observations predicted by the algorithm to all have the same outcome. *See, e.g.*, *id.* at 113-17 (describing homogeneity, or purity, in the context of classification and regression trees).

depending on the context;[154] it is up to a well-informed analyst to make an appropriate choice when tuning. Furthermore, as we will discuss in the next Part, the specification, or mathematical details, of the objective function offers opportunities to mitigate discriminatory outcomes of algorithms.[155]

In addition to the objective function, algorithms possess a number of tuning parameters that affect in some way what is known as the "bias-variance tradeoff."[156] This phrase refers to a common phenomenon in statistics — altering an algorithm to have less bias (essentially, how far away the predictions are, on average, from the truth)[157] increases the variance (essentially, how inconsistent predictions for a given subject's outcome would be if, hypothetically, the algorithm were retrained many times on different training datasets), and vice versa. Tuning parameters come into play in this tradeoff because they frequently affect the complexity and speed of an algorithm's learning, and more complex, slower learning tends to yield less bias but more variance.[158] An analyst possessing a massive data set might feel comfortable building a more complex, slower-learning

[154] *See id.* at 270-71 (describing different possible loss functions for stochastic gradient boosting and noting that implementing asymmetric costs could be accomplished via a Laplace loss function).

[155] *See infra* Part III.A.

[156] *See* ALPAYDIN, *supra* note 65, at 76-80; BERK, *supra* note 65, at 55-56.

[157] More rigorously, bias refers to how close the expected value of the predictions is to the true value of what is being predicted over repeated realizations of the data. *See* BERK, *supra* note 65, at 56. "Repeated realizations" refers to a hypothetical scenario in which the training data are re-generated many times and the algorithm is re-trained on each of those new training datasets. The expected value of something is the result of adding together all of that something's possible values or outcomes, each multiplied by its probability of occurring over repeated realizations. For example, imagine a coin-flipping game where the player flips a coin once; he or she wins two dollars if the coin lands on heads, but loses one dollar if it lands on tails. The expected value of this game, in dollars, can be calculated as: (probability that the coin lands on heads * 2) + (probability that the coin lands on tails * -1), or (0.5 * 2) + (0.5 * -1), which evaluates to $0.50. In other words, if a player were to play this game repeatedly ("repeated realizations" of the game), he or she could expect to win, on average, fifty cents.

[158] There are several parameters that can affect this for various algorithms. To give just a few examples, a lower terminal node size in random forests reduces bias; variance might be increased accordingly, although random forests' bagging might counteract some of this. BERK, *supra* note 65, at 235. Stochastic gradient boosting shares this tuning parameter, as well as another — the depth (e.g., two-way or three-way) of permitted interactions effects; higher order interaction effects can reduce bias but increase variance. *Id.* at 271-73. Finally, in neural networks, a larger number of hidden layers is likely to reduce bias, but increase variance, and vice versa. Stuart Geman et al., *Neural Networks and the Bias/Variance Dilemma*, 4 NEURAL COMPUTATION 1, 12 (1992).

algorithm, as having lots of data can itself tamp down on variance,[159] but an analyst with sparser data may tune for a simpler, faster-learning algorithm.

A third category of tuning parameters can be used to effect an asymmetric cost ratio.[160] There is great mathematical diversity in techniques used to implement asymmetric cost ratios, and cost ratios can even be introduced by changing the objective function's specification.[161] Real-world stakeholders rarely view different kinds of errors as holding the same normative valence. Therefore, analysts frequently rely on tuning parameters to implement these asymmetries.

Finally, tuning parameters can affect how an algorithm is evaluated and assessed. Evaluation and assessment will be the subject of the next subsection, but, as will be seen, algorithms can "self-evaluate" during training through a few processes, including one called cross-validation. Programmers can set these tuning parameters controlling how these processes operate.

2. Assessment

Rarely is an algorithm run once on a training data set, the programmer is entirely satisfied with its performance, and the algorithm is then immediately deployed. Instead, machine learning is a bit (or, sometimes, a lot) of trial and error.[162] After a tuned algorithm is run on training data, an analyst provisionally assesses its performance and often chooses to then re-tune the algorithm, re-train it, and re-asses it. Such a cycle can occur multiple times, and critical to the cycle are appropriate assessment methods.

We discussed earlier one method for assessing accuracy — using a trained algorithm to predict outcomes in a test dataset. But remember that the purpose of test data is to enable assessment of how the

---

[159] *Cf.* BERK, *supra* note 65, at 56 ("[L]arger samples in general provide estimates with a smaller variance.").

[160] To name a few such parameters, in random forests, for example, one can implicitly alter the prior distribution, engage in stratified bootstrap sampling when constructing each tree, or change the majority voting scheme for tree nodes to a voting scheme based on some threshold other than fifty percent. *Id.* at 211. Changing the prior distribution is also applicable in support vector machines. *Id.* at 321-22.

[161] *See supra* note 154 and accompanying text.

[162] *See* BERK, *supra* note 65, at 275 ("[O]ne has no choice but to experiment with different sets of tuning parameter values. Unfortunately, this can be at best a trial-and-error process with too often no definitive resolution."); WITTEN ET AL., *supra* note 103, at 295 ("As in many machine learning situations, trial and error using your own particular source of data is the final arbiter.").

algorithm performs on *unseen* data — that is, data that did not, in any way, factor into how the algorithm was trained. Therefore, the test dataset cannot be used for the kind of iterative assessment cycle described above; even though doing so would not cause the algorithm to "see" the test data in the same way it "sees" (i.e., directly analyzes and learns from) the training data, re-running an algorithm *after* observing its performance in a test dataset causes nearly the same statistical harm as evaluating an algorithm in the same dataset on which it was trained.[163] Therefore, analysts seek an alternative.

Multiple solutions exist that allow algorithms to estimate their own accuracies during the training process. In general, they involve further randomly sub-sectioning, or partitioning, the training data; running each iteration of an algorithm (e.g., each tree in a random forest) on a particular partition of the training data; and then assessing the algorithm's accuracy by predicting an outcome for each subject based only on the iterations that were run on a subsection of the data *not* including that subject — iterations that did not "see" that subject.[164] This paradigm is not quite as rigorous as using test data; although each iteration of an algorithm may be generating predictions only for subjects it has not seen, other iterations of the algorithm may have seen those subjects if the subsections of the training data were overlapping, which frequently occurs. Nevertheless, such assessment processes — of which a key procedure called cross-validation[165] is one — serve an invaluable function. They provide an actionable estimate of accuracy that can be used to refine an algorithm's training without falling into the trap of assessing the accuracy of an entire algorithm on the entirety of the training data. And, most importantly, they do so while preserving the sanctity of the test data.

Only after an analyst is completely satisfied with the algorithm as trained — following any cycles of re-tuning and re-assessment — does she turn to the test data. With a final model in hand, she can use it to predict outcomes in the test data, which should, subject to our earlier

---

[163] *See* WITTEN ET AL., *supra* note 103, at 164 ("It is important that the test data is not used *in any way* to create the classifier. For example, some learning schemes involve two stages, one to come up with a basic structure and the second to optimize parameters involved in that structure, and separate sets of data may be needed in the two stages. Or you might try out several learning schemes on the training data and then evaluate them — on a fresh dataset, of course — to see which one works best. But none of this data may be used to determine an estimate of the future error rate.") (emphasis in original).

[164] *See generally id.* at 167-72; SUTHAHARAN, *supra* note 114, at 183-96.

[165] *See supra* note 164.

qualifications, provide the best possible estimate of how accurately the algorithm will perform in the real world.

Legal scholars and policymakers must be aware of these differences in kinds of accuracy. If laws or rules mandate that accuracy testing be done before deploying a running model, they may have to specify the kind of accuracy rates that must be reported. After all, an accuracy rate measured in test data is generally more rigorous than one measured via a procedure like cross-validation, which in turn is more rigorous than one measured in the training data.

3. Feature Selection

One aspect of model training we have neglected so far is feature selection[166] — trimming down the algorithm's set of input variables during cycles of repeated training, assessment, and re-tuning.[167] Such simplification can be beneficial for multiple reasons. For one, analysts try to avoid the "curse of dimensionality."[168] This phrase refers to a multitude of statistical phenomena, but, at its base, includes the fact that the amount of training data an analyst needs increases exponentially with linear increases in the number of input variables. In other words, as an algorithm analyzes more input variables, an analyst needs an exorbitantly large amount of training data to cover all possible combinations of values across those variables and, thus, to obtain accurate predictions. Feature selection can help get around this curse.[169] Feature selection also reduces the risk of overfitting, as there is a lower chance of correlationally relevant noise in fewer variables.[170]

---

[166] We must be careful here about terminology. There are several "feature"-related steps in machine-learning workflows, but they carry different meanings, and how they are used can vary between machine learning practitioners. Importantly, we are not using "feature selection" here to mean the *creation* of inputs, often by obtaining some information from unstructured data or by transforming existing input variables. Such processes are more commonly referred to as "feature extraction" or "feature processing" or "feature construction" or "feature transformation" or "feature engineering." *See supra* note 92. Also, this section may be less applicable to some deep learning techniques, which often do not need an explicit feature selection step because its effect is achieved implicitly in deep learning's changing of how input features are represented during training.

[167] *See, e.g.*, ALPAYDIN, *supra* note 65, at 110-12; Severtson et al., *supra* note 92.

[168] *See, e.g.*, Jerome H. Friedman, *On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality*, 1 DATA MINING & KNOWLEDGE DISCOVERY 55, 65 (1997); Domingos, *supra* note 104, at 82-83.

[169] *See* Isabelle Guyon & André Elisseef, *An Introduction to Variable and Feature Selection*, 3 J. MACHINE LEARNING RES. 1157, 1158 (2003) ("There are many potential benefits of variable and feature selection: . . . defying the curse of dimensionality to improve prediction performance."); *see, e.g.*, ALPAYDIN, *supra* note 65, at 110-12

In practice, feature selection typically occurs within the iterative assessment-tuning cycles mentioned above.[171] This is because a critical goal of feature selection is not just to avoid the curse of dimensionality and reduce overfitting, but to do so without sacrificing accuracy; if too many variables are pruned, then predictive power starts decreasing.[172] So, an algorithm might be trained once, have its performance assessed via cross-validation, have its features trimmed, be re-trained, etc. This cycle continues until reductions in features start impinging on accuracy.

I. Model Deployment

Finally, the algorithm is ready to be deployed, converting it into a "running model." It is ready to start making predictions in the real world, predictions that will carry real consequences when forming the bases of decisions. There is great variation in how models are deployed, and, for the most part, this is not the domain of data scientists or statisticians, but rather of more conventional computer programmers and information technologists.[173] For these reasons, we do not devote much space to this stage.

One challenge that must be tackled at this stage is making the running model capable of running at scale. Many machine-learning algorithms when deployed are not run merely occasionally, but continuously; product recommendation systems, for example, can run in real-time, dynamically serving recommendations to customers as

(describing how feature selection is used for dimensionality reduction).

[170] *See* Yvan Saeys et al., *A Review of Feature Selection Techniques in Bioinformatics*, 23 BIOINFORMATICS 2507, 2507 (2007) ("The objectives of feature selection are manifold, the most important ones being: (a) to avoid overfitting and improve model performance . . . .").

[171] *See, e.g.*, ALPAYDIN, *supra* note 65, at 110-12 (describing the processes of either sequentially adding or removing input variables, retraining the model on the training data, and then assessing error in a validation set).

[172] *See id.* at 111 ("We stop [adding features in sequential forward selection] if adding any feature does not decrease [the error assessed in a validation set]."); *id.* at 112 ("We stop [removing features in sequential backward selection] if removing a feature does not decrease the error.").

[173] *Cf.* WITTEN ET AL., *supra* note 103, at 30 ("[Deploying a model] normally involves integrating it into a larger software system, so the model needs to be handed over to the project's software engineers. This is the stage where the implementation details of the modeling techniques matter. For example, to slot the model into the software system it may be necessary to reimplement it in a different programming language.").

they peruse items and fill up their carts.[174] This requires writing programs that continuously feed new data into the trained algorithm, as well as building back-end data infrastructure. Machine-learning algorithms running at scale may also be turned into *online* learning systems — systems in which the algorithms are regularly and automatically re-trained upon the collection of new data.[175] These similarly require ancillary programs and infrastructure.

Another challenge is making the algorithm end-user-friendly. Recall the hypothetical algorithm predicting whether new prisoners will be involved in violent altercations. The end user of that algorithm would likely be a prison warden who may have to, when encountered with a new prisoner, feed his or her data into the algorithm, obtain a prediction, and then make a housing decision based, at least in part, on that prediction. The warden would not be a data scientist and could not take those steps by working directly with the source code of the algorithm; he or she would need the algorithm to be packaged into some kind of user interface, and it would be the charge of programmers and technicians to create this interface and any supporting infrastructure.

Again, there is much more detail and nuance to how algorithms are operationalized. But ensuring it operates at scale; potentially online; and encased in a user interface, when necessary, help give rise to the running models — the models, like predictive policing and credit scoring algorithms, that have fixated many legal scholars.

III. APPLYING THE STAGES

How might legal scholars and policymakers use this taxonomy? Put simply, it paves the road for richer discussions about the harms and benefits of machine learning. We do not intend this work to fundamentally recalibrate scholarship about automated decision-making, but rather to fill the gaps we identified in Parts I and II. A common reason a legal scholar might take up machine learning in the first place is to bemoan its harms or to extoll its potential benefits,[176] but inattention to the technical details of machine learning renders

[174] *See, e.g.*, *Using Machine Learning on Compute Engine to Make Product Recommendations*, GOOGLE CLOUD PLATFORM (Feb. 14, 2017), https://cloud.google.com/solutions/recommendations-using-machine-learning-on-compute-engine.

[175] *See* Óscar Fontenla-Romero et al., *Online Machine Learning*, *in* EFFICIENCY & SCALABILITY METHODS FOR COMPUTATIONAL INTELLECT 27, 27-28 (Boris Igelnik ed., 2013).

[176] *See supra* Part I.

analyses incomplete. With our taxonomy, such scholars will be better placed to continue their important work with the appropriate rigor.

In this Part, we focus on three kinds of legal harms or benefits implicated by various statistical effects of machine learning, each corresponding roughly to the theme of a paper discussed in Part I. For each, we show how a deeper understanding of the machine-learning workflows enriches discussions of those harms and benefits. First, we take up discrimination, or disparate impacts. Second, we address the reason-giving potential of machine learning — the ability to explain the algorithm's decision, particularly in the context of the Fourth Amendment requirement of articulable suspicion. Finally, we turn to due process, with a focus on inaccuracy.[177]

A. Discrimination

If there is one aspect of algorithmic decision-making that has received the most attention from legal scholars, it is likely discrimination — or differential accuracies of an algorithm on different groups of individuals.[178] As a result, many of the algorithmic *sources* of disparate impact have already been surveyed in prior work; outcome variables can be disadvantageously defined,[179] data can be collected in a nonrepresentative manner,[180] data can have baked into them preexisting human biases,[181] and a particular set of input

---

[177] As Citron and Pasquale noted, the legal concept of due process encompasses more than just accuracy; its underlying values also include transparency, accountability, participation, and fairness. Citron & Pasquale, *supra* note 4, at 20 (citing Martin H. Redish & Lawrence C. Marshall, *Adjudicator, Independence, and the Values of Procedural Due Process*, 95 YALE L.J. 455, 478-89 (1986)). But it should be clear that many of these values — particularly fairness and accountability — will overlap significantly with our discussions of discrimination and reason-giving, respectively. Therefore, we have chosen to break off accuracy as our primary focus when considering due process. This choice is further buttressed by the degree to which legal tests for due process are tied to the accuracy or inaccuracy of a decision-making process, particularly in administrative contexts. *See* Cary Coglianese & David Lehr, *Regulating by Robot: Administrative Decision Making in the Machine-Learning Era*, 105 GEO. L.J. 1147, 1184-91 (2017).

[178] We avoid using the term "bias" to avoid confusion with the "bias" of the bias-variance tradeoff — a bias with a very specific technical definition. *See supra* note 157 and accompanying text.

[179] Barocas & Selbst, *supra* note 6, at 677-80.

[180] *See, e.g.*, *id.* at 684-87; Crawford, *supra* note 99.

[181] *See, e.g.*, Barocas & Selbst, *supra* note 6, at 681-84; Citron & Pasquale, *supra* note 4, at 13-16; Jeremy Kun, *Big Data Algorithms Can Discriminate, and It's not Clear What to Do About It*, CONVERSATION (Aug. 13, 2015), http://theconversation.com/big-data-algorithms-can-discriminate-and-its-not-clear-what-to-do-about-it-45849 [https://perma.cc/7R5N-

variables can be more predictive for one group than another.[182] We do not rehash all of those points here. That said, we do want to highlight one overlooked source of disparate impacts. And, more importantly, we will briefly describe the myriad statistical tools being developed to *counter* disparate impacts; as will be seen, implementing many of these requires intervening in the typically overlooked stages of machine learning.

Many of the sources of algorithmic discrimination described so far have been data-related. And this is for good reason; there is not much about the technical details of an algorithm that will, in and of itself, make it perform worse on certain groups than on others. The same math is being used to make predictions for everyone. Thus, what typically produces disparities in predictions is features of the underlying data on which an algorithm operates. That said, there is one aspect of how algorithms function that is important to keep in mind because, if left unchecked, it can facilitate the translation of data disparities into prediction disparities — overfitting. Previous scholarship has documented how data, particularly survey data, can often be noisier for minority groups than for others,[183] and an algorithm that overfits risks improperly capitalizing on this noise more so than an algorithm that does not overfit.[184] As a result, an overfitting algorithm could generate less accurate predictive rules for minority groups than for others. This is yet another reason why analysts should seriously consider the potential for overfitting when selecting a kind of model.

Turning to ways of countering disparate impacts, emerging statistical literature reveals just how harmful existing ignorance of machine learning's stages is. This literature has put forward several ways of mitigating algorithmic discrimination, and many require intervening during model tuning and model training. Toshihiro Kamishima and his co-authors, for instance, have proposed reducing disparate impacts during model tuning through a process called "regularization." In brief, this involves modifying an algorithm's objective or loss function to "punish" it if it makes highly disparate

KHH8].

[182] *See* Barocas & Selbst, *supra* note 6, at 688-90.

[183] *See, e.g.*, Joost Kappelhof, *Survey Research and the Quality of Survey Data Among Ethnic Minorities*, *in* TOTAL SURVEY ERROR IN PRACTICE 235, 242-43 (Paul P. Biemer et al. eds., 2017) ("Measurement error can seriously bias the accuracy of estimates and can differentially affect the responses of ethnic minorities.").

[184] *See supra* note 102 and accompanying text.

classifications.[185] And researchers from Google have shown that, during model training, one can change the threshold at which predicted probabilities are turned into classifications to force predictions to meet certain definitions of fairness.[186] Therefore, ignoring these stages of the machine-learning process means ignoring ways of reducing discrimination.[187] Furthermore, even though legal scholars have paid due attention to how discrimination can *result* during data collection, they have ignored novel methods for reducing it at that stage; research has shown that a certain version of discrimination can be reduced by, in essence, adding noise to the data.[188] In sum, reducing algorithmic discrimination is possible, but it requires intervening at several key, often overlooked, stages of machine learning.

B. Reason-Giving

The first, early wave of legal scholarship on machine learning and discrimination has led to a second trend, an onslaught of work on what is sometimes referred to as "explainability" — the ability of

[185] *See* Toshihiro Kamishima et al., *Fairness-Aware Classifier with Prejudice Remove Regularizer*, *in* MACHINE LEARNING & KNOWLEDGE DISCOVERY IN DATABASES 35, 35-50 (Peter A. Flach, Tijl De Bie & Nello Cristianini eds., 2012). The regularizer becomes larger when a class is predicted mainly on the basis of sensitive features in the data, so sensitive features become less influential in the final classifications. This regularizer is under the control of a shrinkage parameter, whose tuning controls the degree to which the algorithm is regularized.

[186] *See* Moritz Hardt et al., *Equality of Opportunity in Supervised Learning*, ARXIV:1610.02413, Oct. 11, 2016, at 8-10, https://arxiv.org/pdf/1610.02413.pdf [https://perma.cc/9D8U-3DBH].

[187] Another key theme of this body of technical literature is that there are different possible mathematical definitions of fairness, and that optimizing for one often precludes achieving another. *See* Richard Berk et al., *Fairness in Criminal Justice Risk Assessments: The State of the Art*, ARXIV:1703.09207V2, May 30, 2017, at 1, https://arxiv.org/pdf/1703.09207.pdf [https://perma.cc/3EGN-TDJM]; Jon Kleinberg et al., *Inherent Trade-Offs in the Fair Determination of Risk Scores*, ARXIV:1609.05807V2, Nov. 17, 2016, at 3, https://arxiv.org/pdf/1609.05807.pdf [https://perma.cc/6QVD-PZ93]. Thus, at the machine-learning stages we consider here, not only do analysts have to consider technical methods for achieving fairness, but they have to wrestle with highly normative questions of what kind of fairness matters most in a given context.

[188] *See* Michael Feldman et al., *Certifying and Removing Disparate Impact*, ARXIV:1412.3756V3, July 16, 2015, at 11-15, https://arxiv.org/pdf/1412.3756.pdf [https://perma.cc/48PZ-EW48]. This approach succeeds because it makes it less possible to predict an individual's protected class membership from values of their other input variables.

machine learning to give reasons for its estimations.[189] This is especially true among Fourth Amendment scholars, like Michael Rich, for whom the ability to offer articulable suspicion is paramount,[190] and European scholars[191] focused on the General Data Protection Regulation's right to "meaningful information about the logic involved"[192] in algorithmic decisions.

Almost all legal scholarship references machine learning as a "black box,"[193] and many authors state something like, "Even the programmers of an algorithm do not know how it makes its predictions."[194] The force of such rhetoric is to, if not explicitly state, then certainly imply that a key harm of machine learning is its lack of explainability — the inability of humans to interrogate an algorithm and say why it made the predictions it did. What has been missing from this early work, however, is a deep exploration of the different technical ways in which an algorithm could or could not give reasons and what methods exist for increasing explainability. Here we take a brief stab at exactly those questions. We find that, just as many of the

[189] *See, e.g.*, Jenna Burrell, *How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms*, 3 BIG DATA & SOC'Y 1, 1-2 (2016); Marco Tulio Ribeiro et al., *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*, PROC. 22ND ACM SIGKDD INT'L CONF. KNOWLEDGE DISCOVERY & DATA MINING 1135, 1135 (2016); Andrew Tutt, *An FDA for Algorithms*, 69 ADMIN. L. REV. 83, 101-11 (2017); Selbst & Barocas, *supra* note 133; *Algorithms and Explanations*, *supra* note 133; *2017 Papers*, FAT/ML (July 4, 2017), http://www.fatml.org/schedule/2017/page/papers-2017 [https://perma.cc/U2PF-MFXT] (listing papers focused on "interpretable," "explorable," and "transparent" algorithms). Note that we will use the terms "explainability" and "reason-giving" interchangeably.

[190] *See, e.g.*, Ferguson, *supra* note 1; Rich, *supra* note 1.

[191] *See, e.g.*, Bryce Goodman & Seth Flaxman, *European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation,"* ARXIV:1606.08813V3, Aug. 31, 2016, at 6-7, https://arxiv.org/pdf/1606.08813.pdf [https://perma.cc/2NEQ-JD9Y]; Lilian Edwards & Michael Veale, *Slave to the Algorithm? Why a 'Right to an Explanation' Is Probably Not the Remedy You Are Looking for*, 15 DUKE L. & TECH. REV. (forthcoming 2017), https://ssrn.com/abstract=2972855 [https://perma.cc/U9C7-Z3Q5].

[192] Council Regulation 2016/679 of Apr. 27, 2016, General Data Protection Regulation, art. 13-14, 2016 O.J. (L 119/1) 40, 40-42.

[193] *See, e.g.*, PASQUALE, *supra* note 5; Citron & Pasquale, *supra* note 4, at 6; Nicholson Price II, *Black-Box Medicine*, 28 HARV. J.L. & TECH. 419, 421 (2015); Rich, *supra* note 1, at 886; Andrea Roth, *Machine Testimony*, 126 YALE L.J. 1972, 1977 (2017).

[194] *See, e.g.*, Rich, *supra* note 1, at 886 ("[E]ven the original programmers of the algorithm have little idea exactly how or why the generated model creates accurate predictions.").

methods for mitigating disparate impact arise in the overlooked stages of machine learning, so do the methods for increasing explainability.

1. The Less Attainable and Useful Versions of Reason-Giving

We contend that there are four different ways in which a machine-learning algorithm can be explainable. Two of those merit slightly less attention, one for being essentially unattainable and the other for often being of little legal use.

The first, less attainable version is the ability to say, *for each individual or subject* on which a running model is deployed, exactly what about that particular individual or subject caused its prediction. In other words, it is the ability to say exactly how changes in a certain input variable's values for that individual or subject would have yielded a different prediction, holding all of the individual's other input variable values constant. For example, this level of reason-giving in a risk assessment algorithm might entail being able to say that a convicted criminal was predicted as committing a violent crime if released on probation *because of* his being male and twenty-one years old; if he had instead been, say, female and thirty-five years old, but everything else about him were the same, then he would have been predicted as not committing a violent crime while on probation. The problem with demanding this level of explainability, though, is that one can never legitimately ask such *ceteris paribus* questions.[195] One cannot ask how the hypothetical criminal's prediction would be different if he were instead a thirty-five year-old female, holding everything else about him constant, because it is not in fact possible to hold everything else constant outside of a true scientific experiment. If this male had instead been born fourteen years earlier and female, then every other attribute of her would by necessity be different from those same attributes of the twenty-one year-old male. Thus, asking for this kind of reason is rather futile.

On the other end of the explainability spectrum is asking not why each individual's predictions resulted, but why, in the most general sense, an algorithm makes predictions the way it does. An easy answer is that the algorithm's predictions result because making those predictions optimizes the algorithm's objective function. In other words, the form of the objective function — what was optimized —

[195] For a review of some of the philosophical debates over causation and counterfactuals, see generally John Collins et al., *Counterfactuals and Causation: History, Problems, and Prospects*, *in* CAUSATION & COUNTERFACTUALS 1 (John Collins, Ned Hall & L. A. Paul eds., 2004).

tells one all one needs to know about the inner workings of a machine-learning algorithm. It should be clear that, in many legal contexts, this is far from a satisfactory reason. By and large, legal demands for reasons and justifications in contexts like criminal justice[196] or financial services[197] would be left unfulfilled by such a superficial version of explainability — one that amounts to little more than "because the algorithm said so."

2. The More Attainable and Useful Versions of Reason-Giving

Asking an algorithm to explain why each individual prediction resulted is epistemologically problematic, and explaining the predictions generally by referencing an optimization process is unsatisfying in most legal contexts. Is there a happy medium? We think there is. In fact, there are roughly two different kinds of approaches to "peeking inside the black box" that are both attainable and informative. But taking advantage of them requires making choices at the model selection and model training stages.

One family of approaches fundamentally attempts to describe how important different input variables are to the resulting predictions. Within this family, some methods operate on a global, or "algorithm-wide," level; they do not ask how important certain input variables are to generating a prediction for a given individual on which a running model is deployed, but how important they were to the algorithm's accuracy during training across many individuals.[198] The output of such methods is known as a variable importance plot, which displays graphically the relative importances of the different input variables.[199] Similar plots have been developed even for "standard" neural networks,[200] but they have not yet been fully extended to stochastic

196 *Cf.* Rich, *supra* note 1, at 893-95 (discussing how, in the Fourth Amendment context, *individualized* suspicion is required — reasons must be given for why a particular individual was searched or seized).

197 *Cf.* Selbst & Barocas, *supra* note 133 at 5 (discussing the Fair Credit Reporting Act's and Equal Credit Opportunity Act's requirement for adverse action reports that offer individual-level explanations for why an individual was subject to an adverse financial decision).

198 *See* BERK, *supra* note 65, at 213-22 (describing such plots for the random forests algorithm); *id.* at 274 (describing such plots for stochastic gradient boosting, but with the caveat that, unlike random forests' plots, they are not constructed on the basis of out-of-bag observations, meaning that they do not as faithfully recount contributions of input variables to forecasting accuracy).

199 *See id.* at 218-21, 279, 284.

200 *See* G. David Garson, *Interpreting Neural-Network Connection Weights*, AI EXPERT, Apr. 1991, at 47-48; A. T. C. Goh, *Back-Propagation Neural Networks for*

gradient boosting and forms of "deep learning,"[201] like convolutional neural networks. The other kind of importance-measuring methods attempts to operate on the individual level, explaining what the most important variables were for a given individual's predictions.[202] But these methods are particularly novel and have yet to be thoroughly tested. Furthermore, they may not be extensible to more complex algorithms like deep neural networks.[203] Regardless of the kind of importance method one wishes to implement — global or individual — doing so takes place during model training, so this stage cannot be ignored by legal scholars calling for explanation. Furthermore, given that importance methods are differentially available for different kinds of algorithms, an analyst must also make informed choices at the model selection stage.[204]

Outside of reason-giving methods that focus on measuring variable importance, another kind of approach seeks to describe how increases or decreases in the various input variables translate to changes in the outcome variable.[205] In other words, these methods reveal the "functional form" of the relationship between an input variable and

Modeling Complex Systems, 9 ARTIFICIAL INTELLIGENCE ENGINEERING 143, 146-49 (1995) (building upon Garson's approach).

201 For a brief overview of deep learning, see WITTEN ET AL., *supra* note 103, at 464-66.

202 *See* Anupam Datta et al., *Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems*, 2016 IEEE SYMP. ON SECURITY & PRIVACY 598, 601, 608-09.

203 *Cf. id.* at 614 ("We have not considered situations where inputs do not have well understood semantics. . . . With the proliferation of immense processing power, complex machine learning models such as deep neural networks have become ubiquitous in these domains. Defining transparency and developing analysis techniques in such settings is important future work.").

204 Choosing the right kind of model is important not just to get one over the threshold of being able to use an importance-measuring method, but also to use particularly interpretable methods. Some methods implemented for some algorithms provide relatively interpretable importance measures. In a random forests algorithm applied to binary classification, for instance, an importance value of 0.2 for a given variable for the "True" class can be interpreted as meaning that, if that variable were excluded from the algorithm, the algorithm would make twenty percent more errors when predicting the "True" class. *See, e.g.*, BERK, *supra* note 65, at 214-22. But such easily interpretable numbers do not come with all importance-measuring methods; sometimes the resulting metrics are simply *relative* measures of importance, and other times there is no easy way to put into understandable prose what an importance value indicates. *See, e.g.*, Garson, *supra* note 200. Thus, wise selection of an algorithm can facilitate not just the use of any importance-measuring method, but the use of a particularly interpretable one.

205 *See, e.g.*, BERK, *supra* note 65, at 222-29 (describing such plots for the random forests algorithm); *id.* at 273-74 (describing such plots for stochastic gradient boosting, with some caveats about interpreting them).

the outcome variable. This can provide a useful way of intuitively understanding what correlations the algorithm is keying in on when making its predictions. More specifically, they produce plots — often called partial dependence or individual conditional expectation plots — that graph the outcome variable as a function of a given input variable.[206] Unfortunately, these methods are not always available. They have been implemented for certain machine-learning algorithms — such as random forests,[207] gradient boosting algorithms,[208] and "standard" neural networks[209] — but not for more complex methods of deep learning. Thus, just as for the variable importance methods, informed model selection is critical to enable explainability during model training.

C. Due Process

Ask commentators why there is so much "hype" surrounding machine learning, and the response will often be a variant of one word — accuracy.[210] Put simply, machine-learning algorithms perform at least as accurately as, and often significantly more accurately than, standard predictive techniques.[211] This advantage is especially valuable when predicting complex phenomena — criminality, financial risk, etc. — that often pose a challenge for less powerful techniques. But algorithms still make mistakes, and it is these mistakes that keep legal scholars up at night. In particular, scholars, like Citron and Pasquale,

206 *See, e.g.*, *id.* at 226-29, 277-92.

207 *See, e.g.*, *id.* at 222-29.

208 *See, e.g.*, *id.* at 273-74.

209 *See, e.g.*, Alex Goldstein et al., *Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation*, ARXIV:1309.6392V2, Mar. 20, 2014, at 1-12, https://arxiv.org/pdf/1309.6392.pdf [https://perma.cc/6XMK-PZ3S].

210 *Cf., e.g.*, ROB SCHAPIRE, COS 511: THEORETICAL MACHINE LEARNING: LECTURE #1, at 2 (Feb. 4, 2008), https://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0204.pdf [https://perma.cc/E58K-UG5R] ("What is the advantage of machine learning over direct programming? First, the results of using machine learning are often more accurate than what can be created through direct programming."); *Machine Learning: What It Is and Why It Matters*, SAS (July 4, 2017, 2:28 PM), https://www.sas.com/en_us/insights/analytics/machine-learning.html [https://perma.cc/T9WW-BM5L] ("[I]t's possible to quickly and automatically produce [machine-learning] models that can analyze bigger, more complex data and deliver faster, more accurate results — even on a very large scale."); Dorian Pyle & Cristina San Jose, *An Executive's Guide to Machine Learning*, MCKINSEY Q., June 2015, http://www.mckinsey.com/industries/high-tech/our-insights/an-executives-guide-to-machine-learning [https://perma.cc/CU9N-BWU8] ("As a result, [machine learning] can yield insights that human analysts do not see on their own and make predictions with ever-higher degrees of accuracy.").

211 *See supra* note 94.

focused on due process worry that an inaccurate algorithm will be the source of severe deprivations of process.[212]

If these scholars turn their attention to stages in the machine-learning process beyond data collection, they will discover not only new, worrisome sources of inaccuracy but, more importantly, where interventions can be made. Just as scholars talk about machine learning as a monolith, so they treat accuracy as a monolith. They tend to view accuracy or inaccuracy as the byproduct almost exclusively of data quality. Yes, data quality is critical to accuracy, but there are also opportunities to create a more or less accurate algorithm at later, non-data-related stages of the machine-learning process. Without recognizing these nuances, legal scholars cannot offer sensible prescriptions or regulations for ensuring algorithmic accuracy. We offer that analysis here.

1. Failure to Fit

As we discussed in Part III, analysts go to great lengths to assess how their algorithms will perform when confronted with real-world data; an algorithm that exhibits near-perfect accuracy in the training data is of no use if it falters when deployed. There are, roughly speaking, two kinds of reasons why an algorithm can falter. First, a model can simply fail to fit any data — training or test — well. In such a scenario, even if the training and test data were perfectly representative of real-world data, the model would be inaccurate when deployed. Second, an algorithm can fit its training and, perhaps, test data well, but fail to generalize and perform equally well in real-world data.[213] In this subsection, we take up the first possibility, demonstrating how such failure to fit can result from choices taken at different stages of machine learning. The subsequent subsection takes up failure to generalize.

Before a model is selected and trained, choices made when building a dataset can preclude the model from fitting the data well. More specifically, choices that result in a *noisy* dataset can preclude good fits.[214] Put simply, a noisy dataset is one in which much of the

---

[212] *See* Citron & Pasquale, *supra* note 4, at 19 ("Protections could draw insights from what one of has called 'technological due process' — procedures enduring that predictive algorithms live up to some standard of review and revision to ensure their fairness and accuracy.").

[213] For a discussion of generalizability, see *supra* note 97 and accompanying text.

[214] *Cf.* WITTEN ET AL., *supra* note 103, at 7 ("[O]ften, because of errors or *noise* in the data, misclassifications occur even on the data that is used to create the classifier.").

variation between subjects is spurious and not representative of the real variation that an algorithm should exploit to legitimately distinguish between subjects. If this noise is present in the training and test data, then the algorithm's accuracy will be limited in those data and, by necessity, the real world.

What causes a noisy dataset? Two sources stand out. For one, when collecting data, the analyst can inaccurately measure input and/or output variables. Imagine, for instance, that a desired input to an algorithm is the number of years of education an individual has received. Also imagine that the analyst has access to official education records accurately reflecting this information, as well as to surveys of individuals in which they self-report their years of education. If that survey contains other information an analyst would like to code into input variables, then an analyst might, simply for the sake of convenience, choose to obtain the education information from the self-report forms instead of having to also parse the official education records. But self-report data can be notoriously inaccurate,[215] so this reliance risks introducing measurement error — a kind of noise — into the data, which can in turn hinder the algorithm's ability to fit the data well.

In addition to introducing noise himself through measurement error, an analyst can fail to properly weed out "naturally occurring" randomness. As mentioned earlier, there is inherent randomness in virtually all data, and this randomness can be particularly problematic if, for some observations, it factors into the generation of outliers.[216] An astute analyst can catch and remove these outliers by reviewing summary statistics. A less thorough analyst who chooses to forgo this review and dive right into model selection and training risks overlooking this noise.

Noisy data can be a prime contributor to an algorithm's failure to fit; an algorithm is only as good as its data, after all.[217] But failure to fit can also be caused by choices taken when training — more

---

[215] *Cf.* David Chan, *So Why Ask Me? Are Self-Report Data Really That Bad?*, *in* STATISTICAL & METHODOLOGICAL MYTHS AND URBAN LEGENDS: DOCTRINE, VERITY AND FABLE IN THE ORGANIZATIONAL AND SOCIAL SCIENCES 309 (Charles E. Lance & Robert J. Vandenberg eds., 2009) (arguing that some concerns about self-report data are overblown, but acknowledging that those data can suffer from four different kinds of measurement error).

[216] *See supra* Part II.E.

[217] *See* BERK, *supra* note 65, at 338-39 ("[T]here is no substitute for good data. . . . One cannot count on statistical learning successfully coming to the rescue. Indeed, some forms of statistical learning are quite fragile and easily pulled off course by noisy data, let alone data that have systematic measurement error.").

particularly, when tuning — an algorithm. As we discussed in Part II.H.1, many tuning choices affect an algorithm's bias-variance tradeoff. If an analyst tunes certain parameters to optimize more for low variance than for low bias, she risks creating an algorithm that does not fit the data well. Of course, the reason that the bias-variance tradeoff is referred to as a tradeoff is because one cannot optimize for both low variance and low bias at the same time. Therefore, an analyst optimizing for low variance may be making a calculation that the risk of failure to generalize, described subsequently, is greater than the risk of failure to fit. If this calculation is correct, then any resulting failure to fit is not as concerning. But, if the calculation is wrong, then an inability to fit has been unnecessarily introduced. Thus, this points to the importance of an analyst putting careful thought into the relative risks of failure to fit and failure to generalize; the analyst should bring subject matter knowledge to bear to assess how different the real-world data are likely to be from the training and test data.

2. Failure to Generalize

It is quite difficult to recover from failure to fit; if an algorithm cannot predict accurately in its training and test data, then it is highly unlikely that the algorithm will predict accurately in the real world. But obtaining a well-fitting model is not the end of the road. If a well-fitting model cannot *generalize* and perform just as well on data it has not seen previously, then it will not perform well in the real world. Analysts must, therefore, ensure that a model is generalizable.

Just as noisy data can cause failure to fit, other data defects can cause failure to generalize. Particularly, if data are nonrepresentative of the population of interest, then generalizability is hampered;[218] the algorithm will have generated rules to predict the outcome in a very different group of individuals than that to which it is eventually applied. Because nonrepresentative data can result from nonrandom sampling from the population of interest, analysts should ideally engage in random sampling. In practice, however, analysts may not always have access to the true population of interest from which to randomly sample.

Imagine, for instance, that a company is building an algorithm to select whom to hire and that the company is on the cusp of changing its job advertising practices; it will soon post more on job boards than on industry blogs, which is where it had formerly targeted most of its advertising. The training and test data may comprise applicants hired

[218] *See supra* Part II.C.

in the past who had mostly visited industry blogs, whereas the population of interest when the algorithm is deployed will be individuals who mostly visit job boards. It is possible that those two kinds of applicants differ in important, job-relevant ways. Therefore, an algorithm developed to predict the job performance of one may not generalize when applied to the other. This is precisely the kind of problem an analyst must be attuned to. If he is, he might find ways of increasing generalizability; if he had data on, for instance, where each previously hired applicant had seen the job listing, he could engage in *nonrandom* sampling to preferentially include in the training and test data applicants who had seen the job listing on a job board. But, again, this requires careful attention and subject matter knowledge on the part of the analyst.

Outside of data-related issues arising during the early stages of our machine-learning workflows, three other issues arising at later stages can cause failure to generalize: succumbing to overfitting, favoring low bias in the bias-variance tradeoff, and improperly assessing accuracy during training. We take these three in turn.

As described in Part II, overfitting occurs when an algorithm capitalizes on idiosyncrasies in a dataset, generating rules to fit them. By virtue of these idiosyncrasies resulting from randomness, they will not be the same in the real-world data. Therefore, rules developed to fit them will not generalize to the real world; accuracy will be reduced. But certain machine-learning algorithms — namely, those that incorporate bagging — are not as vulnerable to overfitting.[219] Thus, during the model selection stage, analysts could often reduce the harm of failure to generalize by opting for a bagging algorithm.

When tuning the model, analysts must be mindful of the bias-variance tradeoff, just as we discussed in the context of failure to fit. Here, though, what is problematic is tipping the scales in favor of low bias over low variance; a less biased, but more variant, model could fail to generalize if there are significant differences between the training/test data and the real world. Again, this means that, to strike the best tradeoff, an analyst will have to be acutely aware of how likely such disparities are.

Finally, there is always the possibility that an algorithm could fail to generalize if an analyst deploys an algorithm without, during model training, bothering to probe how generalizable the algorithm seems. As detailed in Part II.H.2, a key step in model training is assessment — trying to get a sense of how the algorithm will likely perform in the

[219] *See supra* notes 137–41 and accompanying text.

real world. This process typically involves estimating accuracy either with some form of cross-validation or with test data. Although such testing is a paramount step of the machine-learning process, a sloppy analyst could skip that step, instead assessing accuracy in the training data themselves. This would lead to a skewed, overly optimistic sense of the algorithm's generalizability, perhaps causing the analyst to deploy an algorithm that will, in fact, fail to generalize well. Furthermore, even if the analyst does not skip this crucial step, choosing some form of cross-validation metric will be less rigorous and replicative of real-world generalization than using test data. Thus, an analyst worried about the potential for failure to generalize should, if possible, employ test data.

D. New Prescriptions

We believe that greater attention to the middle stages of machine learning will help legal scholars not only more accurately assess the harms and benefits of automated decision-making but also point them towards new prescriptions. When legal scholars focus on a limited subset of the processes of automated decision-making, as with some of the examples in Part I, all they can propose are solutions like correcting the garbage-in-garbage-out problem; they propose rooting out biases in input data.[220] Similarly, they often occupy themselves with fatalistically advocating legacy prescriptions, such as transparency and discriminatory impact tests,[221] that cannot fully address the underlying problems. To these scholars, the running model is a static, completed fact about the world, not the result of a living, breathing set of human processes, as we prefer to think of it.

By focusing on the many neglected middle stages of machine learning, legal scholars and policymakers will find creative new methods for detecting and ameliorating harm. For one, they might limit the use of certain kinds of models in particularly sensitive contexts. For example, if decision-making could lead to imprisonment or the loss of life, perhaps particularly unexplainable approaches such as convolutional neural networks should not be used. For another, if particular algorithms could be subject to disparate impact litigation, perhaps analysts should be required to keep records of whether and

[220] *See, e.g.*, Barocas & Selbst, *supra* note 6, at 677-88; Citron & Pasquale, *supra* note 4, at 13-16.

[221] *See, e.g.*, Andrew D. Selbst, *Disparate Impact in Big Data Policing*, 49 GA. L. REV. (forthcoming 2017), https://ssrn.com/abstract=2819182 [https://perma.cc/AC9M-GGQ8]).

how they implemented an asymmetric cost ratio, given that practice's potential to shift the balance of errors in ways that could produce disparate impacts. These are intentionally crude and stylized examples; we could provide better examples given a particular context and a particular set of machine-learning processes. But they at least suggest that new solutions might be possible.

Other oft-heard calls that deserve a second look in the light of playing with the data are for a "human in the loop."[222] Many think that the best way to ensure fairness or justice is to inject a human into the decision-making process, perhaps with the veto power to override the inanimate counterpart. We are worried that if we simply thrust the human at the output end of the running model, there is very little she can do to root out bias. The human becomes a rubber stamp for the machine, providing nothing more than a cosmetic reason to lull ourselves into feeling better about the results. There might be better, more productive roles for human oversight elsewhere in the process.

E. Is Machine Learning "More Art than Science"?

The more nuanced we become about the intricacies of the middle stages of machine learning, the better equipped we will be to counter the received wisdom that is so often repeated about machine learning. Consider an already well-worn aphorism — some aspects of machine learning are "more art than science."[223] This saying most often applies to the model selection and model training stages of our machine-learning workflows. For example, in response to a Consumer Financial Protection Bureau Request for Information on "alternative" (i.e., machine-learning based) credit modeling, Equifax addresses a question about decisions made during training by stating that "there is some art to building statistical models."[224] Because legal scholars have never adequately scrutinized what happens at these training phases, they tend to be too credulous to these kinds of claims. We wonder if we ought to be more skeptical. After all, our analysis reveals that, while there is some degree of trial-and-error in model selection and training, it is possible to parse out distinct considerations and approaches.

---

[222] *See supra* note 11.

[223] *See supra* note 10.

[224] Letter from Stephanie Gunselman, Dir. of Gov't Relations, Equifax, to Monica Jackson, Consumer Fin. Prot. Bureau (May 19, 2017), https://www.regulations.gov/document?D=CFPB-2017-0005-0085 [https://perma.cc/JKK4-8Z2H].

A detailed understanding of what it means to select or train a model might cause us to be less credulous of the "more art than science" claim. It might simply betray an immaturity in scientific understanding. At this early stage of development in machine learning, human intuition plays a prominent role in these processes only because we have not yet systematized our knowledge and processes in a way that will let us formalize, for example, a full accounting of the differences between various models. Over time, we imagine the boundary between "art" and "science" will continue to shift, with more processes becoming understandable science and fewer remaining inscrutable art.

CONCLUSION

Machine-learning algorithms are not the same kind of algorithms underlying a smartphone's calculator — ones that are merely digital instantiations of human logic. Nor are they magical running models spit out of a computer's mysterious depths. Instead, they are the complicated outputs of intense human labor — labor from data scientists, statisticians, analysts, and computer programmers. From the moment these humans conceptualize a predictive task to the moment the running model is deployed, they exert significant and articulable influence over everything from how the data are cleaned to how simple or complex the algorithm's learning process is. Along the way, they have the power to affect the running model's accuracy, explainability, and discrimination.

By pulling back this curtain, we have provided a new set of knowledge that we hope will be of great use to legal scholars and practitioners bravely taking on complex and unfamiliar technology. We hope that this Article will serve as a new primer on machine learning for these adventurers. And we hope that this Article will foster a shared vocabulary between lawyers and technologists. After all, collaboration will be key for tackling some of the most intractable problems at this new juncture.