tiny [table]font=tiny

archaeologyname=archaeology, description=Michel Foucault defines archaeology as a description that questions the already-said at the level of its existence. Alternately, archaeology describes discourses as practices specified in the element of the archive [@Foucault$_1972, 131$

Machine Learning: Archaeology of a Data Practice

Adrian Mackenzie

Preface

This book is not an ethnography although it has an ethnographic situation. If it has a field site, it lies close to the places where the writing was done – in universities, on campuses, in classrooms and online training courses (including MOOCs), and then amidst the books, documents, websites, software manuals and documentation, and a rather vast accumulation of scientific publications.

Readers familiar with textbooks in computer science and statistics can see the traces of this site in some typographic conventions drawn from the fields I write about. Important conventions include:

1. Typesetting the name of any actual machine learners, code or devices that do machine learning, and datasets on which machine learners operate in a san serif font like `this`;

2. Presenting formulae, functions, and equations using the bristling indexicality of mathematical typography

Why emulate the apparatus of science and engineering publication in this way? Social science and humanities researchers, even when they are observant participants in their field sites, rarely experience a coincidence between their own writing practices and those of research subjects. The object of study in this book is a knowledge practice that documents itself in code, equations, diagrams and statements circulated in articles, books and various online formats (blogs, wikis, software repositories).

A dense feedback signal runs through the many propositions, formulations, diagrams, equations, citations and images in this book. I've been writing code for years (

Cutting code: software and sociality **nodate**). Writing code was nearly always something distant from writing about code. Recent developments in ways of analysing and publishing scientific data bring coding and writing closer together, such that implementing things can be done almost in the same space as writing about those things. This looping between writing code and writing about code brings about sometimes generative, sometimes frustrating, encounters with various scientific knowledge (mathematics, statistics, computer science), with infrastructures on many scales (ranging across networks, databases here and there, hardware and platforms of various kinds, as well as interfaces) and many domains. At many points in researching the book, I digressed a long way into quite technical domains of statistical inference, probability theory, linear algebra, dynamic models as well as database design and data standards. Much of the code I've written in implementing machine learning models or in reconstructing certain data practices does not appear in this text, just as not all of the words I've written in trying to construct arguments or think about data practices has been included. Much has been cut away and left on the ground (although the `git` repository of the book preserves many traces of the writing and code; see https://github.com/datapractice/machinelearning). So, like the recipe books, cookbooks, how-tos, tutorials and other documentations I have read, the code, the graphics and the prose have been tidied here. Many exploratory forays are lost and almost forgotten. Nevertheless, the several years I have spent writing about data practice has felt substantially different to any other project by virtue of a strong coupling between code in text, and text in code. Practically, this is made possible by working on code and text within the same file, in the same text editor. Switching between writing `R` and Python code (about which I say more below) to retrieve data, to transform it, to produce graphics, to construct models or some kind of graphic image, and within the same file be writing academic prose, might be one way to write about machine learning as a data practice.

The capacity to mix text, code and images depends on an ensemble of software tools that differ somewhat from the typical social scientist or humanities researchers' software toolkit of word processor, bibliographic software, image editor and web browser. In particular, it relies on software packages in the `R` programming language programming languages!as mode of writing

such as the '`knitr`' (

knitr **nodate**

; New Tools for Reproducible Research with R **nodate**) and in python, the `ipython` notebook environment (

IPython **nodate**). Both have been developed by scientists and statisticians in the name of 'reproducible research.' Many examples of this form of writing can be found on the web: see IPython Notebook Viewer for a sample of these. These packages are designed to allow a combination of code written in `R`, python or other programming languages, scientific text (including mathematical formula) and images to be included, and importantly, executed together. In order to do this, they typically combine some form of text formatting or 'markup,' that ranges from very simple formatting conventions (for instance, the 'Markdown' format used in this book is much less complicated than HTML, and uses markup conventions readable as plain text and modelled on email (

Markdown **nodate**);) to the highly technical (LaTeX, the de-facto scientific publishing format or 'document preparation system' (

Document Preparation System **nodate**) elements of which are also used here to convey mathematical expressions). They add to that blocks of code and inline code fragments that are executed as the text is formatted in order to produce results that are shown in the text or inserted as figures in the text.[1]

1. There are a few different ways of weaving together text, computation and images together. Each suffers from different limitations. In `ipython`, a scientific computing platform dating from 2005 (

IPython **nodate**) and used across a range of scientific settings, interactive visualization and plotting, as well as access to operating system functions are brought together in a `Python` programming environment. Especially in using the `ipython` notebook, where editing text and editing code is all done in the same window, and the results of changes to code can be seen immediately, practices of working with data can be directly woven together with writing about practice. By contrast, `knitr` generates documents by combining text passages and the results (graphs, calculations, tabulations of data) of code interleaved between the text into one output document. When `knitr` runs, it executes the code and inserts the results (calculations, text, images) in the flow of text. Practically, this means that the text editor used to write code and text, remains somewhat separate from the software that executes the code. By contrast, `ipython` combines text and `Python` code

In making use of the equipment created by the people I study, I've attempted to bring the writing of code and writing about code-like operations into proximity. Does proximity or mixing of writing code and writing words make a practical difference to an account of practice? If recent theories of code and software as forms of speech, expression or performative utterance are right (

Speaking Code **nodate**

; *Coding freedom* **nodate**), it should. Weaving code through writing in one domain of contemporary technical practice, machine learning, might be one way of keeping multiple practices present, developing a concrete sense of abstraction and allowing an affective expansion in relation to machines.

---

more continuously, but at the cost of editing and writing code and text in a browser window. Most of the conveniences and affordances of text editing software is lost. While `ipython` focuses on interactive computation, `knitr` focuses on bringing together scientific document formatting and computation. From the perspective of praxiography, given that both can include code written in other languages (that is, python code can be processed by `knitr`, and `R` code executed in `ipython`), the differences are not crucially important [^P3]. This whole book could have been written using just Python, since Python is a popular general purpose programming language, and many statistical, machine learning and data analysis libraries have been written for Python. Widely used Python modules such as NumPy, SciPy, Scikit-learn, open-cv or Pandas allow anything done in `R` to be done in Python. It is difficult to generalise about the differences between the two programming languages. I have used both, sometimes to highlight tensions between the somewhat more research-oriented `R` and the more practical applications typical of Python, and sometimes because code in one language is more easily understood than the other.

Acknowledgments

My Texas Instruments TI-97 scientific calculator conveyed in high school mathematics classes something of correlation. There I first glimpsed the idea of fitting a line to points without using just a pencil and a ruler to work out the best fit. I also would like to thank my mathematics teachers at Wollongong High for the simple pleasures of numerical optimization they first introduced to me. Newton's method for finding the minimum value of a continuous function still operates in machine learning.

Decades later, from 2007-2012, I benefited greatly from a research position in the UK Economic and Social Research Council-funded Centre for Economic and Social Aspects of Genomics at Lancaster University. Certain colleagues there, initially in the Sociomics Core Facility, participated in the inception of this book. Ruth McNally first of all, but also Paul Oldham, Maureen McNeil, Richard Tutton, and Brian Wynne were participants in many discussions concerning the transformation of life sciences around which my interest in machine learning first crystallised. Various academic staff in the Department of Applied Mathematics and Statistics at Lancaster University shepherded me through their post-graduate training courses: Brian Francis for his course of 'Data Mining,' David Lucy for his course on 'Bayesian Statistics', Thomas Jakl for his course 'Genomic Data Analysis' and TBA's course on 'Missing Data.' My colleagues in science studies at Lancaster, especially Maggie Mort, Lucy Suchman, and Claire Waterton have . I have been very fortunate to have worked with inspiring and adventurous doctoral students at Lancaster during the writing of this book. Lara Houston, Mette Kragh Furbo, Felipe Raglianti, Emils Kilis, Xaroula Charalampia, and Nina Ellis have all helped and indeed challenged me in different ways. Sjoerd Bollebakker very kindly updated many of the scientific literature searches towards the end of the book's writing.

Contents

List of Figures

List of Tables

Mitchell, Tom

machine
learner!computer
program as

Breiman, Leo

learning!from experience

Chapter 1

Introduction: Into the Data

> *Definition*: A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, improves with experience E (
>
> *Machine learning* **nodate**, 2).

> In the past fifteen years, the growth in algorithmic modeling applications and methodology has been rapid. It has occurred largely outside statistics in a new community—often called machine learning—that is mostly young computer scientists. The advances, particularly over the last five years, have been startling (
>
> Statistical modeling **nodate**, 200)

> The key question isn't 'How much will be automated?' It's how we'll conceive of whatever *can't* be automated at a given time. (
>
> *Who owns the future?* **nodate**, 77)

A relatively new field of scientific-engineering devices said to 'learn from experience' has become operational in the last three decades. Known by various names –

code!machine learning as

machine learner!National Security Agency Skynet

machine learner!logistic regression

data science!relation to machine learning

Google Search

Apple Siri

Facebook!news feed

machine learning, pattern recognition, knowledge discovery, data mining – the field and its devices, which all take shape as computer programs or code, seem to have quickly spread across scientific disciplines, business and commercial settings, industry, engineering, media, entertainment and government. Heavily dependent on computation, they are found in breast cancer research, in autonomous vehicles, in insurance risk modelling, in credit transaction processing, in computer gaming, in face and handwriting recognition systems, in astronomy, advanced prosthetics, ornithology, finance, surveillance (see the U.S. Government's `SkyNet` for one example of a machine learning surveillance system (

SKYNET **nodate**)) or robots(see a Google robotic arm farm learning to sort drawers of office equipment such as staplers, pens, erasers and paperclips (

Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection **nodate**). Sometimes machine learning devices are understood as *scientific models*, and sometimes they are understood as *operational algorithms*. In very many scientific fields, publications mention or describe these techniques as part of their analysis of some experimental or observational data (as in the logistic regression classification models found in a huge number of biomedical papers). They anchor the field of 'data science' (

Doing data science **nodate**), as institutionalised in several hundred data science institutes scattered worldwide. Not so recently, they became mundane mechanisms deeply embedded in other systems or gadgets (as in the decision tree models used in some computer game consoles to recognise gestures or the neural networks used to recognise voice commands by search engine services such as `Google Search` and `Apple Siri` (

How Google Retooled Android With Help From Your Brain **nodate**)). In platforms setting, they operate behind the scenes as part of the everyday functioning of services ranging from player ranking in online games to border control face recognition, from credit scores to news feeds on Facebook . Machine learning is said to transform the

Google!Google Trends

artificial intelligence

data mining

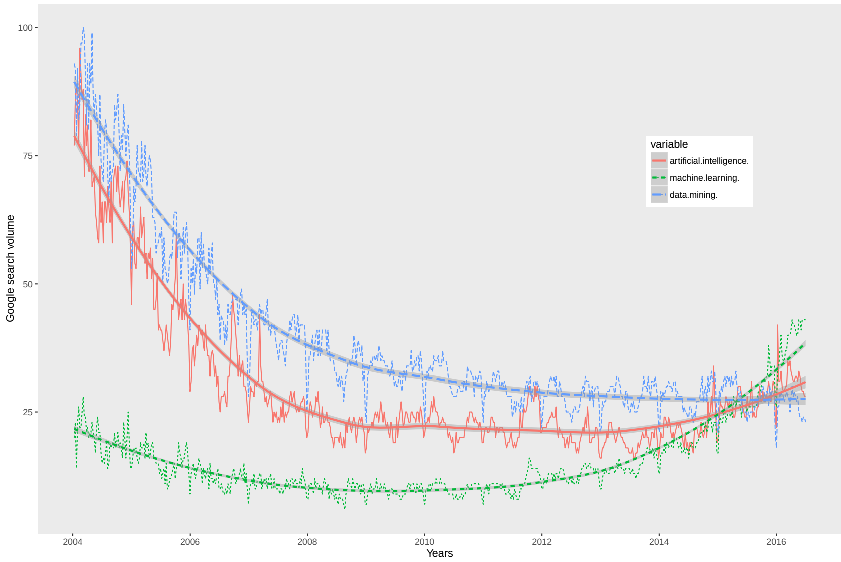

Figure 1.1: Google Trends search volume for ‘machine learning‘ and related query terms in English, globally 2004-2015

nature of knowledge, but might it transform the practice of critical thought? This book is an experiment in such practice.

Three accumulations: settings, data and devices

Three different accumulations stratify in machine learning: settings, data and devices. The volume and geography of searches on Google Search provides some evidence of the settings or sites doing machine learning. If we search for terms such as `artificial intelligence`, `machine learning` and `data mining` on the Google Trends service , the results for the last decade or so suggest shifting interest in these topics.

In Figure 1.1, two general search terms that had a very high search volume in 2004 – ‘artificial intelligence’ and ‘data mining’ – slowly decline over the years before starting to increase again in the last few years. By contrast, `machine learning` loses volume until around 2008, and then gradually rises again so that by mid-2016 it exceeds the long-standing interests in data-mining and artificial intelligence. Whatever the difficulties in understanding GoogleTrends results, these curves suggest an accumulation!of settings

Facebook!AI-Flow

machine learner!Skynet

data!plenitude

data!practice

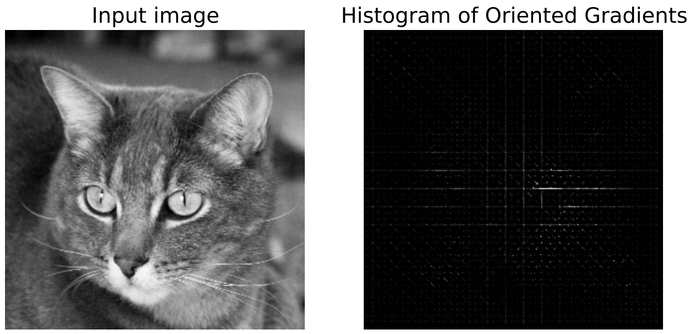

Figure 1.2: Close up of cat. The image on the left is already signal-processed as a JPEG format file. The image on the right is further processed using histogram of oriented gradients (HOG) edge detection. `kittydar` models HOG features. Cat photo courtesy photos-public-domain.com

accumulation of sites and settings turning towards machine learning.[1] What does it mean that machine learning surfaces in so many different places, from fMRIs to Facebook's `AI-Flow` (**Facebook_2016**), , from fisheries management to Al Queda courier network monitoring by `SkyNet`?[2]

A second accumulation concerns the plenitude of things in the world as data. . If we wanted to describe the general horizon of machine learning as a data practice in its specificity, we might turn to cats. Cat images accumulate on websites and

1. In the plot (Figure 1.1, the weekly variations in search volume on Google give rise to many spikes in the data. These spikes can be linked to specific events such as significant press releases, public debates, media attention and film releases. It is hard to know who is doing these searches. The data provided by Google Trends includes geography, and it would be interesting to compare the geographies of interest in the different terms over time.

2. The diagram shown in Figure 1.1 actually draws two lines for each trend. The 'raw' weekly GoogleTrends data – definitely not raw data, as it has been normalized to a percentage (

"Raw Data" is an Oxymoron **nodate**) – appears in the very spiky lines, but a much smoother line shows the general trend. This smoothing line is the work of a statistical model – a local regression or loess model (

Local regression models **nodate**) developed in the late 1970s. The line depends on intensive computation and models (linear regression, k nearest neighbours,). The smoother lines make the spiky weekly search counts supplied by Google much easier to see. They construct alignments in the data by replacing the irregular variations with a curve that unequivocally runs through time with greater regularity. The smoothed lines shade the diagram with a predictive pattern. The lineaments of machine learning already appear in such lines. How have things been arranged so that smooth lines run through an accumulated archive of search data?

social media platforms as de-centred, highly repetitive data forms. Like the billions of search engine queries, email messages, tweets, or much contemporary scientific data (e.g. the DNA sequence data discussed in chapter ??), images accumulate in archives of communication. Take the case of `kittydar,` a machine learner in the area of image recognition (see kittydar): 'Kittydar is short for kitty radar. Kittydar takes an image (canvas) and tells you the locations of all the cats in the image' (

harthur/kittydar **nodate**) . This playful piece of code demonstrates how machine learning can be amidst mundane accumulation. Heather Arthur, who developed `kittydar` writes:

image recognition
kittydar
Arthur, Heather
data!image as
data!vector
data!vector
machine learner!neural net

> Kittydar first chops the image up into many "windows" to test for the presence of a cat head. For each window, kittydar first extracts more tractable data from the image's data. Namely, it computes the Histogram of Orient Gradients descriptor of the image … This data describes the directions of the edges in the image (where the image changes from light to dark and vice versa) and what strength they are. This data is a vector of numbers that is then fed into a neural network … which gives a number from 0 to 1 on how likely the histogram data represents a cat. The neural network … has been pre-trained with thousands of photos of cat heads and their histograms, as well as thousands of non-cats. See the repo for the node training scripts (
>
> postnote
>
> harthur/kittydar **nodate**). a

This toy device finds cat heads in digital photographs, but also exemplifies many key traits of machine learning. Large accumulations of things become in a dataset. A dataset is used to train a typical machine learning device, a neural net, and the neural net *classifies* subsequent images probabilistically. The code for all this is 'in

code!circulation of
gradient|see gradient descent
data!training
classification
neural network|see machine learner!neural net
support vector machine|see machine learner!support vector machine
linear regression
machine learner!logistic regression
machine learner!neural net
machine learner!linear discriminant analysis

the repo.' Based on how it locates cats, we can begin to imagine similar pattern recognition techniques in use in self-driving cars (

Stanley **nodate**), border control facial recognition systems, military robots or wherever something seen implies something to do.

Faced with the immense accumulation of cat images on the internet, `kittydar` can do very little. It only detects the presence of cats that face forward. And it sometimes classifies people as cats. As Arthur's description suggests, the software finds cats by cutting the images into smaller windows. For each window, it measures a set of gradients – a spatial order of great significance in machine learning - running from light and dark, and then compares these measurements to the gradients of known cat images (the so-called 'training data'). The work of classification according to the simple categories of 'cat' or 'not cat' is given either to a neural network (as discussed in chapter **??**, a typical machine learning technique and one that has recently been heavily developed by researchers at Google (

Building high-level features using large scale unsupervised learning **nodate**), themselves working on images of cats among other things taken from Youtube videos (

Google 'brain' machine spots cats **nodate**), or to a support vector machine (a technique first developed in the 1990s by researchers working at IBM; see chapter **??**).

A final accumulation comprises machine learning techniques and devices. Machine learning range from the mundane to the esoteric, from code miniatures such as `kittydar` to the infrastructural sublime of computational clouds and clusters twirling in internet data-streams. Like the images that `kittydar` classifies, the names of machine learning techniques and devices proliferate and accumulate in textbooks, instructional courses, website tutorials, software libraries and code listings: linear regression, logistic regression, neural networks, linear discriminant analysis, support vector machines, k-means clustering, decision treesdecision tree|see {machine learner!decision tree}, k nearest neighbours, random forests, principal component analysis, or naive Bayes classifier to name just some of the most commonly used. Sometimes they have proper names: `RF-ACE`, `Le-Net5`, or `C4.5`. These names refer to predictive models and to computational algorithms of various ilk and provenance. Intricate data practices – normalization, regularization, cross-validation, feature engineering, feature selection, optimisation – embroider datasets into shapes they can recognise. The techniques, algorithms and models are not necessarily startling new or novel. They take shape against a background of more than a century of work in mathematics, statistics, computer science as well as disparate scientific fields ranging from anthropology to zoology. Mathematical constructs drawn from linear algebra, differential calculus, numerical optimization and probability theory pervade practice in the field. Machine learning itself amounts to a concrescing accumulation.

support vector machine
k-means clustering
\
nearest neighbours|see machine learner!_k_ nearest neighbours
principal component analysis
machine learner!Naive Bayes
data!practices
mathematics
machine learner|textbf
positivity!of knowledge

Who or what is a machine learner?

I am focusing on machine learners – a term that refers to both humans and machines throughout this book – situated amidst these three accumulations of settings, data and devices. While it is not always possible to disentangle machine learners from the databases, infrastructures, platforms or interfaces they work through, I will argue that data practices associated with machine learning delimit a of knowing. The term 'positivity' comes from Michel Foucault's *The Archaeology of Knowledge* (

The archaeology of knowledge and the discourse on language (Trans **nodate**), and refers to specific forms of accumulation of grouped in a discursive practice. Foucault attributes a lift-off effect to positivity:

> The moment at which a discursive practice achieves individuality and autonomy, the moment therefore at which a single system for the formation of statements is put into operation, or the moment at which

positivity!threshold of
\
generalization
operational formation
code!writing of

this system is transformed, might be called the threshold of positivity.

(

postnote

The archaeology of knowledge and the discourse on language (Trans

nodate, 186)Foucault, Michel!

Machine learners today circulate into domains that lie far afield of the eugenic and psychology laboratories, industrial research institutes or specialized engineering settings in which they first took shape (in some cases, such as the linear regression model or principal component analysis, more than a century ago; in others such as support vector machines or random forests, in the last two decades). If they are not exactly new and have diverse genealogies, the question is: what happen as machine learners shift from localized mathematical or engineering techniques to an everyday device that can be generalized to locate cats in digital images, the Higgs boson in particle physics experiments or fraudulent credit card transactions? Does the somewhat unruly generalization of machine learning across different epistemic, economic, institutional settings – the pronounced uptick shown in Figure 1.1 – attest to a re-definition of knowledge, decision and control, a new operational formation in which 'a system is transformed'?

Algorithmic control to the machine learners?

Written in code, machine learners operate as programs or computational processes. Machine learning can also be viewed as a change in how programs or the code that controls computer operations are developed and operate (see chapter **??** for more detailed discussion of this}. The term 'learning' in machine learning points to this change and many machine learners emphasize it. Pedro Domingos, for instance, a computer scientist at the University of Washington, writes:

> Learning algorithms – also known as learners – are algorithms that make other algorithms. With machine learning, computers write their own programs, so we don't have to.(
>
> *The Master Algorithm* **nodate**, 6)Domingos, Pedro|on machine learning]

\
control!crisis of
Beniger, James
machine learner!\textttkittydar
facial recognition

Viewed from the perspective of control, and how control is practiced, digital computer programs stem from and epitomise the 'control revolution' (

The control revolution **nodate**) that arguably has, since the late nineteenth century, programmatically re-configured production, distribution, consumption, and bureaucracy by tabulating, calculating and increasingly communicating events and operations. With the growth of digital communication networks in the form of the internet, the late 20th century entered a new crisis of control, no longer centred on logistic acceleration but on communication and knowledge. . Almost all accounts of the operational power of machine learning emphasise its power to inflect the control of processes of communication – border flows, credit fraud, spam email, financial market prices, cancer diagnosis, targetted online adverts – processes whose unruly or transient multiplicity otherwise evades or overwhelm us – with knowledge (classifications and predictions in particular) derived from algorithms that make other algorithms. On this view, `kittydar` can isolate cats amidst the excessive accumulation of images on the internet because neural net learning algorithms (back-propagation, gradient descent) have written a program – 'a pre-trained' neural net – during its training phase.

If a newly programmatic field of knowledge-control takes shape around machine learning, how would we distinguish it from computation more generally? Recent critical research on algorithms offers one lead. In a study of border control systems, which often use machine learners to do profiling and facial recognition , Louise Amoore advocates attention to calculation and algorithms:

> Surely this must be a primary task for critical enquiry – to uncover and

Amoore, Louise

Munster, Anna

calculation|see mathematics!calculation

decision

machine learning!human-machine difference

machine learning|as automation

infrastructure

> probe the moments that come together in the making of a calculation that will automate all future decisions. To be clear, I am not proposing some form of humanist project of proper ethical judgement, but rather calling for attention to be paid to the specific temporalities and norms of algorithmic techniques that rule out, render invisible, other potential futures (
>
> Data Derivatives On the Emergence of a Security Risk Calculus for Our Times **nodate**).

As Amoore writes, some potential futures are being 'ruled out' as calculations automate decisions. Anna Munster puts the challenge more bluntly: 'prediction takes down potential' (

An Aesthesia of Networks **nodate**). I find much to agree with here. Machine learning is a convoluted but nevertheless concrete and historically specific form of calculation (as we will see in exploring algebraic operations in chapter **??**, in finding and optimising certain mathematical functions in chapter **??** or in characterising and shaping probability distributions in chapter **??**). It works to mediate future-oriented decisions (although all too often, very near-future decisions such as ad-click prediction).

I am less certain about treating machine learning as automation. Learning from data, as we will see, does often sidestep and substitute for existing ways of acting, and practices of control, and it thereby re-configures human-machine differences. Yet the notion of automation does not capture well how this comes about. The programs that machine learner 'write' are formulated as probabilistic models, as learned rules or association, and they generate predictive and classificatory statements ('this is a cat'). They render calculable some things which hitherto appeared intractable to calculation (for instance, the argument of a legal case). Such calculation, with all the investment it attracts (in the form of professional lives, in the form of infrastructures , in reorganisation of institutions, corporations and governments, etc.) does rule out some and reinforce other futures. [3] If this transformed calculability is automation, we need to understand the specific contemporary reality of automation as it takes shape in machine learning. We cannot conduct critical enquiry into the calculation that will automate future decisions without opening the very notions of calculation and automation into question.

calculation!historical specificity of

algorithm!primacy

Does the concept of algorithm help us identify the moments that come together in machine learning without resorting to a-historical concepts of automation or calculation? In various scholarly and political debates around changes business, media, education, health, government or science, quasi-omnipotent agency has been imputed to algorithms (

Italian Operaismo and the Information Machine **nodateNeyland_2014**

; The concept of algorithm as an interpretative key of modern rationality **nodate**

; Theses on the Philosophy of History **nodate**

; Popular culture, digital archives and the new social life of data **nodate**

; *Evil Media* **nodate**

; Toward an Anthropology of Computer-Mediated, Algorithmic Forms of Sociality **nodate**

; *Governing Algorithms* **nodate**

; The Relevance of Algorithms **nodate**

; A new algorithmic identity soft biopolitics and the modulation of control **nodate**

3. As for consequences, we need only consider some of the many forms of work that have already been affected by or soon could be affected by machine learning. Postal service clerks no longer sort the mail because neural net-based handwriting recognition reads addresses on envelopes . Locomotives, cars and trucks are already driven by machine learners, and soon driving may not be same occupational cultural it was. Hundreds of occupational categories have to some degree or other machine learners in their near future. Carl Benedikt Frey and Michael Osborne model the chances of occupational change for 700 occupations using, aptly enough, the machine learning technique of Gaussian Processes (
The Future of Employment **nodate**).

Pasquinelli, Paolo
algorithm!as abstraction
mathematics!historicity of

; *Protocol: how control exists after decentralization* **nodate**) or sometimes just 'the algorithm.' This growing body of work understands the power of algorithms in the social science and humanities literature in different ways, sometimes in terms of rules, sometimes as functions or mathematical abstractions, and increasingly as a located practice. but there is general agreement that algorithms are powerful, or at least, can bear down heavily on people's lives and conduct, re-configuring, for instance, culture as algorithmic (

Recommended for you **nodate**).

Some of the critical literature on algorithms identifies abstractions as the source of their power. For instance, in his discussion of the 'metadata society,' Paolo Pasquinelli proposes that

> a progressive political agenda for the present is about moving at the same level of abstraction as the algorithm in order to make the patterns of new social compositions and subjectivities emerge. We have to produce new revolutionary institutions out of data and algorithms. If the abnormal returns into politics as a mathematical object, it will have to find its strategy of resistance and organisation, in the upcoming century, in a mathematical way (
>
> Anomaly Detection **nodate**).

'Moving at the same level of abstraction as the algorithm' offers some purchase as a formulation for critical practice, and for experiments in such practice. Since in mathematics let alone critical thought, abstraction can be understood in many different ways, any direct identification of algorithms with abstraction will, however, be difficult to practice. Which algorithm, what kind of abstraction and which 'mathematical way' should we focus on? Like automation and calculation, abstraction and mathematics have mutable historicities. We cannot 'move at the same level' without taking that into account. Furthermore, given the accumulations of settings, data and devices, there might not be any single level of abstraction to move at, only a torque and flux of different moments of abstraction at work in generalizing, classifying, circulating and stratifying in the midst of transient and plural multiplicities.

abstraction!algorithm as
human-machine relations!practice in
knowledge!power
classifier

The archaeology of operations

Given mathematics and algorithms do loom large in machine learning, how do we address their the workings without pre-emptively ascribing potency to mathematics, or to algorithms? In the chapters that follow, I do explore specific learning algorithms (gradient descent in chapter **??** or recursive partitioning **??**) and mathematical techniques (the sigmoid function in chapter **??** or inner products in chapter **??**) in greater empirical and conceptual depth. Following much scholarship in science and technology studies, I maintain that attention to specificity of practices is an elementary prerequisite to understanding human-machine relations, and their transformations. The archaeology of operations that I will develop combines an interest in machine learning as a form of knowledge production and a strategy of power. Like Foucault, I see no exteriority between techniques of knowledge and strategies of power ('between techniques of knowledge and strategies of power, there is no exteriority, even if they have specific roles and are linked together on the basis of their difference' (

The Will to Knowledge **nodate**, 98)).

If we understand machine learning as a data practice that re-configures local centres of power-knowledge through a re-drawing of human-machine relations, then the specific roles and differences associated with machine learners in the production of knowledge should be a focus of attention. Differences are a key concern here since many machine learners classify things. They are often simply called '.' Some of the practice of difference works in terms of categories. `Kittydar` classifies images as `cat` with some probability, but categorisation and classification in machine learning

Couldry, Nick

categories|seealso differences|categories

power

occurs much more widely.[4] We might understood the importance of categories sociologically. For instance, in his account of media power, Nick Couldry highlights the importance of categories and categorisation:

> *Category* is a key mechanism whereby certain types of ordered (often 'ritualized') practice produce power by enacting and embodying categories that serve to mark and divide up the world in certain ways. Without *some* ordering feature of practice, such as 'categories', it is difficult to connect the multiplicity of practice to the workings of power, whether in the media or in any other sphere. By understanding the work of categories, we get a crucial insight into why the social world, in spite of its massive complexity still appears to us as a *common* world (
>
> *Media, society, world* **nodate**, 62) ,

The orderings of categorical differences undergo a great deal of intensification via machine learning. Categories are often simply an existing set of classifications derived from institutionalised or accepted knowledges (for instance, the categories of customers according to gender or age). Machine learners also generate new categorical workings or mechanisms of differentiation. As we will see (for instance in chapter **??** in relation to scientific data from genomes), machine learners invent or find new sets of categories for a particular purpose (such as cancer diagnosis or prognosis). These differentiations may or may not bring social good. The person who finds themselves paying a higher price for an air ticket by virtue of some unknown combination of factors including age, credit score, home address, previous travel, or educational qualifications experiences something of the classificatory power.

4. John Cheney-Lippold offers a quite general overview of categorization work. He writes: 'algorithm ultimately exercises control over us by harnessing these forces through the creation of relationships between real-world surveillance data and machines capable of making statistically relevant inferences about what that data can mean' (

A new algorithmic identity soft biopolitics and the modulation of control **nodate**, 178). . Much of my discussion here seeks to explore the space of 'statistical inference of what that data can mean.'

Both the intensification of ordering categories and the invention of new classifications feed directly into the regulation of conduct, communication, and ways of living, as will be familiar to readers of Foucault (

Discipline and punish **nodate**). the mathematical abstractions, whether they come from calculus, linear algebra, statistics, or topology, maximise regularities. The power of machine learning to find patterns, to classify or predict regularizes and normalizes differences, although perhaps via some novel ordering practice.

machine learning!regularization

differences!orderings of

Amazon!recommendations

Association Rule Mining|seemachine learner!\textitapriori algorithm

Hammerbacher, Jeff

Asymmetries in a common world

What can critical thought, the kind of enquiry that seeks to identify the conditions that concretely constitute what anyone can say or think or do, learn from machine learning? If we see a massive regularization of order occurring in machine learning, what is at stake in trying to think through those practices? They display moments of formalisation (especially mathematical and statistical), circulation (pedagogically and operationally), generalization (propagating and proliferating in many domains and settings) and stratification (the socially, epistemically, economically and sometimes politically or ontologically loaded re-iterative enactment of categories). I am not sure that understanding how a support vector machine or a random forest orders differences would change how would we relate to what we see, feel, sense, hear or think in the face of a contemporary webpage such as Amazon's that uses Association Rule Mining , an app, a passport control system that matches faces of arriving passengers with images in a database, a computer game, or a genetic test (all settings in which machine learning is likely to be operating).

Machine learners themselves sometimes complain of the monolithic and homogeneous success of machine learning. Some expert practitioners complain of a uniformity in its applications. Jeff Hammerbacher , previously chief research scientist at Facebook, co-founder of a successful data analytics company called Cloudera, and currently advertising, online

digital humanities!use of machine learning

Jockers, Matthew

topic model

Mohr, John

Jockers, Matthew!on topic models

working also on cancer research at Mount Sinai hospital, complained about the spread of machine learning in 2011: 'the best of my generation are thinking about how to make people click ads' (

This Tech Bubble Is Different **nodate**) . Leaving aside debates about the ranking of 'best minds' (a highly competitive and exhaustively tested set of subject positions; see chapter **??**), Hammerbacher was lamenting the flourishing use of predictive analytics techniques in online platforms such as Twitter, Google and Facebook, and on websites more generally, whether they be websites that sell things or advertising space. The mathematical skills of many PhDs from MIT, Stanford or Cambridge were wrangling data in the interests of micro-targeted advertising. As Hammerbacher observers, they were 'thinking about how to make people click ads,' and this 'thinking' mainly took and does take the form of building predictive models that tailored the ads shown on websites to clusters of individual preferences and desires.

Hammerbacher's unhappiness with ad-click prediction resonates with critical responses to the use of machine learning in the digital humanities. Some versions of the digital humanities make extensive use of machine learning. To cite one example, in *Macroanalysis: Digital Methods and Literary History*, Matthew Jockers describes how he relates to one currently popular machine learning or statistical modelling technique, the topic model (itself the topic of discussion in Chapter **??**; see also (

Introduction—Topic models **nodate**)):

> If the statistics are rather too complex to summarize here, I think it is fair to skip the mathematics and focus on the end results. We needn't know how long and hard Joyce sweated over *Ulysses* to appreciate his genius, and a clear understanding of the LDA machine is not required in order to see the beauty of the result. (
>
> *Macroanalysis* **nodate**, 124)

The widely used Latent Dirichlet Allocation or models provide a litmus test of how relations to machine learning is taking shape in the digital humanities. On the one hand, these models promise to make sense of large accumulations of documents (scientific publications, news, literature, online communications, etc.) in terms of underlying themes or latent 'topics.' As we will, large document collections have long attracted the interest of machine learners (see chapter **??**). On the other hand, Jockers signals the practical difficulties of relating to machine learning when he suggests that 'it is fair to skip the mathematics' for the sake of 'the beauty of the result'. While some parts of the humanities and critical social research exhorts closer attention to algorithms and mathematical abstractions, other parts elides its complexity in the name of 'the beauty of the results.'

data!latent variables in

Galloway, Alex|on knowledge production

decision tree

credit scoring|FICO and Equifax

Critical thought has not always endorsed the use of machine learning in digital humanities. Alex Galloway makes two observations about the circulation of these methods in humanities scholarship. The first points to its marginal status in increasingly machine-learned media cultures:

> When using quantitative methodologies in the academy (spidering, sampling, surveying, parsing, and processing), one must compete broadly with the sorts of media enterprises at work in the contemporary technology sector. A cultural worker who deploys such methods is little more than a lesser Amazon or a lesser Equifax.110 (
>
> The Cybernetic Hypothesis **nodate**, 110)

Galloway highlights the asymmetry between humanities scholars and media enterprises or credit score agencies (Equifax). The 'quantitative methodologies' that he refers to as spidering, sampling, processing and so forth are more or less all epitomised in machine learning techniques (for instance, the Association Rule Mining techniques used by Amazon to recommend purchases, or perhaps the decision tree techniques used by the credit-rating systems at Equifax and FICO (

FICO® Analytic Modeler Decision Tree Professional | FICO™ **nodate**)). Galloway's

Galloway, Alex!on capitalist work

capitalism!intellectual work in

machine learning!production of knowledge in

argument is that the infrastructural scale of these enterprises along with the sometime very large technical workforces they employ to continually develop new predictive techniques dwarfs any gain in efficacy that might accrue to humanities research in its recourse to such methods.

Galloway also observes that even if 'cultural workers' do manage to learn to machine learn, and become adept at re-purposing the techniques in the interests of analyzing culture rather than selling things or generating credit scores,they might actually reinforce power asymmetries and exacerbate the ethical and political challenges posed by machine learning:

> Is it appropriate to deploy positivistic techniques against those self-same positivistic techniques? In a former time, such criticism would not have been valid or even necessary. Marx was writing against a system that laid no specific claims to the apparatus of knowledge production itself—even if it was fueled by a persistent and pernicious form of ideological misrecognition. Yet, today the state of affairs is entirely reversed. The new spirit of capitalism is found in brainwork, self-measurement and self-fashioning, perpetual critique and innovation, data creation and extraction. In short, doing capitalist work and doing intellectual work—of any variety, bourgeois or progressive—are more aligned today than they have ever been (
>
> The Cybernetic Hypothesis **nodate**, 110).

This perhaps is a more serious charge concerning the nature of any knowledge produced by machine learning. The 'techniques' of machine learning are positivist (and hence implicitly at odds with critical thought?), and moreover complicit – 'aligned' – with capitalist work. Intellectual work of the kind associated with machine learning is definitely at the centre of many governmental, media, business and scientific fields of operation and increasingly they anchor the operations of these fields. Yet neither observation – asymmetries in scale, alignment with the capitalist knowledge economy – exhaust the potentials of machine learning, particularly if, as many people claim, it transforms the nature of knowledge production and hence 'brainwork.'

What cannot be automated?

Jaron Lanier's question – how will we conceive at a given time what cannot be automated? – suggests an alternative angle of approach. Like Galloway, I'm wary of certain deployments of machine learning, particularly the platform-based media empires and their efforts to capture sociality (

The politics of 'platforms' **nodate**

; Facebook and the engineering of connectivity **nodate**). Machine learners do seem to be 'laying claim to the apparatus of knowledge production.' Yet even amidst the trashy ephemerality of targeted online advertising or the more elevated analytics of literary history, the transformations in knowledge and knowing do not simply simply appropriate intellectual work to capitalist production. Machine learning as a data practice is not simply automating existing economic relations, even if that reproduction heavily steers its normal practice. While Hammerbacher and Galloway are understandably somewhat dismissive of the existential gratifications and critical efficacy of building targeted advertising systems or document classifiers, the 'deployment' of machine learning is not a finished process, but very much in train, constantly subject to revision, re-configuration and alteration. Importantly, the concerns of critical social thought to create knowledge about differences, power, materiality, subject positions, agency, etc. somewhat overlap with the claims that machine learning produces knowledge of differences, of nature, cultural processes, communication and conduct. Unlike other objects of critical thought, machine learners (understood always as human-machine relations) are themselves closely phronesis!predictive

interested in producing knowledge, albeit scientific, governmental or operational.

HERE

Could machine learners become engines of difference? Where in the algorithms, calculations, abstractions and regularizing practices of machine learning would differences be re-draw?

Machine learning in journalism, in specific scientific fields, in the humanities, in social sciences, in art, media, government or civil society sometimes overflows the platform-based deployments and their trenchantly positivist usages. A fairly explicit awareness of the operation of machine-learning driven processes is taking shape in some quarters. And this awareness couples critical and practical responses in a situationally aware calculative knowledge-practice, a form of predictive phronesis (

Nicomachean Ethics **nodate**). For instance, the campaign to re-elect Barack Obama as U.S. President in 2011-12 relied heavily on micro-targetting of voters in the leadup to the election polls (

The Definitive Story of How President Obama Mined Voter Data to Win A Second Term | MIT Technology Review **nodate**

; Code repositories as expressions of urban life **nodate**). In response to the data analytics-driven election campaign run by the US Democrats in support of the 2012 re-election of President Barack Obama, data journalists at the non-profit news organisation *ProPublica* reverse engineered the machine learning models that allowed the Obama re-election team to target individual votes with campaign messages (

How ProPublica's Message Machine Reverse Engineers Political Microtargeting **nodate**). They built their own machine learning model - the 'Message Machine' - using emails sent in by readers and supporters to identify the workings of the Obama campaign team's micro-targetting models. While the algorithmic complexity and data infrastructures used in the Message Machine hardly match those at the disposal of the Obama team, it combines natural language processing (NLP) techniques such as measures of document similarity and machine learning models such as decision trees to disaggregate and map the micro-targetting processes . This kind of reverse engineering work can be found in other quarters. In response to the personalised recommendations generated by streaming media service Netflix, journalists at *The Atlantic* working with Ian Bogost, a media theorist and programmer , reverse engineered the algorithmic production of around 80,000 micro-genres of cinema used by Netflix (

natural language processing
Bogost, Ian
Netflix
Google!Google Flu

How Netflix Reverse Engineered Hollywood **nodate**) . While Netflix's system to categorise films relies on much manual classification and tagging with meta-data, the inordinate number of categories they use is typical of the classificatory regimes that are developing in machine learning-based settings. Certainly, high-profile claims for the power of predictive models to pre-emptively forecast events has come into question. The epidemic predictions of the Google Flu system, a predictive model based on the geography of search engine queries, were wrong on several occasions, mainly because people's online search behaviour changed as a result of coverage (

Life in the network **nodate**

; When Google got flu wrong **nodate**) .

While these cases may be exceptional achievements, and indeed highlight the suffocating weight of the ad-tech application of machine learning, the proliferation of scientific usage suggests that the generalization of machine learning cannot be reduced to personalized advertising. Despite their many operational deployments, the coming together of algorithm, calculation and technique in a form of data practice is not fully coherent or complete. In order to qualify or specify how machine learners exist in their generality, we would need to specify their operations at a level of abstraction that neither attributes a mathematical essence to them nor frames them as producers of relative surplus value. Finding ways of accommodating their diversity, loose couplings and mutability would mean grasping their operational power and their capacity to create new forms of difference.[5]

Different abstractions in machine learning?

Table 1.1 presents a small sample of scientific literature at the intersection of 'climate' and machine learning. This sample, while no doubt dwarfed by the flood of publications on recommendation systems, targeted advertising or handwriting recognition, is typical of the epistemic atmosphere associated with machine learners. (I return to this topic in Chapter **??** in discussing how the leveraging of scientific data via predictive models and classifiers deeply affects the fabric and composition

5. Certain strands of social and cultural theory have taken a strong interest in algorithmic processes as operational forms of power. For instance, the sociologist Scott Lash distinguishes the operational rules found in algorithms from the regulative and constitutive rules studied by many social scientists:

> in a society of pervasive media and ubiquitous coding, at stake is a third type of rule, algorithmic, generative rules. 'Generative' rules are, as it were, virtuals that generate a whole variety of actuals. They are compressed and hidden and we do not encounter them in the way that we encounter constitutive and regulative rules. Yet this third type of generative rules is more and more pervasive in our social and cultural life of the post-hegemonic order. They do not merely open up opportunity for invention, however. They are also pathways through which capitalist power works, in, for example, biotechnology companies and software giants more generally (
>
> Power after Hegemony: Cultural Studies in Mutation? **nodate**, 71).

The term 'generative' is somewhat resonant in the field of machine learning as generative models, models that treat modelling as a problem of specifying the operations or dynamics that could have given rise to the observed data, are extremely important. If we consider only Andrew Ng's CS229 machine learning lectures on Youtube (
Lecture 1 | Machine Learning (Stanford) **nodate**) (lectures I discuss in chapter **??** and draw on throughout this book), we can see that they introduce generative models in Lecture 5 and 6. Although this seems to be only a small part of the 18 lectures given in the course, later lectures on the expectation maximisation algorithm (12-13), and then on unsupervised learning techniques such as factor analysis and principal component analysis, independent component analysis, are also effectively exploring generative models. A similar distribution of topics can be found in *Elements of Statistical Machine Learning*(
The elements of statistical learning **nodate**). Generative models, while perhaps slightly less common in practice than discriminative models, nevertheless capture the sense that algorithms are not just implementations of rules for filtering, sorting, or deciding, but carry within them ontological commitments that might actually challenge social theory in interesting ways. In contrast to Lash, I would suggest that the generativity of these algorithms needs to be differentiated from the algorithmic processes that implement rules more generally. Moving into the data via a generative probabilistic model is very different to moving into the data through say a database query. The models, whether generative or discriminative (models such as decision tree, logistic regression or even neural networks that are more limited in their probabilistic underpinnings), are more like meta-algorithms that reorganize other algorithmic processes on varying scales.

scientific publications

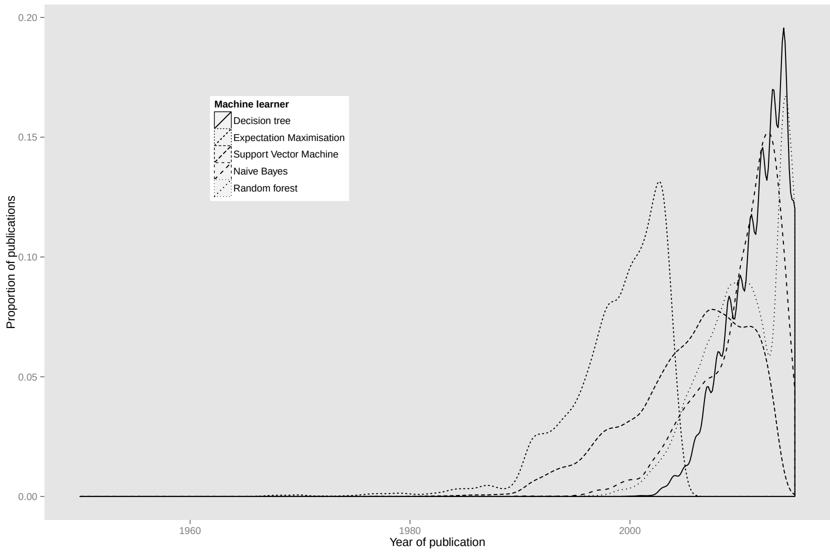

Figure 1.3: Machine learners in scientific literature. The lines in the graph suggest something of the changing fortunes of machine learners over time. The publication data comes from Thomson Reuter's *Web of Science*. Separate searches were run for each machine learner. In these plots, as in the GoogleTrends data, the actual counts of publications have been normalised. In contrast to the GoogleTrend plots, these plots do not show the relative counts of the publications, only their distribution in time.

of objects of scientific knowledge.) But the longevity and plurality of experiments, variants, alternative techniques, implementations and understandings associated with machine learning makes it difficult to immediately reduce them to capitalist captures of knowledge production.

Figure 1.3 derives from counts of scientific publications that mention particular machine learners in their title, their abstract or keywords. The curves, which are probability density plots, suggest a distribution of statements and operations over time for different techniques. Both the duration and the ebbs and flows of work on specific techniques, platforms, knowledges and power relations is still largely occluded. Like the Google Trends results, the lines shown in Figure 1.3 have been normalised in order to adjust for an overall increase in the volume of scientific publications over the last five decades. Unlike the Google Trends search patterns, the scientific literature displays a much more granular and differentiated texture in which different techniques, different terms over the last half century diverge

abstraction!accounts of
abstraction!operational practice of
abstraction!lived

widely from each other. A quick glance at science shows less homogeneity than Hammerbacher and perhaps Galloway see.

Given some degree of pluralism in machine learning as a data practice, what level of abstraction usefully specifies it? If the algorithm and the mathematical constrict it too much, what other level of abstraction could we turn to? Abstraction stands out as one of the most richly developed theoretical resources in the social sciences and humanities. Accounts of abstraction – Karl Marx's real abstraction, Gilles Deleuze and Félix Guattari's abstract machines, Henri Bergson's lived abstraction, Alfred North Whitehead's fortunate abstraction, or Isabelle Stengers' experimental abstractions – are not lacking.[6] Although the theories of abstraction I have just listed differ in many ways, they all, without exception, array themselves in opposition to any limitation of abstraction to mathematical or modern scientific knowledge in particular. All of them, by often convoluted conceptual paths, seek to retrieve from abstraction something conducive to change, differences and contestation.

Any theory of abstraction faces a real test when confronted by operational practices of abstraction such as machine learning. Can a theory of abstraction affirm an operational abstraction without omitting its technical practice? Is machine learning an abstraction we can live with, a lived abstraction as Brian Massumi would call it (

Semblance and Event **nodate**)? I find the question of whether machine learning is a liveable abstraction too hard to answer conclusively. Certainly, accounts of abstraction – abstract machine, real abstraction for instance – help make sense of important movements in machine learning. But without grounding these abstractions in the practice of machine learning, it is very hard to sense how it abstracts. Without tracing how number, chance, classification and event come together in it, the texture of its abstraction, and hence any possible sense of its liveable relationality, remains unfelt and unthought.

6. (
Geography and abstraction Towards an affirmative critique **nodate**) reviews some of the large literature on abstraction. The differences between different accounts of abstraction will run through many of the following chapters.

The diagram

machine learner!SkyNet

Much of this book attempts to identify good levels of abstraction for the data practices of machine learning. The diagram – a form of drawing that smooths away many of the frictions and variations in drawing – anchors much of my account (see chapter **??** for a fuller framing). Diagrams practically abstract. They retain a connection to doing things, such as learning, that other accounts of abstraction sometimes struggle with. Perceptually and technically, they span and indeed criss-cross between human and machine machine learners. They exhibit compositional characteristics of substitution, variation and superimposition, as well as a play or movement amongst their elements. By virtue of its diagrammatic composition, machine learning might bring something new into the worlds it traverses. If machine learners reorganise data, calculation, classification, decision, control and prediction, that might have some precedent. But precedent or novelty would be quite hard to grasp without being able to trace its diagrammatic composition. Similarly, in order to understand the operational forms of power associated with machine learning, the connections connecting data structures, infrastructures, processors, databases and lives might be traceable in diagonal lines, in overlays, and indeed in the densely operational indexes of mathematical formalisms. For instance, how many cat photos are on the internet? This empirical question might be answerable only through machine learning. `kittydar` would need 300,000 cores on Google Compute Engine for several hours. Similarly, NSA's `Skynet` purports to identify Al-Qaeda couriers in Pakistan, but seems to also detect well-known Al-Jazeera journalists (

SKYNET **nodate**). Even the existence of such egregious errors or any other claim to know predicated by `Skynet` could be understood as a diagrammatic practice.

Above all, diagrams can be implemented in multiple ways. Using implementations, graphical and mathematical forms, books and the heavy accumulation of scientific publications from many disciplines (for instance, as seen in figure 1.3), the automation!limits of

book follows six main operations diagrammatically: vectorisation, optimisation, probabilisation, pattern recognition, regularization and propagation. These generic operations compose a diagram of machine learning spanning hardware and software architectures, organizations of data and datasets, practices of designing and testing models, intersections between scientific and engineering disciplines, ongoing historical transformations of notions of pattern, class, rank and order, professional and popular pedagogies, and their associated subject positions. With varying degrees of formalization and consistency, these operations seek to offer an alternative account of machine learning, an account in which some feeling of agency and of affective movement can take route. Where do the operations come from? They as technical processes and practices, sometimes at a quite low level (for instance, vectorisation) and other times widely distributed (for instance, in pedagogy). They become figures when drawn on a diagram. As an abstraction, the diagram of the operational power of machine learning does not map the footprint of a strategic monolith, but highlights the local relations of force that feed into the generalization and plurality of the field in both its monumental and peripheral variations. These local orderings, distributions, rankings and estimation can support hegemonic platforms, but they may also concatenate in different vectors of movement.

TBA

The coming together of data, devices and settings in machine learning may have a play in them that notions of automation, calculation and algorithm abstraction do not fully accommodate. Terms such as automation and algorithm over-prune machine learners as forms of data practice. As I will suggest, algorithms, calculations, techniques and data cohere as machine learners in specific ways. This makes them harder to see as devices or indeed in context. While they can be contextualised in industry, science or government, they themselves contextualise data, decisions, classifications and rankings. Viewed as an ordering practice, we might analyse how machine learners diagonally weave, layer and leverage techniques, calculations and algorithms in infrastructures, institutions and everyday lives. With the certain variations in focus (for instance, paying close attention to the substitution and connection of various elements; see chapter **??**), could we see some stable forms – and these often achieve mathematical formalization as they become stable – amidst the maelstrom of platforms, devices, skills, claims and advocates flowing around them? We might see change that occurs more slowly than what flows around them. Understood as a data practice, we can begin to see the emergence of regularities and forms of order that allow higher levels of abstraction – the algorithm, the data, the mathematical, indeed, abstraction itself – to cohere.

machine learner!diagrammatic composition of

mathematics!as stabilisation of practice

and develop some terms to describe them less generically (common vector space in Chapter **??**, partial observer trajectory in Chapter **??**, decision surface in Chapter **??**, inverse probabilization in Chapter **??**), but all of the facets I describe cluster around the problem of understanding how an almost banal, not esoteric, operation is repeated, borrowed, copied and diffused in so many settings.

Title	Year
Studying climate effects on ecology through the use of climate indices: the North Atlantic Oscillation, El Nino Southern Oscillation and beyond	2003
Decadal Climate Simulations Using Accurate and Fast Neural Network Emulation of Full, Longwave and Shortwave, Radiation	2008
Forecasting climate variables using a mixed-effect state-space model	2011
Climate change threatens the survival of highly endangered Sardinian populations of the snake Hemorrhois hippocrepis	2011
Plant species vulnerability to climate change in Peninsular Thailand	2011
Both topography and climate affected forest and woodland burn severity in two regions of the western US, 1984 to 2006	2011
Climatic niche, ecological genetics, and impact of climate change on eastern white pine (Pinus strobus L.): Guidelines for land managers	2013
Defining spatial conservation priorities in the face of land-use and climate change	2013
Predicting thermal vulnerability of stream and river ecosystems to climate change	2014
Selecting GCM Scenarios that Span the Range of Changes in a Multimodel Ensemble: Application to CMIP5 Climate Extremes Indices	2015

Table 1.1: A small sample of titles of scientific articles that use machine learning in relation to ”climate”

Glossary

classifier A machine learner that assigns instances to classes or categories.. 6

operational formation is a variation on Michel Foucault's discursive formation that highlights the collective human-machine regularities of power-knowledge. While operation and operational fields are intrinsic to Foucault's account of discursive practice, they are somewhat overshadowed by the figures of the document, the utterance, and the proposition.. 7

vectorised Operations on data that transform vectors of values.. *see also* vector

Bibliography

Agency, National Security. **nodate**SKYNET: Courier Detection via Machine Learningday

. **urlseen**. https://theintercept.com/document/2015/05/08/skynet-courier/.

Amoore, Louise. **nodate**Data Derivatives On the Emergence of a Security Risk Calculus for Our Timeseditortypeeditor. **byeditor**maintitle

maintitle

maintitle

maintitle

maintitle

maintitle

maintitle

maintitle

Theory, Culture & Society

28, **number** 6month

:24–43urlmonth.Aristotle_1981Arthur_2012Barocas_2013BBC_2012Beer_2013Beniger_1986Breiman_2001aButler_2013Cheney-Lippold_2011Cleveland_1992Coleman_2012Couldry_2012Cox_2012Domingos_2015aFico_2015Foucault_1972Fou-

cault_1977Foucault_1998Frey_2013Fuller_2012Galloway_2014Galloway_2004Gillespie_2010Gillespie_2014Gitelman_2013Gruber_2004Hallinan_2014Hastie_2009Issenberg_2012Jockers_2013Lamport_1986Lanier_2013Larsen_2012Lash_2007aLazer_2009Le_201
zie_2006Mackenzie_2016aMadrigal_2014Massumi_2013McCormack_2012McMillan_2013Mitchell_1997Mohr_2013Munster_2013Pasquinelli_2014Pasquinelli_2015Perez_2007Sch
taro_2014VanDijck_2012Vance_2011Wilf_2013Xie_2013Xie_2012

Index