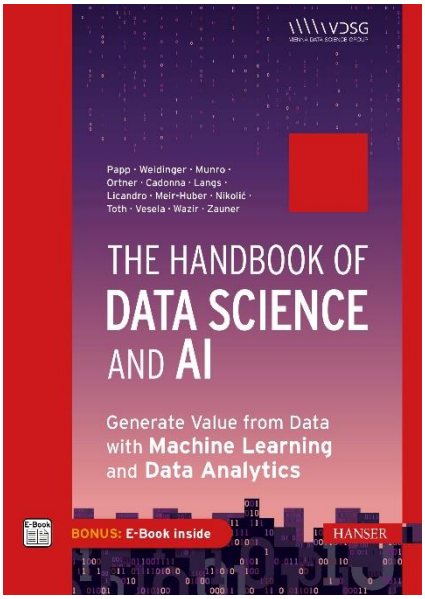
HANSER



Sample Pages

The Handbook of Data Science and AI

Stefan Papp, Wolfgang Weidinger, Katherine Munro, Bernhard Ortner, Annalisa Cadonna, Georg Langs, Roxane Licandro, Mario Meir-Huber, Danko Nikolić, Zoltan Toth, Barbora Vesela, Rania Wazir, Günther Zauner

Print ISBN: 978-1-56990-886-0

E-Book ISBN: 978-1-56990-887-7

ePub ISBN: 978-1-56990-888-4

For further information and order see

www.hanserpublications.com (in the Americas)

www.hanser-fachbuch.de (outside the Americas)

Table of Contents

Foreword ..... **XV**

Preface ..... **XVIII**
Acknowledgments XX

1 Introduction **1**
1.1 What are Data Science, Machine Learning and Artificial Intelligence? 2
1.2 Data Strategy 8
1.3 From Strategy to Use Cases 10
1.3.1 Data Teams 11
1.3.2 Data and Platforms 16
1.3.3 Modeling and Analysis 17
1.4 Use Case Implementation 18
1.4.1 Iterative Exploration of Use Cases 19
1.4.2 End-to-End Data Processing 21
1.4.3 Data Products 22
1.5 Real-Life Use Case Examples 22
1.5.1 Value Chain Digitization (VCD) 22
1.5.2 Marketing Segment Analytics 23
1.5.3 360° View of the Customer 23
1.5.4 NGO and Sustainability Use Cases 24
1.6 Delivering Results 25
1.7 In a Nutshell 27

2 Infrastructure **29**
Stefan Papp
2.1 Introduction 29
2.2 Hardware 31
2.2.1 Distributed Systems 34
2.2.2 Hardware for AI Applications 37
2.3 Linux Essentials for Data Professionals 38

2.4 Terraform 54
2.5 Cloud 58
2.5.1 Basic Services 61
2.5.2 Cloud-native Solutions 65
2.6 In a Nutshell 68

3 Data Architecture 69
Zoltan C. Toth

3.1 Overview 69
3.1.1 Maslow's Hierarchy of Needs for Data 69
3.1.2 Data Architecture Requirements 71
3.1.3 The Structure of a Typical Data Architecture 71
3.1.4 ETL (Extract, Transform, Load) 72
3.1.5 ELT (Extract, Load, Transform) 73
3.1.6 ETLT 73
3.2 Data Ingestion and Integration 74
3.2.1 Data Sources 74
3.2.2 Traditional File Formats 75
3.2.3 Modern File Formats 77
3.2.4 Summary 79
3.3 Data Warehouses, Data Lakes, and Lakehouses 79
3.3.1 Data Warehouses 79
3.3.2 Data Lakes and the Lakehouse 83
3.3.3 Summary: Comparing Data Warehouses to Lakehouses 85
3.4 Data Processing and Transformation 86
3.4.1 Big Data & Apache Spark 86
3.4.2 Databricks 93
3.5 Workflow Orchestration 94
3.6 A Data Architecture Use Case 96
3.7 In a Nutshell 100

4 Data Engineering 101
Stefan Papp, Bernhard Ortner

4.1 Data Integration 102
4.1.1 Data Pipelines 102
4.1.2 Designing Data Pipelines 108
4.1.3 CI/CD 110
4.1.4 Programming Languages 112
4.1.5 Kafka as Reference ETL Tool 115
4.1.6 Design Patterns 119
4.1.7 Automation of the Stages 120
4.1.8 Six Building Blocks of the Data Pipeline 120

4.2 Managing Analytical Models 125
4.2.1 Model Delivery 126
4.2.2 Model Update 127
4.2.3 Model or Parameter Update 128
4.2.4 Model Scaling 128
4.2.5 Feedback into the Operational Processes 129
4.3 In a Nutshell 130

5 Data Management 131
Stefan Papp, Bernhard Ortner

5.1 Data Governance 133
5.1.1 Data Catalog 134
5.1.2 Data Discovery 136
5.1.3 Data Quality 140
5.1.4 Master Data Management 141
5.1.5 Data Sharing 142
5.2 Information Security 143
5.2.1 Data Classification 144
5.2.2 Privacy Protection 145
5.2.3 Encryption 147
5.2.4 Secrets Management 149
5.2.5 Defense in Depth 150
5.3 In a Nutshell 151

6 Mathematics 153
Annalisa Cadonna

6.1 Linear Algebra 154
6.1.1 Vectors and Matrices 154
6.1.2 Operations between Vectors and Matrices 157
6.1.3 Linear Transformations 160
6.1.4 Eigenvalues, Eigenvectors, and Eigendecomposition 161
6.1.5 Other Matrix Decompositions 162
6.2 Calculus and Optimization 163
6.2.1 Derivatives 164
6.2.2 Gradient and Hessian 166
6.2.3 Gradient Descent 167
6.2.4 Constrained Optimization 169
6.3 Probability Theory 170
6.3.1 Discrete and Continuous Random Variables 171
6.3.2 Expected Value, Variance, and Covariance 174
6.3.3 Independence, Conditional Distributions, and Bayes' Theorem 176
6.4 In a Nutshell 177

7 Statistics – Basics ... 179
Rania Wazir, Georg Langs, Annalisa Cadonna

7.1 Data ... 180
7.2 Simple Linear Regression ... 181
7.3 Multiple Linear Regression ... 189
7.4 Logistic Regression ... 191
7.5 How Good is Our Model? ... 198
7.6 In a Nutshell ... 199

8 Machine Learning ... 201
Georg Langs, Katherine Munro, Rania Wazir

8.1 Introduction ... 201
8.2 Basics: Feature Spaces ... 203
8.3 Classification Models ... 206
8.3.1 K-Nearest-Neighbor-Classifier ... 206
8.3.2 Support Vector Machine ... 207
8.3.3 Decision Tree ... 208
8.4 Ensemble Methods ... 209
8.4.1 Bias and Variance ... 210
8.4.2 Bagging: Random Forests ... 211
8.4.3 Boosting: AdaBoost ... 215
8.5 Artificial Neural Networks and the Perceptron ... 215
8.6 Learning without Labels – Finding Structure ... 218
8.6.1 Clustering ... 218
8.6.2 Manifold Learning ... 219
8.6.3 Generative Models ... 220
8.7 Reinforcement Learning ... 221
8.8 Overarching Concepts ... 223
8.9 Into the Depth – Deep Learning ... 224
8.9.1 Convolutional Neural Networks ... 224
8.9.2 Training Convolutional Neural Networks ... 225
8.9.3 Recurrent Neural Networks ... 227
8.9.4 Long Short-Term Memory ... 228
8.9.5 Autoencoders and U-Nets ... 230
8.9.6 Adversarial Training Approaches ... 231
8.9.7 Generative Adversarial Networks ... 232
8.9.8 Cycle GANs and Style GANs ... 234
8.9.9 Other Architectures and Learning Strategies ... 235
8.10 Validation Strategies for Machine Learning Techniques ... 235
8.11 Conclusion ... 237
8.12 In a Nutshell ... 237

9 Building Great Artificial Intelligence ... 239
Danko Nikolić

9.1 How AI Relates to Data Science and Machine Learning ... 239
9.2 A Brief History of AI ... 243
9.3 Five Recommendations for Designing an AI Solution ... 245
9.3.1 Recommendation No. 1: Be pragmatic ... 245
9.3.2 Recommendation No. 2: Make it easier for machines to learn – create inductive biases ... 247
9.3.3 Recommendation No. 3: Perform analytics ... 252
9.3.4 Recommendation No. 4: Beware of the scaling trap ... 254
9.3.5 Recommendation No. 5: Beware of the generality trap (there is no such a thing as free lunch) ... 263
9.4 Human-level Intelligence ... 268
9.5 In a Nutshell ... 270

10 Natural Language Processing (NLP) ... 273
Katherine Munro

10.1 What is NLP and Why is it so Valuable? ... 273
10.2 NLP Data Preparation Techniques ... 275
10.2.1 The NLP Pipeline ... 275
10.2.2 Converting the Input Format for Machine Learning ... 281
10.3 NLP Tasks and Methods ... 283
10.3.1 Rule-Based (Symbolic) NLP ... 284
10.3.2 Statistical Machine Learning Approaches ... 287
10.3.3 Neural NLP ... 295
10.3.4 Transfer Learning ... 301
10.4 At the Cutting Edge: Current Research Focuses for NLP ... 312
10.5 In a Nutshell ... 314

11 Computer Vision ... 317
Roxane Licandro

11.1 What is Computer Vision? ... 317
11.2 A Picture Paints a Thousand Words ... 319
11.2.1 The Human Eye ... 319
11.2.2 Image Acquisition Principle ... 321
11.2.3 Digital File Formats ... 326
11.2.4 Image Compression ... 327
11.3 I Spy With My Little Eye Something That Is... ... 328
11.3.1 Computational Photography and Image Manipulation ... 330
11.4 Computer Vision Applications & Future Directions ... 334
11.4.1 Image Retrieval Systems ... 334
11.4.2 Object Detection, Classification and Tracking ... 337
11.4.3 Medical Computer Vision ... 338

11.5 Making Humans See 341
11.6 In a Nutshell 343

12 Modelling and Simulation – Create your own Models 347
Günther Zauner, Wolfgang Weidinger

12.1 Introduction 347
12.2 General Aspects 349
12.3 Modelling to Answer Questions 349
12.4 Reproducibility and Model Lifecycle 351
12.4.1 The Lifecycle of a Modelling and Simulation Question 352
12.4.2 Parameter and Output Definition 354
12.4.3 Documentation 357
12.4.4 Verification and Validation 357
12.5 Methods 361
12.5.1 Ordinary Differential Equations (ODEs) 361
12.5.2 System Dynamics (SD) 362
12.5.3 Discrete Event Simulation 365
12.5.4 Agent-Based Modelling 368
12.6 Modelling and Simulation Examples 371
12.6.1 Dynamic Modelling of Railway Networks for Optimal Pathfinding Using Agent-based Methods and Reinforcement Learning 371
12.6.2 Agent-Based Covid Modelling Strategies 373
12.6.3 Deep Reinforcement Learning Approach for Optimal Replenishment Policy in a VMI Setting 378
12.7 Summary and Lessons Learned 381
12.8 In a Nutshell 381

13 Data Visualization 385
Barbora Vesela

13.1 History 386
13.2 Which Tools to Use 391
13.3 Types of Data Visualizations 393
13.3.1 Scatter Plot 394
13.3.2 Line Chart 394
13.3.3 Column and Bar Charts 395
13.3.4 Histogram 396
13.3.5 Pie Chart 397
13.3.6 Box Plot 398
13.3.7 Heat Map 398
13.3.8 Tree Diagram 399
13.3.9 Other Types of Visualizations 400
13.4 Select the right Data Visualization 400

13.5 Tips and Tricks 402
13.6 Presentation of Data Visualization 407
13.7 In a Nutshell 407

14 Data Driven Enterprises 411
Mario Meir-Huber, Stefan Papp

14.1 The three Levels of a Data Driven Enterprise 412
14.2 Culture 412
14.2.1 Corporate Strategy for Data 413
14.2.2 The Current State Analysis 415
14.2.3 Culture and Organization of a Successful Data Organisation 417
14.2.4 Core Problem: The Skills Gap 424
14.3 Technology 426
14.3.1 The Impact of Open Source 426
14.3.2 Cloud 426
14.3.3 Vendor Selection 427
14.3.4 Data Lake from a Business Perspective 427
14.3.5 The Role of IT 428
14.3.6 Data Science Labs 428
14.3.7 Revolution in Architecture: The Data Mesh 429
14.4 Business 431
14.4.1 Buy and Share Data 431
14.4.2 Analytical Use Case Implementation 432
14.4.3 Self-service Analytics 433
14.5 In a Nutshell 433

15 Legal foundation of Data Science 435
Bernhard Ortner

15.1 Introduction 435
15.2 Categories of Data 436
15.3 General Data Protection Regulation 437
15.3.1 Fundamental Rights of GDPR 437
15.3.2 Declaration of Consent 438
15.3.3 Risk-assessment 440
15.3.4 Anonymization und Pseudo-anonymization 441
15.3.5 Types of Anonymization 442
15.3.6 Lawful and Transparent Data Processing 444
15.3.7 Right to Data Deletion and Correction 445
15.3.8 Privacy by Design 446
15.3.9 Privacy by Default 446
15.4 ePrivacy-Regulation 446
15.5 Data Protection Officer 447
15.5.1 International Data Export in Foreign Countries 447

15.6 Security Measures 448
15.6.1 Data Encryption 449
15.7 CCPA compared to GDPR 449
15.7.1 Territorial Scope 450
15.7.2 Opt-in vs. Opt-out 450
15.7.3 Right of Data Export 450
15.7.4 Right Not to be Discriminated Against 451
15.8 In a Nutshell 451

16 AI in Different Industries 453
Stefan Papp, Mario Meir-Huber, Wolfgang Weidinger, Thomas Treml, Marek Danis

16.1 Automotive 456
16.1.1 Vision 457
16.1.2 Data 458
16.1.3 Use Cases 458
16.1.4 Challenges 459
16.2 Aviation 461
16.2.1 Vision 461
16.2.2 Data 462
16.2.3 Use cases 462
16.2.4 Challenges 463
16.3 Energy 463
16.3.1 Vision 464
16.3.2 Data 464
16.3.3 Use Cases 465
16.3.4 Challenges 466
16.4 Finance 466
16.4.1 Vision 466
16.4.2 Data 467
16.4.3 Use Cases 467
16.4.4 Challenges 469
16.5 Health 469
16.5.1 Vision 470
16.5.2 Data 471
16.5.3 Use Cases 471
16.5.4 Challenges 471
16.6 Government 472
16.6.1 Vision 472
16.6.2 Data 473
16.6.3 Use Cases 473
16.6.4 Challenges 476

16.7 Art 476
16.7.1 Vision 477
16.7.2 Data 477
16.7.3 Use cases 477
16.7.4 Challenges 478
16.8 Manufacturing 478
16.8.1 Vision 479
16.8.2 Data 479
16.8.3 Use Cases 479
16.8.4 Challenges 480
16.9 Oil and Gas 481
16.9.1 Vision 481
16.9.2 Data 481
16.9.3 Use Cases 482
16.9.4 Challenges 484
16.10 Safety at Work 484
16.10.1 Vision 484
16.10.2 Data 485
16.10.3 Use Cases 485
16.10.4 Challenges 486
16.11 Retail 487
16.11.1 Vision 487
16.11.2 Data 487
16.11.3 Use Cases 488
16.11.4 Challenges 488
16.12 Telecommunications Provider 489
16.12.1 Vision 489
16.12.2 Data 490
16.12.3 Use Cases 490
16.12.4 Challenges 492
16.13 Transport 492
16.13.1 Vision 492
16.13.2 Data 493
16.13.3 Use Cases 493
16.13.4 Challenges 494
16.14 Teaching and Training 494
16.14.1 Vision 495
16.14.2 Data 496
16.14.3 Use Cases 496
16.14.4 Challenges 497
16.15 The Digital Society 497
16.16 In a Nutshell 499

17 Mindset and Community **501**
Stefan Papp

17.1 Data-Driven Mindset 501
17.2 Data Science Culture 504
17.2.1 Start-up or Consulting Firm? 504
17.2.2 Labs Instead of Corporate Policy 505
17.2.3 Keiretsu Instead of Lone Wolf 505
17.2.4 Agile Software Development 507
17.2.5 Company and Work Culture 507
17.3 Antipatterns 510
17.3.1 Devaluation of Domain Expertise 510
17.3.2 IT Will Take Care of It 511
17.3.3 Resistance to Change 511
17.3.4 Know-it-all Mentality 512
17.3.5 Doom and Gloom 513
17.3.6 Penny-pinching 513
17.3.7 Fear Culture 514
17.3.8 Control over Resources 514
17.3.9 Blind Faith in Resources 515
17.3.10 The Swiss Army Knife 516
17.3.11 Over-Engineering 516
17.4 In a Nutshell 517

18 Trustworthy AI **519**
Rania Wazir

18.1 Legal and Soft-Law Framework 520
18.1.1 Standards 522
18.1.2 Regulations 522
18.2 AI Stakeholders 524
18.3 Fairness in AI 525
18.3.1 Bias 526
18.3.2 Fairness Metrics 529
18.3.3 Mitigating Unwanted Bias in AI Systems 532
18.4 Transparency of AI Systems 533
18.4.1 Documenting the Data 534
18.4.2 Documenting the Model 535
18.4.3 Explainability 536
18.5 Conclusion 538
18.6 In a Nutshell 538

19 The authors **539**

Index **545**

Foreword

"Mathematical science shows what is. It is the language of unseen relations between things. But to use and apply that language, we must be able to fully appreciate, to feel, to seize the unseen, the unconscious." – Ada Lovelace

As Computer Literacy over a generation ago represented a new set of foundational skills to be acquired, Artificial Intelligence (AI) Literacy represents the same for our current generations and beyond. Over the last two decades Data Science has come to encompass the mathematical architecture and corresponding language with which we build and interact with systems that extend our senses and decision-making abilities. Thus, it's no longer sufficient to be able to send point-and-click commands into computers, but rather it's vitally important to be able to interpret and interact with AI-enabled recommendations coming out of computers. Currently, machines, as in computers coupled with sensors (in the broadest sense), are processing an increasingly wide array of data including text, images, video, audio, network graphs and a multitude of information from the web, private industry, and public sector sources. Considering diversity of data, the authors of this book approach Data Science as a key underlying topic for society and do so with great insight, from multiple key vantage points, and in enjoyable style that resonates with novices and experts alike.

To gain value from data is arguably the unifying objective of the 21st century knowledge worker. Even professional areas thought of as classically distant from data such as sales and art, now have data-driven sub-areas such as marketing automation and computational design. For the benefit of readers, the authors bring to bear first-hand experiences and diligent research to provide a compelling narrative on how we all have a role to play when attempting to leverage data for better outcomes. Indeed, the breadth conveyed in this work is impressive, spanning that of hardware performance considerations (e.g. CPU, Network, Memory, I/O, GPU) to that of different team member roles when building machines that can find patterns in data. Moreover, the authors provide important coverage on the ways that machines can now see and read, namely, Computer Vision and Natural Language Processing, with implications across nearly every industry area being profound.

As you read this book, I encourage you to be curious and have on top of mind a set of questions on how your professional journey and society as you see it is currently being impacted by increasingly advanced machines: from the capabilities available on your smartphone to that of how jobs are being refashioned in the marketplace with automation tools. Here are some questions to help you get started:

- How does the ratio of what tasks you spend your time on shift with the emergence of increasingly advanced machines in your job area?
- What are the implications of having machines that have perceptive abilities analogous to your own, as in to see, hear, smell, taste, touch and beyond?

- How as society do we grapple with bias in and trust around data?
- How do we make the building and the use of machines that learn more inclusive?
- What distinctly human abilities can you accentuate to help organizations that you care about to be more competitive and sustainable?

I've been cautious not to use the term thinking machines, or artificial general intelligence, as to be wary about overstatements. What I would like to focus your attention on is the wide applicability of what we're seeing coming out of research surrounding machines that have learning capabilities. From my time in laboratories at Columbia and Cornell Universities, to that of the Princeton Plasma Physics Laboratory, the American University of Armenia and NASA-backed TRISH (Translational Research Institute for Space Health) which is collaborating with TrialX, it's clear that machines can find patterns in data across a tremendously wide range of domains and alert humans in both regular and mission critical contexts. Thus, the impacts to human experience are multi-faceted and Data Scientists have an important role in supporting the design of systems where human interaction with machine output is positive sum. I can't underscore this enough that a zero-sum approach to automation in sub-optimal. Entrepreneurs though tend to find a way toward maximum sum.

With colleagues and through my work at the BAJ Accelerator and Covenant Venture Capital, I support startups to engage in a type of tandem learning: how a rapidly growing company can transform an industry by spotting market gaps to that of how a company's invention can learn and provide new capabilities for customers. For example, in the powerful technology area of Computer Vision that is a mainstay in Data Science, three companies stand out as trailblazing in three very different industry areas: Embodied, Scylla and cognaize in healthcare, security and finance, respectively.

- Embodied's flagship product, Moxie, is a robot that supports the emotional well-being and social development of children. To do so, Moxie must see and communicate with family members in a compelling way, understanding visually as well as via other cues the emotional state of people it's interacting with as to engage in meaningful dialogue. Thus, healthcare providers have a new robotic team member to collaborate with. Embodied has been on the cover of TIME Magazine.
- Scylla enables an organization's security team to be proactive in improving safety. With real-time detection capabilities, camera networks no longer need to be passive and can be transformed to being proactive. Applications are numerous from detecting slip-and-falls in hospitals and stadiums as they happen to improve health outcomes to that of making intruder alerts at manufacturing facilities and office buildings to better protect staff. Scylla has been featured in Forbes.
- cognaize supports financial institutions and insurance organizations process a tremendous amount of unstructured data when making risk determinations. A key insight is considering documents not only as text, but rather also considering visual information: style, tables, structure. In addition, cognaize has a human-in-the-loop whereby colleagues and the system overall continually learn. cognaize has been featured on the NASDAQ screen in Times Square.

In the above three examples of rising unicorn startups, Data Scientists work in close collaboration with engineers, analysts, designers, content creators, domain specialists and customers to build machines that learn and interact with humans in nuanced ways. The

result is a transformation in the nature of work: humans are alerted to the most important documents or moments in time and human experience is learned from to improve quality. This is representative of a new shift requiring AI Literacy, where jobs in nearly every facet of the economy will have aspects requiring machine interaction: humans making corrections, learning new skills, reacting to and interpreting alerts, and having a faster response time in helping other humans leveraging machines in support. In the years ahead, I'm excited about the role of Data Science in interface research, new algorithms and how humans can have a force multiplication on their work.

As I co-wrote the first edition of *The Field Guide to Data Science* nearly a decade ago, it's remarkable how much the discipline has advanced both in terms of what has been technically achieved and in an aspirational sense on what is yet to be. The Handbook of Data Science and AI advances the discipline along both of those dimensions and carries the torch forward.

Read on.

Fall 2021

Armen R. Kherlopian, Ph.D.

Preface

"The job of the data scientist is to ask the right questions."
Hillary Mason

Reading the foreword written for our first publication two years ago, I couldn't shake the feeling that some trends essentially stayed the same while others emerged all of a sudden and hit society and companies like an avalanche.

Starting with the changes, that struck society profoundly, it is obvious that the pandemic is one of them. Setting aside the myriad of consequences it had and continues to have on our lives, I want to focus on the facets which relate to the subject of this book: Data Science and AI.

Put simply, the impact there was that entire societies and our whole way of living became data driven in an instant. Key performance indicators like the seven-day incidence rate or forecasts based on pandemic simulations steered our daily life and temporarily even altered basic rights, like the right to leave our homes. This led to discussions and questions, which every Data Scientist with some experience is familiar with and has encountered repeatedly during their working life:

- Can we trust these models and their predictions?
- Is the chosen KPI really the right one for this purpose?
- Is the underlying data quantity and quality good enough?

and so on.

All of these are valid questions and are, just as they were two years ago, fueled by another trend: Digitization. The engine for this is data. On top of that, Data Scientists are still following the same goal:

Giving understandable answers to questions by using data.

Despite all trends, this purpose stays the same and always will be one of the central pillars of doing Data Science.

But this is not the only trend which has remained or become even stronger. The most important, continuing phenomenon is the still massive hype caused by phrases like "Artificial Intelligence" and "Data Science". While these fields are incredibly valuable and powerful, discussions around them unfortunately often evoke false promises and skewed expectations, which in turn lead to disappointment. Some companies already started large, ambitious initiatives in the past, which led to underwhelming results, because expectations were set too high and timelines too short. For example, fully autonomous driving is one particularly challenging problem to solve.

Nevertheless, Artificial Intelligence remains the hope for many companies. Investors perceive it as a general purpose, technology that can be applied almost anywhere. The situation is comparable with the development during the nineties when all things related to the 'Internet' surged. Suddenly, every company needed a web page and significant investments were made to train web programmers. Nowadays, a similar thing happens with everything AI related. Again the investments into AI are enormous and we have a rush of courses on the topic. In the end, the development concerning the 'Internet' led to a vast ecosystem of companies and applications which influence the lives of billions of people in a profound way and it seems that AI follows a similar path.

This explains at least partly another noticeable trend: the further specialization of data science roles with names like "data translator" or "machine learning engineer." It is a somehow natural development as this is a sign that the field is getting more mature, but it also raises the risk of data science responsibilities being scattered across poorly coordinated organizations, and thus, not reaching its full potential. Chapter 14 and 17 go into this in further detail.

Finally, "Trustworthy AI" is emerging as another, highly important movement within Data Science. This is the field of research, which aims to tackle some previously unmet needs, like explainability or fairness. It is therefore included as one of the new chapters in this book (Chapter 18).

Given all these trends in Data Science, one of the reasons for founding the Vienna Data Science Group (VDSG) has become even more important over the last two years: to create a neutral place where interdisciplinary exchange of knowledge between all involved experts can take place internationally. We are still very much dedicated to the development of the entire Data Science ecosystem (education, certification, standardization, societal impact study, and so on), both across Europe and beyond.

A product of the exchange in our community can be found in the 2nd edition of this book, which has been vastly expanded to cover topics like AI (Chapter 9), Machine Learning (Chapter 8), Natural Language Processing (Chapter 10), Computer Vision (Chapter 11) or Modelling and Simulation (Chapter 12) in more depth. To follow our goal to educate society about Data Science and its impacts, a very relevant use case was included in Chapter 12: An agent-based COVID-19 model, which aims to give ideas about the potential impact of certain policies and their combination on the spread of the disease.

To provide our readers with a firm foundation, an introduction to the underlying mathematics (Chapter 6) and statistics (Chapter 7) used in Data Science has been included, and finished with a visualization section (Chapter 13).

Although a lot of content has been added, the goal of this book stays the same and has become even more relevant: to give a realistic picture of Data Science.

Because despite all trends, data science remains the same as well: an interdisciplinary science gathering a very heterogeneous crowd of specialists, which is made up of three major streams:

- Computer Science/IT
- Mathematics/Statistics
- Domain expertise in the industry in which Data Science is applied.

Science aims to generate new knowledge, and this is still used to

- improve existing business processes in a given company (Chapter 16)
- enable completely new business models

Data Science is here to stay and its direct and indirect impact on society is growing at a fast pace, as can be seen during the pandemic. In some areas a bit of disillusionment has set in, but this can be seen as a healthy development to counter the hype. Data Science team roles are becoming more differentiated, and more companies are putting Data Science projects into production.

So, Data Science has grown up and is entering a new era.

Fall 2021
Wolfgang Weidinger

Acknowledgments

We, the authors, would like to take this opportunity to express our sincere gratitude to our families and friends, who helped us to express our thoughts and insights in this book. Without their support and patience, this work would not have been possible.

A special thanks from all the authors goes to Katherine Munro, who contributed a chapter to this book and spent a tremendous amount of time and effort editing our manuscripts.

For my parents, who always said I could do anything. We never expected it would be a thing like this.
Katherine Munro

I'd like to thank my wife and the Vienna Data Science Group for their continuous support through my professional journey.
Zoltan C. Toth

When I think of the people who supported me most, I want to thank my parents, who have always believed in me no matter what and my partner Verena, who was very patient during the last months when I worked on this book. In addition I'm very grateful for the support and motivation I got from the people I met through the Vienna Data Science Group.
Wolfgang Weidinger

9 Building Great Artificial Intelligence

Danko Nikolić

"We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer."
John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon in 1955

Questions Answered in this Chapter:

- What is AI and how is it different from simply creating machine learning models?
- What does it take to create a great AI product?
- What are the common traps when designing and developing an AI, and how can you avoid those traps?

9.1 How AI Relates to Data Science and Machine Learning

You may be asking yourself, what is the function of a chapter on Artificial Intelligence (AI) in a book on data science? Often, AI is understood as just a fancy name for machine learning models, models that data scientists build anyway as part of their job. If that were the case, AI would simply be a part of data science and there would be no need to write a separate chapter on AI as the rest of the book is all about that technology. Well, this is not exactly correct. Although it is true that one of the most important – and perhaps most juicy – parts of AI is in the machine learning models, there is a lot more to AI than just machine learning. There are a few critical considerations that one needs to keep in mind when developing an AI product; considerations which you will not normally find covered in a typical data science book, or indeed, even in other chapters of this book. Critically, if you make a mistake in one of these areas, your final product may disappoint. For example, you may run into a situation in which everything seems fine early in the process of creation, but the final product is underwhelming and does not satisfy the needs and expectations of the end users.

Let us first see which kind of machines we consider today as being examples of AI. What may immediately come to mind is perhaps a robot. However, not just any robot. Most robots are not very intelligent. Robots consist of mechanical components such as arms and actuators. And then there are batteries and sensors. But those alone are not enough to describe a robot as having AI. There are many robots that are quite useful to us but are plain dumb. Examples are vacuum cleaner robots at homes and industrial robots on factory floors. What makes a difference to whether robots will receive the title of being "artificially intelligent" or not is what they can do autonomously with all their hardware. Only a smart robot, one with capabilities far exceeding the plain programming of movements, will be worthy of the honor of being called an AI. We are here looking for a robot that can exhibit a variety of different behaviors, or be able to find its way in a complex environment, or accomplish tasks in a variety of novel situations. For example, think of an anthropomorphic robot capable of clearing up a table full of dirty dishes, then manually washing these dishes and finally, drying them and putting them into the cupboard – and all that without breaking anything! Robots with such a level of skill do not yet exist.

To begin creating such a robot it may soon be clear that training deep learning models will not be enough. One may choose to rely on deep learning to a high degree and yet, the robot will need a lot more than what deep learning can offer. To foster the required intelligence in the robot, we will need to create and use technologies much broader than what machine learning can offer – and also, much broader than what data science covers. Still, data scientists will play a critical role in developing such robots. Hence, you find yourself reading this chapter.

One type of a robot has obtained significant attention from the industry and also a great deal of investments: our cars. A lot of money has been poured into making cars capable of driving by themselves and thus into turning them into intelligent robots. The problem of autonomous driving is not an easy one, especially not if the vehicle is driving in the "real world" and not a controlled test environment. The variety of different situations that the vehicle may encounter is huge. Hence, such vehicles present a great challenge for the technology. Perhaps the autonomous driving problem is as difficult as cleaning up a table with dishes. The pressure for the quality of the solution, that is, not making an error, is high too. While our manual dish-washing robot may in the worst case break a few glasses or plates, a car robot carries a much bigger responsibility; it is responsible for human lives. This is an additional reason that makes a successful self-driving car a tough goal. Nevertheless, there has been quite some progress in this domain. Arguably, autonomous vehicles are the smartest, most intelligent robots which mankind has built so far. And yet, there is still work to be done. The question is then: What did it take to make those machines intelligent? And which intelligence related problems and hurdles do these machines still face? Is it all simply the data science of building bigger and smarter models, or is there more to it?

To address these questions, let us first establish that AI does not equal a machine learning model. To understand that, it helps to make a distinction between a product and a critical component necessary to build a product. A product is a lot more than just its critical components. A knife is more than a blade, although a blade is its critical component. A monitor is more than its critical component, the screen. A memory stick is more than an SSD chip. A bicycle is more than a pair of wheels and pedals. In all these cases we note that a product is more than its critical components.

We can appreciate this difference in the example of a car. A car suitable to sell on the market and thus suitable to produce value for the customer is more than an engine placed on four wheels. For a car, one needs a steering wheel and brakes. Yet, this is still not a complete product. A full product also requires headlights for night driving, a windshield, doors, windows on those doors, wipers on the windshield. One also needs a full cabin with seats. Then one needs a heating system, air-conditioning, and an entertainment system. All this needs to be packed into a beautiful design which is pleasing to a human eye. Only after putting all of this together, are we beginning to have a full product called a car.

An AI is like a full product. It is a machine that does some service for a human and in order to get this service done in a satisfactory fashion, the machine has to be complete. One must create a full product. So, a machine learning model may be a critical component for an AI, maybe the equivalent of what an engine is for a car. Importantly, however, we have an AI only after we have built a product around that (machine learning) engine.

In practice, as a bare minimum, creating a product will require putting the model into production and establishing an interface for acquiring inputs that will go into the machine learning model and then also generating some form of output. Often, there is a lot more required to create a useful product. As we have seen in the case of the autonomous vehicle, there is a lot of hardware needed to create a complete car.

But it is not only "non intelligent" components that one needs to add to machine learning models in order to create an AI. A deeper reason why machine learning alone is not enough for AI is that AI solutions are often a lot more complex than what could be achieved by a single machine learning model. For example, let us consider a chat bot. Let's assume that all we need to create, outside of the intelligent component, is a minimal interface consisting of text fields to enter users' questions and print the machine's answers. One may conclude, then, that it should suffice to place in-between these two components a large, well-trained machine learning model to do the chatting with a human user. Unfortunately, this is not how it works. Every elaborate intelligent chatting assistant (think of Alexa, Siri, Cortana, etc.) is a lot more complex than relying on a single deep learning model.

Below is the architecture of the original Watson AI solution – a machine that made history in 2010 for winning the game of Jeopardy against the top human players in that game. It is clear that the organization of this AI was a lot more elaborate than a single machine learning model. In fact, many of its components do not even rely on machine learning and yet, they nevertheless contribute to the overall intelligence of Watson. It is necessary to understand that only the machine as a whole is an AI; no single component alone is one. Much of this overall intelligence comes from the architecture – how the flow of computation is organized and how it is decided which component will be executed when. Thus, it is not only the weights in the machine learning models that contribute to the overall intelligence. There is a lot more, including the rules by which different models mutually interact and help each other. Only the full combination of all the parts, the Watson, is a full product and is an AI.

Something similar holds for the intelligence of autonomous vehicles. The internal architectures of the algorithms driving the cars are not any simpler than that of Watson. Moreover, over time, as cars become smarter and better drivers, the number of components and the internal complexity of overall AI solutions tends to increase.

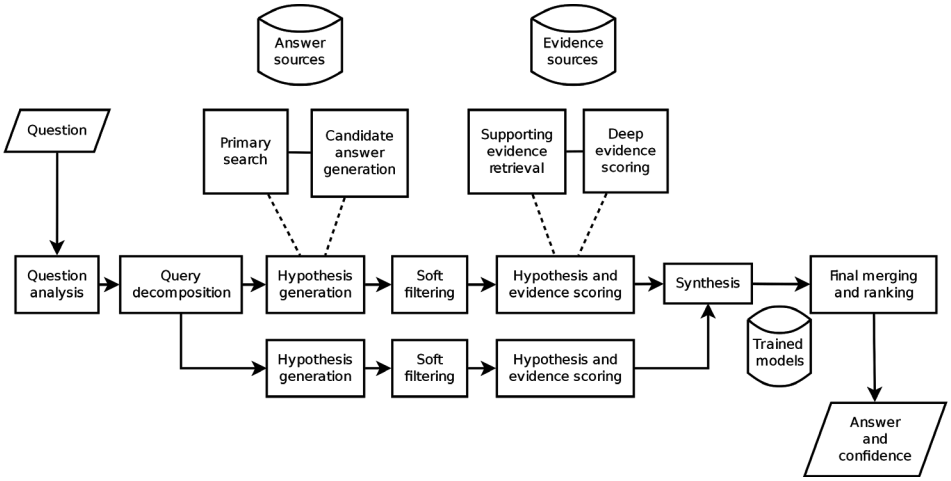


Figure 9.1 The architecture of the original Watson AI that won the game of Jeopardy against the best human competitors *(https://en.wikipedia.org/wiki/File:DeepQA.svg)*

Importantly, many of the components of such solutions are also not machine learning models, but employ some other algorithms. These other components may involve searches through databases, brute-force approaches to finding optimal solutions, pure scientific calculations, rule-based decision making, and so on. Again, all those components jointly contribute to the overall intelligence of the AI.

Finally, there is one more reason why machine learning and AI are not the same thing. Machine learning is often used for purposes other than building intelligent machines. Machine learning has uses that exceed what AI is meant to do. In particular, machine learning is often used as a tool for data analysis. The author of this chapter has extensively used machine learning tools as a means for analyzing how the brain stores sensory information. We trained machine learning models to read information from brain signals. Critically, what interested us was not to build a product. Rather, we asked questions about the brain, for example, how long does the brain hold information about an image that we briefly presented on the screen? Or, how fast can this information be erased by a newly presented stimulus? In this way, we generated numerous insights on how the brain maintains sensory information [1-3]. For pure engineers, such a use of machine learning may come as a surprise. However, for a data scientist, this should not be so unexpected. No scientist should hesitate from using machine learning algorithms as analytics tools. There are great benefits from such uses of machine learning, especially in situations in which the data are complex and insights are difficult to achieve with traditional analytics methods.

To understand the relationship between machine learning and AI, it is common to draw Venn diagrams, like those depicted in Figure 9.2. The Venn diagram on the left is the one which can often be seen in AI literature. But the one on the right is more correct, as it also takes into account the fact that machine learning can be used for purposes other than AI.

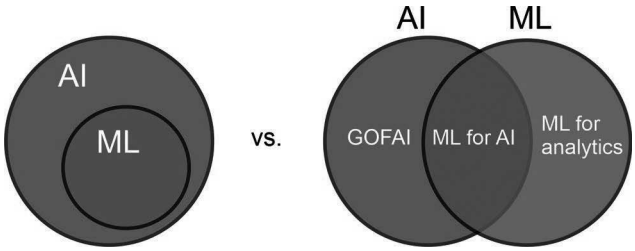


Figure 9.2 The relationship between AI and machine learning (ML). Left: the relationship as commonly depicted in the literature. Right: A more realistic depiction, showing that machine learning can be used for purposes other than AI, such as analyzing data. GOFAI stands for 'Good old-fashioned AI', which does not employ machine learning.

9.2 A Brief History of AI

To appreciate the fact that AI does not need to exclusively rely on machine learning, it is best to take a look at its history. Notably, the history of AI is longer than that of machine learning. In fact, roughly speaking, AI had two major stages. The first stage focussed on the development of algorithms that had nothing to do with machine learning. Instead, these early algorithms relied solely on machine knowledge that was manually fed into the machine – by humans. For example, during this first stage, a large, rule-based decision tree would be considered a state-of-the-art algorithm for AI. Characteristic of this form of AI is that it did not store its knowledge in the form of numbers and did not make conclusions by applying equations to those numbers. The reason was simple: It would have been very hard for humans to feed number-based knowledge, such as the connection weights of artificial neural networks, into a machine. Instead, most of the knowledge was stored in a symbolic form – a form understandable to humans. For example, a knowledge item could use symbols to represent "if fever, then flu". Inferences were made by applying logical rules on those symbols. The rules were again human understandable.

We often refer to this stage of AI as *symbolic* AI (see also Chapter 10). Another commonly known term is Good-old-fashioned-AI, abbreviated as GOFAI. Research on symbolic AI began already in the 1950s, with the official birthplace being the historical Dartmouth conference in 1956. The organizer of this conference, John McCarthy – the person who formulated the term "Artificial Intelligence" – also introduced the first programming language to help computers achieve symbolic intelligence: LISP. The two-letter abbreviation 'AI' did not come into widespread use until after Steven Spielberg's, "A.I. Artificial Intelligence", from 2001.

Machine learning entered the field of AI only later, ushering in its second – and still current – stage. In fact, it wasn't until the 1970's that machine learning really caught on, although some of the algorithms existed a lot earlier. The reason for the delay was that it took time to realize that symbolic AI had limitations and that another approach was needed. One problem was that symbolic AI could not effectively learn on its own; the knowledge needed to be spoon-fed by humans. This produced a huge bottleneck as often the amount of knowledge

needed to be manually set up was too overwhelming. Therefore, the GOFAI approach to increasing the intelligence of machines became unsustainable. As a result, many projects failed to reach the level of usefulness, not moving much further from the initial proof-of-concept; what worked well on a small scale did not materialize on a larger, more useful scale.

Today, in the second stage, we overwhelmingly rely on machine learning algorithms to feed knowledge into machines, transforming it from large datasets into matrices of model parameters. These algorithms provide a great relief from manual work. All that humans need to do is provide examples of intelligent behavior. The machine is then able to extract the rules by which this behavior is being made.

Obviously, this way, we have achieved a great advancement in our ability to increase machine's intelligence. However, it is incorrect to assume that the world has moved away from symbolic AI and that GOFAI algorithms are history. Not at all. The symbolic approach is still alive and well. Every complex AI solution created today is a mash of machine learning and GOFAI components. Symbolic AI is no less important a part. It is only that GOFAI components are not being advertised, which has more to do with the current hype and marketing strategies than with the facts on how the machines work under the hood. Symbolic AI is all over the place. Often it is GOFAI who decides which deep learning algorithm to run next. Other times, GOFAI receives outputs from machine learning models to make the next decision. In other approaches, machine learning assists GOFAI in finding an optimal solution. And so on. Often, the two components are nested: a symbolic algorithm calls machine learning model which in turn calls another GOFAI component for help, going back to machine learning and so on. The possibilities are limitless. Watson could not win a game of Jeopardy without GOFAI components. Without using a GOFAI, in 2016 alphaGo could not have won the game of go against the world champion, Lee Sedol (the score was four to one for the machine). An autonomous vehicle cannot drive without old-fashioned AI components. Alexa, Siri and co. cannot engage in a conversation with you without symbolic parts of their overall intelligence architectures. And so on.

What does it all mean for a data scientist today who is tasked with developing an AI product? Very likely, your solution will need to involve a lot more things than just a machine learning model. There will be a lot of engineering needed outside of machine learning. It will be difficult to avoid symbolic components. This means you will have to make wise architectural decisions about the entire solution and these decisions will include a lot more than just machine learning. Moreover, to create an effective product, you may even need components that lie outside of engineering. A good design of the interface for your AI may be as critical for its success as will the performance of the underlying model. Much like one needs to add an ergonomic handle to a blade to make a good knife, or needs to provide comfortable seats to make a great car, your AI will need to evolve in many different dimensions in order to present a great product. Machine learning models will be just a part of the entire result and thus, only a part of the entire customer experience.

9.3 Five Recommendations for Designing an AI Solution

On the way to creating an AI solution, a data scientist will need to make a number of decisions. You, as a data scientist, will necessarily have to create an architecture combining components of different types, interacting and jointly bringing the intelligence to your machine. Perhaps you will draw this architecture with multiple boxes and arrows, like the drawing of the Watson architecture in Figure 9.1. The question is then: Which strategies can you use and what should you look for to avoid certain common mistakes?

9.3.1 Recommendation No. 1: Be pragmatic

In the previous chapters of this book, you have seen various recipes on how to solve data science problems. This is all presented to you as individual pieces; for example, as individual machine learning algorithms. Also, the pieces are shown in an idealized world, independent from real life. When you design a real AI – a complete product – you will need to think about how to pick algorithms for an imperfect world. You will need to think about how to combine them. Also, you will need to find and use algorithms not described in this book. It is important not to stick with one set of algorithms just because they worked for you in the past, or just because this is what you know. Expand your knowledge, as you need it. Pick the algorithms based on their suitability for a given problem, not based on convenience. Keep in mind that your new problem will always be slightly different from anything else that you have seen in the past. Be eclectic in selecting the tool to solve the tasks. Choose from the widest selection that you possibly can. Do not limit yourself.

Also, stay pragmatic. Your first concern should be achieving the goal. You do not always need to use the latest algorithms, the hottest and most-hyped tool. Rather, take whatever works best for the problem at hand. I have seen data scientists falling "in love" with certain types of models and then playing favorites. But success in data science does not come when you play favorites. I have seen people trying to solve every problem with the same approach. There are individuals who expect that everything must be solved with deep learning. I have also seen die-hard fans of Bayesian approaches. Sure, both Bayesian and deep learning methods are charming and have some attractive features, giving them unique "super-powers". However, both also have disadvantages. In fact, any approach you pick will have some advantages over others, and necessarily also some disadvantages. Your job is to consider both sides and weigh the pros and cons in order to make a good choice.

It is paramount to be aware of both advantages and disadvantages of any given method or algorithm. Disadvantages may be harder to learn about because authors who publish papers about their new methods tend to focus on the positive aspect. The rosy pictures are what motivates them to perform the research and write papers in the first place. So, we should have some understanding. Nevertheless, one still needs to acquire a skill for "reading between lines" and detecting possible limitations and pitfalls. An experienced data scientist will be able to smell possible disadvantages of a new method, even if they are not as clearly spelled out as are the advantages. Develop such a skill, as it will give you a lot of powers for

making good design decisions for your AI architectures. The goal is to acquire knowledge about a lot of algorithms, models, and optimization techniques.

The pool of tools to pick from is huge. A single person can probably never have a full overview of the data science field. Acquiring comprehensive knowledge on machine learning methods and AI algorithms requires life-long learning. And you are never finished. Moreover, the pace with which new algorithms are being proposed is increasing rapidly as more and more people work on the topic, universities open new AI and data science departments, and governments funnel more money towards research in the AI. Keeping up with everything that is going on is a challenge. You should never stop learning but also never expect to know it all.

What helps in navigating this ever-growing forest of new works is a thorough understanding of algorithms. You will be more efficient in understanding a new algorithm if you already have a deep understanding of a related, existing one. Superficial understanding of methods is not nearly as powerful. Proper understanding of several different algorithms, each belonging to a different category, is probably the best strategy one can undertake towards mastering the field of data science. New algorithms are often related to the existing ones. Rarely, researchers come up with an entirely novel approach to solve a machine learning problem (although occasionally they do exactly that). If you understand deeply one algorithm, then it becomes easy for you to quickly grasp the essence of its cousins – they become a variation on the theme. In contrast, if you only superficially understand an algorithm in the first place, a variation of this algorithm may be a mystery for you, and you may have difficulties deciding whether this new variation will be helpful for your new problem or not.

One can always try the algorithm on the data and see what happens. There are also tools for trying multiple algorithms automatically and picking the best one (referred to as autoML). But this cannot get you far. You cannot develop an autonomous vehicle by randomly trying different architectures. By building AI, you will have to do good old human thinking – and a lot of it. In this case you want to minimize decision making by trying the algorithms on your data. Sure, you will have to do that at some point, there's no doubt about this. However, what makes a difference between an experienced AI developer and an inexperienced one is that the former can achieve the task with more thinking and less trying. Experienced people can sift through possibilities in their heads, without having to train the algorithm on the data. The extended knowledge allows them to detect that something is not going to work well even before they try to make it work. This saves a lot of time.

What else can help you make good decisions? A good idea is to draw your future architecture before you start coding. Specify the details and try doing mental simulations of the flow of data throughout the system. At each step ask yourself a question: Do I see a reason this step would fail or have difficulties? If you do see possible problems, address these problems immediately. Pragmatic is to address the weakest points first. Do not hope that a miracle will happen after you spend time working on the easy part.

There is a common belief that with enough computational power and a sufficient amount of data, anything is possible: that anything can be learned by a machine. Although there is some truth to this statement, there is also quite a bit of falsehood there, too. Some of these issues I will address later within this chapter. The bottom line is that blindly following a strategy of more-data-with-more-computation-power is almost guaranteed to bring you problems. It is much better to thoroughly clean up your algorithms by using your understanding

of statistics, machine learning and AI in general. Keep the faith in big data and computational power as your last resource.

Certainly, you will need to try out different designs. And you will need to use the results of these trials as feedback. They will guide you on how to improve. It is vital to realise that your iterations will be much quicker and much more effective if you understand more deeply what you are doing.

Thinking is comparatively hard. Coding and running models is comparatively easy. Still, not shying away from doing the hard part will likely give you the competitive advantage that you will need to create a product that the market needs and enjoys.

Finally, do not forget that one person does not know everything. Build a team of people with different topics of expertise. Have everyone contribute; everyone should have a say. Make sure you get everyone's talent used towards your final product.

9.3.2 Recommendation No. 2: Make it easier for machines to learn – create inductive biases

There is one simple truth about machine learning algorithms: Some learn faster and better than others. In some cases, it takes just a few examples to reach high performance. In other cases, millions of examples are needed. While there are many reasons for these differences, there is one reason which you have the power to control: One factor that determines the learning efficiency of an algorithm are its inductive biases. Inductive bias is like a piece of knowledge added into an algorithm, enabling it to skip some learning steps and walk quicker and more confidently towards the end. Literally, inductive biases enable algorithms to jump to conclusions. And if you have inserted the right inductive biases, your algorithm will jump to the right conclusions, too.

So, what is an inductive bias? It is a predisposition towards finding (i.e., inferring, inducing) a certain relationship in the data. Inductive biases help the algorithm find a certain relationship even if the evidence is very weak and would otherwise require going through millions of data points. Inductive bias is a sort of prejudice to detect a given type of pattern in data.[1] For example, if your mathematical model is made from sine and cosine functions and you fit mostly the parameters of such functions (e.g., amplitude and phase of a sine), then your model will likely be able to fit such functions in the data, even with small amounts of data. In other words, the model will have a bias towards finding a sine wave.

What tricks people into ignoring the importance of inductive biases is that one can in theory use the same type of sine-based model to approximate functions other than sine waves. You could combine millions of sine waves to accurately approximate a power-law function. But this is much harder. You will need a bigger model – that is, one with a larger number of elementary sine waves and therefore, a larger number of parameters – and you will need more data for training.[2] This relationship holds for any model and for any data. You can

[1] Inductive biases have nothing to do with biases in the data, which is an entirely different problem.

[2] Fourier Transform is a tool to assess how complex a sine-wave-based model is needed for a time series. Time series that are periodical and resemble the shapes of sine-waves can be approximated by simple models. Others need complex models and many parameters.

approximate almost anything with large enough deep learning algorithms. And you can achieve similar feats with large enough decision trees (see Section 6.2.3 for decision trees). There is even a mathematical theorem, the Universal Approximation Theorem[3], which proves that an artificial neural network with only one hidden layer can approximate any mathematical function provided enough neurons are available in the hidden layer [4]. So, what is the problem then, if we can approximate anything? Why would we worry about adding inductive biases if models can approximate any function without them? I have already hinted at the most obvious problem: If the inductive biases of the model do not match well with the data, you need a lot of data and a big model and a lot of computation. This also means more CO2 released into the atmosphere during the training and production of the model. None of that is good news.

On the other hand, if you add the correct inductive biases, you can reduce the model size. You can then train it with fewer data points as this leaner model does not fall easily into the local minima of overfitting[4]. The advantages of inductive biases are the reason that we have so many different models. Every problem is a little bit different from any other problem and can be thus more optimally tackled with a more specialized set of equations. Every problem has, in theory, a most optimal possible model specialized for just that problem. Hence, we will never run out of space for inventing new models. The list of all possible models is infinite; we will never reach the end of this list.

I learned about the power of inductive biases in practice on one occasion where my team and I wanted to induce overfitting in deep learning neural networks. Our end goal was to test an algorithm that reduces overfitting in a situation of one-shot learning, and our approach was as follows: generate an unlimited amount of data for training the one-shot learning algorithm (see Chapter 10)[5], induce overfitting on this dataset, and then 'save' the network from overfitting, using our new algorithm. My idea was to create our 'unlimited data' using one deep learning network with a random set of weights, and then train another naive deep learning network to learn the same random mappings. We were confident that we could create overfitting this way, but were proven decisively wrong: We kept reducing the size of the training data set, but the new network did not want to overfit. The performance on the test data remained good, sometimes with as little as 10 or 20 data points. At first, my colleagues and I were puzzled. How was that possible? These were supposed to be very hard data to learn, with complex random relationships in a multi-dimensional space. How could the network learn these relationships with only a small number of examples? This learning was efficient even when we changed the architecture of the network, the number of layers and the sizes of each. The ability to efficiently learn the data was robust.

It took a few days for us to realize that the model which we hoped would overfit was 'doomed' not to, as it had perfect inductive biases for the data. We used the same ReLu and sigmoid transfer functions for generating data and for the model that was learning the data, which basically made the learning model's job very easy. This illustrated to me how powerful inductive biases can be: the same network may need a million examples to learn something counterintuitive for its inductive biases, such as recognizing a flower on a photograph, and only ten examples to learn something that is highly complex for any other model but is

[3] *https://en.wikipedia.org/wiki/Universal_approximation_theorem*

[4] *https://en.wikipedia.org/wiki/Overfitting*

[5] One can learn here about one-shot learning: *https://en.wikipedia.org/wiki/One-shot_learning*

perfectly intuitive for this particular network. This is because the network has exactly the right inductive biases.[6]

Inductive biases give us a lot of possibilities to play with when developing models. The game is two-dimensional. One dimension relates to the type of inductive biases: Should we use ReLu or sigmoid transfer functions, or should we use tangent or even sine waves? This way we are changing which assumptions the model makes about the world. We can replace one assumption for another, and by doing so, we change the inductive biases. A linear model makes a specific assumption about a linear relationship between data. A decision tree makes yet another assumption. And so on.

The other dimension along which we can play with inductive biases is, how tight are the assumptions we want to make? We can make a more relaxed set of assumptions, which basically means having a model with more parameters. We can also make a stricter model, with fewer parameters. By adding more units (neurons) to a neural network, we are relaxing its assumptions. Models that are well suited for a given problem, i.e., have exactly the right set of inductive bases, can often do great work with only a handful of parameters. The biggest models today have billions of parameters. These models are quite relaxed: There are a whole lot of different things that they can possibly learn.

As we mentioned, this has direct implications on the amount of data needed to learn. A strict model will be able to learn from only a few data points of course, provided that the inductive bases are correct. If the inductive biases are incorrect, then a small model will never fit well, no matter how many data points you give it for training. Your only two options for improvement are either increasing the size of the model (with a corresponding increase in the data set size), or getting your inductive biases right. Therefore, even with bad inductive biases you can fit data well; all you need is enough parameters and enough data. Deep learning falls into this latter class of models, not specialized, having relaxed assumptions, and requiring a lot of data. See Figure 9.3 for the relationship between the number of data required (expressed as 'Training effort') and the strictness of the model (expressed as 'Specialization'), across different types of models. The strictest models are the laws of physics. For example, $E = mc^2$ has only one parameter to fit, namely c. Then one can use the 'model' to predict E from m.

[6] Later I learned that someone made the same mistake as we did and published a whole paper without realizing the inductive bias issue that we discovered, thereby making the incorrect conclusion that neural networks are not susceptible to overfitting [5].

18 Trustworthy AI

Rania Wazir

All algorithms should be seen as untrustworthy until proven otherwise.
Cathy O'Neil

Questions Answered in this Chapter:

- What is the current hard-law and soft-law framework for trustworthy AI, especially in the EU?
- Who are the possible AI stakeholders?
- What is fairness in AI, and how is bias defined?
- What are different metrics for measuring the fairness impacts of algorithms?
- What are possible techniques for mitigating unwanted bias?
- How can data and models be documented to improve transparency, usability, and trust?
- What are current methods for explaining model decisions?

The broad class of technologies that fall under the umbrella of AI – from expert systems to machine learning driven solutions and data science applications – are revolutionizing industry, pervading most sectors of the economy and beyond, and have the potential to benefit the economy, society, and the environment. However, as has come to light in recent years, these technologies also come with risks[1,2,3]. Public skepticism has been rising, as examples of stereotyping and discrimination, concerns over worker's rights, and detrimental impact on democratic principles and the environment have been exposed. In order for AI technologies to continue enjoying rapidly growing adoption and realize their beneficial potential, there will be increasing demand for AI-based systems that can be trusted. For AI system providers, this trust translates into increased uptake of products where it is present, and to legal and reputational harms where this trust is breached. In the chapter that follows, we will explore in practice what trust in AI systems means, in particular in the context of machine learning and data science solutions; who are the stakeholders that need to be considered; and some practical implementation steps that can guide the development process.

[1] O'Neil, C., Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, Broadway Books, 2017.
[2] Kate Crawford, AI Now Report 2019
[3] Fundamental Rights Agency of the EU (FRA), Getting the Future Right

Our task will be to try to weave the many disparate requirements together, to create a coherent picture that can accompany the AI system development process from start to finish. We start with the legal and soft-law framework, looking at prominent ethics guidelines, and existing and upcoming regulations and standards. Trust will mean different things to different AI stakeholders – and it is important to identify the various stakeholders involved with an AI system in order to ensure its trustworthiness; we therefore take a brief detour into AI stakeholder identification, before focussing on the issues of fairness in AI, and explainability. This chapter can make no claim to completeness, but aims rather to deliver some guidance to AI system providers and/or users who wish to create/deploy products that can be trusted.

18.1 Legal and Soft-Law Framework

Since 2016, there has been an explosion of so-called “ethics guidelines" for AI. In fact, by 2019 there were already over 80 published guidelines.[4] From academic research institutes to the big tech companies, from international NGOs to state governments, everyone had their input on what constituted “ethical” AI. Unfortunately,most guidelines are rather high level, and diverge on the principles they consider necessary for an AI to be “ethical”. According to the research by Jobin et al.[5], there are five general principles referenced by at least half of the guidelines: transparency, justice and fairness, non-maleficence, responsibility, and privacy; however, their precise meaning and corresponding implementation strategies again diverge.

Some of the main international ethics guidelines on AI include:

- OECD Principles on AI[6]
- UNESCO Recommendation on the Ethics of AI[7]
- UNICEF Policy Guidance on AI for Children[8]
- EU HLEG Guidelines for Trustworthy AI[9]
- EU White Paper on AI[10]

A Trustworthy AI, however, goes beyond ethics. An obvious additional requirement is a quality imperative: the system should be robust, reliable, and safe. The OECD Principles, for example, are addressed to governements and other state actors, intending to serve as guid-

[4] Jobin, Anna, Marcello Ienca, and Effy Vayena. “The global landscape of AI ethics guidelines.” Nature Machine Intelligence 1.9 (2019): 389–399.

[5] Jobin, Anna, Marcello Ienca, and Effy Vayena. “The global landscape of AI ethics guidelines.” Nature Machine Intelligence 1.9 (2019): 389–399.

[6] *https://www.oecd.ai/ai-principles*

[7] *https://unesdoc.unesco.org/ark:/48223/pf0000373434*

[8] *https://www.unicef.org/globalinsight/reports/policy-guidance-ai-children*

[9] *https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai*

[10] *https://ec.europa.eu/info/publications/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en*

ance for fostering the development of Trustworthy AI. They propose the following 5 main principles[11]:

1. **Inclusive growth, sustainable development and well-being.** Poses a general requirement for Trustworthy AI to be beneficial: enhancing human capabilities, reducing inequalities, and protecting the environment.
2. **Human-centred values and fairness.** A Trustworthy AI needs to respect rule of law and human rights, including the right to freedom, the right to dignity and autonomy, the right to privacy and data protection, and the right to non-discrimination.
3. **Transparency and explainability.** Requires responsible disclosure of information about the AI system, in order to foster general understanding of such systems; make stakeholders aware of their interactions with an AI system; and allow those affected by an AI system to understand and contest its outputs.
4. **Robustness, security and safety.** Entails traceability for datasets, processes and decisions; as well as appropriate risk managment measures to address risks such as safety, IT security, privacy, and bias, during each phase of the AI system lifecycle.
5. **Accountability.** All actors involved in developing, deploying or operating AI systems, in accordance with their role, should be held accountable for the proper functioning of the AI systems, including ensuring that the above requirements are met.

The EU High Level Expert Group on AI has an even more extensive list of requirements for a Trustworthy AI, this one addressed to AI system developers, providers, and users.[12] A Trustworthy AI needs to be legal, ethical, and robust, and should satisfy the following requirements:

1. **Human agency and oversight.** Including fundamental rights, human agency and human oversight.
2. **Technical robustness and safety.** Including resilience to attack and security, fall back plan and general safety, accuracy, reliability and reproducibility.
3. **Privacy and data governance.** Including respect for privacy, quality and integrity of data, and access to data.
4. **Transparency.** Including traceability, explainability and communication.
5. **Diversity, non-discrimination and fairness.** Including the avoidance of unfair bias, accessibility and universal design, and stakeholder participation.
6. **Societal and environmental wellbeing.** Including sustainability and environmental friendliness, social impact, society and democracy.
7. **Accountability.** Including auditability, minimisation and reporting of negative impact, trade-offs and redress.

The HLEG Guidance is perhaps one of the most practical set of guidelines available so far. It provides a clear understanding of the reasoning behind the requirements, and information on how to implement them in practice. Based on the guidelines, the group also developed

[11] *https://www.oecd.ai/ai-principles*

[12] High Level Expert Group on Artificial Intelligence set up by the European Commission, "Ethics Guidelines for Trustworthy AI", April 2019, p.14. Accessed from *https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai*

the Assessment List for Trustworthy AI (ALTAI)[13], a tool to help AI system providers, developers, and users assess the extent to which their AI system satisfies the seven requirements for a trustworthy AI.

18.1.1 Standards

The path from guidelines to practical implementation is long, and regulation and international standards are necessary stepping stones. Several international standards organizations are actively involved in creating the necessary standards for ensuring Trustworthy AI:

- **IEEE Ethically Aligned Design:** *https://ethicsinaction.ieee.org/#series*. The IEEE has its own set of ethical guidelines, covering almost 300 pages[14]. This is supplemented by the 7000 Series of Standards, specifying specific aspects of ethical AI. The first two to be published cover general principles of ethical design, and specifications for measuring the human well-being impacts of autonomous and intelligent systems.
- **ISO/IEC Standards on AI and Trustworthy AI:** *https://www.iso.org/committee/6794475.html*. ISO and IEC have established a joint committee to address artificial intelligence. Several standards and technical reports have already been published, and many more are in the pipeline. In particular, the recently published ISO/IEC TR 24028: Overview of trustworthiness in artificial intelligence[15] provides an overview of reqirements and pitfalls in developing and deploying a trustworthy AI system, and can be seen as a roadmap for upcoming standards specifications.
- **NIST Standards for Trustworthy and Responsible AI:** *https://www.nist.gov/programs-projects/trustworthy-and-responsible-ai*. NIST's project includes standards for several key aspects of Trustworthy AI, including most recently a draft publication on mitigating harmful bias[16], as well as previously published standards on explainability and security.
- **CEN-CENELEC Committee on Artificial Intelligence:** *https://www.cencenelec.eu/areas-of-work/cen-cenelec-topics/artificial-intelligence/*. CEN and CENELEC have established the new joint committee in response to the EC White Paper on AI and the German Standardization Roadmap for Artificial Intelligence[17].

18.1.2 Regulations

In particular in the EU, there has been a push to develop a digital strategy that goes beyond guidelines, and imposes some regulation on the AI industry. The first piece of legislation in this direction came with the General Data Protection Regulation (GDPR), which came into force in 2018. Other regulations are in the pipeline – for example, the Digital Services Act (DSA) and the Digital Markets Act (DMA), whose goal is to reduce the "Gatekeeper" effect of

[13] *https://altai.insight-centre.org/*
[14] *https://ethicsinaction.ieee.org/#ead1e*
[15] *https://www.iso.org/standard/77608.html*
[16] *https://doi.org/10.6028/NIST.SP.1270-draft*
[17] *https://www.din.de/en/innovation-and-research/artificial-intelligence*

very large online platforms, and give users and consumers more transparency and choice vis a vis these platforms (DSA), and enable smaller players to enter and compete within the platform economy (DMA). However, while these regulations have elements with direct implications for data collection and AI system transparency, the core regulation addressed to AI is the EU AI Act, which came out in draft form in April 2021.

- **EU Digital Strategy:** *https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age_en*
- **GDPR:** *https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en*
- **DSA:** *https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/digital-services-act-ensuring-safe-and-accountable-online-environment_en*
- **DMA:** *https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/digital-markets-act-ensuring-fair-and-open-digital-markets_en*
- **EU Draft AI Act:** *https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence*

The draft AI Act addresses any AI systems being placed on the market, or put into use, within the EU. It takes a risk-based approach to regulating AI, where risk does not just entail physical or psychological harms, but also risks to fundamental rights. For the scope of the regulation, the draft AI Act makes an intentionally broad definition of AI, and includes many algorithms whose inclusion as "AI" has triggered hot debate: not just machine learning algorithms, but also logic-based methods and expert systems, statistical and Bayesian techniques, optimization and search. The full listing is available in Annex I of the draft AI Act.

The AI Act identifies four types of application which are prohibited, involving subliminal manipulation, social scoring, and facial recognition:

- AI systems that manipulate people and can lead them to behave in ways that are physically or psychologically damaging to themselves or to others.
- AI systems that take advantage of vulnerabilities of particular groups, due to their age or a mental or physical handicap, and can lead to behaviour that is physically or psychologically harmful to themselves or to others.
- Social scoring by public authorities
- The use of real-time remote biometric identification systems in publicly accessible spaces for law enforcement purposes (however, this prohibition comes with several exceptions).

The main substance of the proposed regulation is, however, intended for high-risk applications. These are identified in Annex II – which includes a list of applications already subject to sectoral regulation, and where the act imposes additional obligations – and Annex III, which indicates eight new areas of application, with specific use cases within each area identified as being high risk. Annex II includes, among others, AI systems used in toys, machinery, medical devices, aviation, motor vehicles, and other forms of transport. The areas of application listed in Annex III are:

1. Biometric identification and categorisation of natural persons
2. Management and operation of critical infrastructure
3. Education and vocational training

4. Employment, workers management and access to self-employment
5. Access to and enjoyment of essential private services and public services and benefits
6. Law enforcement
7. Migration, asylum and border control management
8. Administration of justice and democratic processes

The novelty in Annex III is that the draft AI Act reserves to the Commission the right to add use cases to the Annex if the use cases belong to one of the eight areas of application, and are found to pose a high risk to safety, health, or fundamental rights. This enables the Commission to side-step renewed parliamentary negotiations on eventual amendments, and embeds a certain degree of flexibility with which to repsond to new evidence of harm.

The proposed regulation imposes some requirements on providers of high risk AI systems, albeit in most cases, no outside auditing is required, and a self-assessment suffices. The main requirements pertain to data quality and governance (Article 10), risk assessment and risk management systems (Article 9), model performance testing (Article 15), and model documentation (Article 11, Annex IV).

18.2 AI Stakeholders

AI Systems are embedded in complex ecosystems involving a broad range of actors. Understanding risks of bias, and how to mitigate them, involves getting a grasp on the various stakeholders, their roles, and their needs. The following list can serve as a guide, but is by no means exhaustive.

- **Data provider:** organization/person that collects, processes, and delivers the data used by the AI provider.
- **AI provider:** organization/person that develops AI systems. Within the organization, specific additional roles can be identified.
  - Management and Board
  - Legal department/Corporate responsibility department
  - Data Protection Officer
  - System Architects, Data Engineers
  - Developers, Machine Learning Engineers, Data Scientists
  - Quality Assurance
- **AI user:** organization/person that deploys an AI system. Within the organization, specific additional roles can be identified.
  - Management and Board
  - Legal department/Corporate responsibility department
  - Quality Assurance
  - Data Protection Officer

- System Architects, Data Engineers
- Human Resources
- Procurement
- People who have to work directly with the new AI system, or whose jobs are replaced by the new AI system

- **AI subject:** organization/person that AI system outputs/predictions are about.
- **Certification body:** organization that certifies compliance with established standards.
- **Regulator:** authority stipulating performance criteria for AI deployed within their jurisdictions.
- **Broader society, including for example human rights organizations, consumer protection organizations, envirnomental protection organizations, and media:** they may need to be kept informed about requirements for Trustworthy AI, and should be able to request that they are upheld.

18.3 Fairness in AI

What is a fair algorithm? According to the Oxford English Dictionary:

Fairness: *Impartial and just treatment or behaviour without favouritism or discrimination.*

This definition is not yet actionable – in order to determine if an AI system is fair, the concept needs to somehow be quantified. However, fairness is a social construct, and is dependent on context and cultural/societal norms. This has led to the creation of many different definitions of fairness (21 and counting[18]), each with its own mathematical formulation (fairness metric) – as will be described below. To add to the confusion, the terms unfair algorithm and biased algorithm are often used interchangeably.

Bias (Oxford English Dictionary): *Inclination or prejudice for or against one person or group, especially in a way considered to be unfair.*

1.1 A concentration on or interest in one particular area or subject.

1.2 A systematic distortion of a statistical result due to a factor not allowed for in its derivation.

This conflation between unfair and biased may seem natural when considering the main definition of bias. Nonetheless, it is important to consider that any classification model must have bias in order to work. Take, for example, a classifier that has to differentiate between pictures of mammals and of birds. It needs to have a bias towards labelling pictures of animals with wings as birds. Instead, if it were completely free of bias, it would not be able to make any distinction at all, and would place all objects in the same category. So the first clarification is necessary: algorithms need to avoid *unwanted* bias – bias which is based on

[18] Verma, S. and Rubin, J., (2018), "Fairness Definitions Explained", Proceedings of the International Workshop on Software Fairness (FairWare), pp. 1–7.

some protected characteristic or some spurious correlation, and which is not relevant to the task at hand.

Furthermore, within the engineering and statistics communities, a certain kind of unwanted bias already exists: bias according to Definition 1.2 (statistical bias). This often leads to confusion and misunderstanding when discussing bias in machine learning: simply put, an algorithm that is "fair" might still have statistical bias, while at the same time, a system that is free of statistical bias might still be unfair.

The crux of the issue lies within the definition: "a systematic distortion of a statistical result" implies that a "ground truth" (or "true value") is known so that a systematic distortion can be detected by comparison. But what is this "ground truth"? If, as has traditionally been the case, this is the current population parameter value, then it should come as no surprise that, for example, a hiring algorithm for an engineering position trained on historical employment data, would disfavor women precisely because it accurately reflected the status quo (and hence, had no statistical bias). This is not just a mere hypothesis – consider the case of Amazon's ditched machine learning driven recruiting tool[19]. Conversely, in trying to achieve greater gender equity and be "fair", it could be deemed necessary to introduce statistical bias into the algorithm. Of course, this contradiction between statistical bias and fairness might not arise if "ground truth" were taken to be some idealized goal (i.e. the ideal gender distribution of engineering employees). However, this is a controversial issue; and changing the terminology would still leave unresolved the fundamental problem of what the ideal distribution should be. For this reason, many current fairness metrics avoid the use of a "ground truth" as a reference parameter.

In order to avoid confusion, in this chapter, we will use bias to describe inputs to, or properties of, a machine learning model (or more generally, an AI system). Fairness, on the other hand, will be used to describe the impact of model-based outputs or predictions on various protected demographics. This is also consistent with a growing body of literature, which tries to identify and mitigate sources of bias in AI systems, and uses fairness metrics to evaluate model effects.

18.3.1 Bias

Bias can come in many forms, and can enter the machine learning and data science life cycles at various stages. To identify the four main stages:

1. The bias may be in the training or test data. Having large amounts of data does not automatically absolve data collectors from the traditional statistical data errors. Sampling bias, selection bias, and non-response bias are just some of the main traps that data holds for the unaware. However, as the above example of training a hiring algorithm by using historical data shows – even if the procedure for procuring the data was correct statistically, the data could still be biased because of embedded human biases. The hiring data used to train the algorithm might accurately reflect the status quo – and thus encode and perpetuate the current societal bias against women in engineering. Word embeddings

[19] J. Dastin, (2018), 'Amazon scraps secret AI recruiting tool that showed bias against women', Reuters, 11 October 2018.

and language models are another example of such kinds of bias – the text used to train these models is full of societal biases, so that the word embeddings reflect not just general semantic patterns, but also gender[20] and ethnic[21] stereotypes and prejudices.

2. Bias can also enter the system when designing the algorithm – for example, a classification system could be biased because of the categories it is designed to select (black/white, male/female[22]); biases could arise in feature engineering (some features might be more predictive for some groups than for others, and selecting features based on overall accuracy could cause the model to perform worse for some groups), or in the choice of algorithm to use (for example, algorithms that are too simple can underfit the data, and lead to bias in the models). A particularly insidious form of bias can enter the algorithm design when attempting to model a concept that is not fully quantifiable – for example, in a university admissions setting, using records of previously admitted students to train a model for detecting successful candidates to a Ph.D. program[23] (in fact, this simply models previous admissions committees preferences and biases); or in a hospital care management setting, using health care costs as a proxy for severity of the illness to be treated[24].
3. Biases can also enter the system post-hoc, for example, in the interpretation of the model results. Alternatively, decisions based on model predictions could affect data that is then fed back into an online learning algorithm, causing the formation of runaway feedback loops[25], and amplifying existing biases in the data or the model.
4. Finally, deployment is also prone to bias: from temporal drift, to inappropriate use (in a context different from the intended one), and from adversarial attacks (consider, for example, Microsft's infamous Chatbot Tay[26]), to selective deployment (for example, using predictive models to determine grades for children in larger classes, but using human evaluation to determine grades for children in smaller classes[27]).

While it is not possible to list all possible kinds of bias that can become implicated in a machine learning model, we briefly describe below some of the more common forms of bias[28].

---

[20] Bolukbasi, T., Chang, K.-W., Zou, J., Saligramma, V., Kalai, A., (2016), 'Man is to computer programmer as woman is to homemaker? debiasing word embeddings', Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS 2016, pp. 4356–4364.

[21] Manzini, T., Yao Chong, L., Black, A. W., Tsvetkov, Y., (2019), 'Black is to Criminal as Caucasian is to Police: Detecting and Removing Multiclass Bias in Word Embeddings', Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 615–621.

[22] Leufer, D., (2021), 'Computers are binary, people are not: how AI systems undermine LGBTQ identity', Access Now, April 2021.

[23] Burke, L., (2020), U of Texas will stop using controversial algorithm to evaluate Ph.D. applicants, Inside Higher Ed, 14 December 2020.

[24] Obermeyer, Z., Powers, B., Vogeli, C., Mullainathan, S., (2019), 'Dissecting racial bias in an algorithm used to manage the health of populations', Science, Vol. 366, pp. 447–453.

[25] Ensign, D., Friedler, S. A., Neville, S., Scheidegger, C., and Venkatasubramanian, S. (2018), 'Runaway feedback loops in predictive policing', Proceedings of the 1st Conference on Fairness, Accountability and Transparency, PMLR, Vol. 81, pp. 160–171.

[26] The Guardian (2016), 'Microsoft 'deeply sorry' for racist and sexist tweets by AI chatbot', 26 March 2016.

[27] Elbanna, A., Engesmo, J., (2020), 'A-level results: why algorithms get things so wrong – and what we can do to fix them', The Conversation, August 19, 2020.

[28] Suresh, H., and Guttag, J., (2021), 'A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle', arXiv preprint, *https://arxiv.org/pdf/1901.10002.pdf*

- **Human Cognitive Bias:** Any kind of bias that can occur when humans are processing and interpreting information
- **Societal Bias:** Biases and prejudices that arise from a social, cultural, or historical context
- **Confirmation Bias:** A tendency to accept model predictions that are consonant with ones pre-existing beliefs
- **Group Attribution Bias:** Occurs when it is assumed that what is true for an individual in a group is also true for everyone in that group.
- **Automation Bias:** A tendency to over-rely on outputs from a predictive model
- **Temporal Bias:** Bias that arises from not taking into account differences in the observed/measured quantities over time
- **Sampling Bias:** Occurs when data is not sampled randomly from the intended population, so that some individuals are more likely to be included in the sample than others.
- **Representation Bias:** Arises when individuals or groups in a study differ systematically from the population of interest. While this can include the case of sampling bias, it is a broader concept. For example, even if data is sampled randomly from the overall population, the sample sizes, or data quality, for certain subgroups can be low, leading to results that do not generalize well to those subgroups.
- **Measurement Bias:** This type of bias can occur when features and/or labels used in the model are proxies for the actual quantity of interest, possibly introducing systematic errors between what is intended, and what is actually measured (as in the example of using health care costs to measure severity of an illness cited above[29]).
- **Evaluation Bias:** Occurs when testing benchmarks are not properly calibrated, or when performance metrics are not appropriate to the model's deployment context. An often-cited example of this would be the poor performance of facial recognition software on women of color, because they were under-represented in the benchmark data sets used for testing such software[30].
- **Statistical Bias:** The systematic difference between a statistical estimate and the true underlying value ("ground truth")

Given the multiple manifestations of bias, and the several stages at which they can enter the machine learning life cycle, how can bias be detected? Bias in the training/validation/testing data can often be detected through good data documentation practices (see Section 18.4.1), and through the traditional exploratory data analysis (EDA). However, sometimes the bias in the data is too subtle; or else the bias arises at a later stage in the machine learning life cycle. In such cases, bias can only be detected through its effect on the model predictions, by applying some *fairness metrics*.

[29] Obermeyer, Z., Powers, B., Vogeli, C., Mullainathan, S., (2019), 'Dissecting racial bias in an algorithm used to manage the health of populations', Science, Vol. 366, pp. 447–453.

[30] Buolamwini, J., and Gebru, T., (2018), 'Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification', Proceedings of Machine Learning Research, Vol. 81, pp. 1–15. *http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf*

Index

Symbols

1644 387

A

A/B testing 124
accuracy 140, 195
actions 221
Adam optimizer 266
ad-hoc decision 413
adversarial training 232
adversary 231
agent 221
agent-based modelling 368, 373
– actor 368
– agent 368
– COVID-19 373
– emerging behavior 369
– flexibility 369
– natural description 369
AGI 268
agile analytics 426
Agile Manifesto 507
AI for health care 254
AI stakeholders 520, 521, 524, 532–535
Alan Turing 255
AlexNet 226
Aliasing effect 324
alphaGo 244
Amazon Redshift 62
amount of data 249
analytical competence 413
analytics 252
analytics department 428
AnalyticsOps 125
anonymization 441
– types 442
antipatterns 510
Apache Airflow 95
Apache Atlas 137
Apache Hadoop 36, 455
Apache Hive 84
Apache Parquet 33
Apache Spark 86, 128
aperture 322
apparent generalization 264
architecture 241
area chart 395
Area Under the Curve 198
Armenia 508
art 476
Artificial General Intelligence 268
Artificial Intelligence (AI) 8, 30, 201, 239, 393
artificial muse 478
artificial neural networks (ANN) 216
association plot 391
Athena 84
attract attention 391
autoencoders 230
automation 120, 130, 415
automotive 457
autonomous driving 240, 458
autonomous vehicles 241
availability 30
aviation 461
AWS 59
AWS Elastic Beanstalk 64
AWS Redshift Spectrum 84
AWS S3 61
Azure 59
Azure Data Factory 95

Azure Synapse Serverless 84
Azure VM 62

B

bagging 213
balanced scorecard 468
bar chart 388, 395
baseline 405
Bayer pattern 326
Bayes' theorem 176
Bayesian approaches 245
Bernoulli distribution 172
bias 210, 211, 216, 521, 522, 525–527, 532–536
BI department 419
BI Engineer 418
BigQuery 84
Bi-modal IT 511
Binomial distribution 172
biological neurons 215
bootstrapping 198
boundaries 353
box plot 398
brain 242
brute-force 242
brute-force search 267
bullwhip effect 363, 378
business analyst 12, 418
Business Data Owner 418
business intelligence 455

C

calculus 163
calibration 355
camera obscura *see* pinhole camera
capital allocation line 252
Center for Internet security (CIS) 143
Central Limit Theorem 174
central organization 419
Change Data Capture (CDC) 121
Charge-Coupled Device (CCD) 324
Chief Data Officer 418
China 508
Cholesky decomposition 163
churn rate 490
Cilium 39
classification model 181, 201
cloud 426
cloud provider 60
cluster 218
Color Filter Array (CFA) 324
column chart 395
column-oriented storage 78
command line tools
– awk 53
– grep 53
– logcheck 53
– logwatch 53
commodity price 479
completeness 140
compression 327
computational photography 330
computer science 391
Computer Vision 317–319, 329–331, 334
computerized model 353
configuration drift 54
configuration management 58
conflicts of responsibility 414
confusion matrix 124, 195
consistency 141
container 127, 128
containerization 126
Continuous Delivery (CD) 125
Convolutional Neural Network (CNN) 224
Corner detector 330
correlation 175, 252
covariance 175
COVID-19 373
– model calibration 377
– model implementation 378
– parametrization 377
– structure and scheduling 375
CPU 31, 37, 55
cross-validation 124, 198
crypto currencies 252
CSV 75
cumulative variables 253
customer journey 23, 467
customer retention 24
customer satisfaction 462

D

DAG 94
Dagster 95
Dartmouth conference 243
data architect 15
data architectures
– maintainability 71

– reliability 71
– scalability 71
data availability 350
databases 103
Databricks 84, 93, 135
– Unity catalog 135
data catalog 134
data classification 144
data collection 121
data deletion 445
data discovery 136
data engineer 13, 14
data export 445–447
DataFrame API
– Apache Spark 87, 90
– MLlib 92
data governance 133, 444
Data Governance Council 139
data lake 83, 105
DataLakeHouse 106
data lineage 17
data loading scenarios 121
data management 132
data monetization 459
DataOps 125
data ownership 133
data pipeline 102, 110, 120
– data cleaning 121
– data collection 121
– delivery 108
– design 108
– dexploration 120
– exploration 123
– interpretation 124
– modeling 120
– visualization 123
data platform 18, 21, 26
data processing 439
data program 453
data protection 30, 509
data protection officer 447
data quality 140
data Science Lab 429
data science projects 10
data scientist 13, 418
data security 449
DataSet API
– Apache Spark 87
data steward 15
data strategy 21, 415
data swamp 132, 428
data type 401
data warehouse 79, 427, 455
de-biasing 532
decentralized teams 419
decision tree 243, 250
declaration of consent 437, 438, 445–447
– example 439
Deep Blue 255
Deep Fakes 342
deep learning 6, 250, 338
deep learning models 110
defense in depth 150
Delta Lake 78
DenseNets 227
Depth of Field 322
derivative 164
descriptive statistics 252
determinant 161
devil's cycle 270
DevOps 14, 125
digital image 326
digital photography 324
digital society 502
digital transformation 453
directories 64
discrete event simulation 365
– dynamic discrete systems 366
– event list 366
– priorities 366
discriminator 231
disease 374
diversity 503
Docker 39, 115
documentation 357
– source code 357
– textual 357
domain adaptation 340
Domain Expert 418
Driver
– Apache Spark 88
dropout 251
du 42
durability 30
DW 79
DynamoDB 62

E

EC2 (Elastic Compute Cloud) 64
edge detection 329
effect ordering for data display 387
efficient frontier 252
eigendecomposition 161
eigenvalues 161
eigenvectors 161
elasticity 58
embedding space 220
encryption 147
end-to-end process 18
energy sector 463
ensemble methods 213
environment 221
ethics guidelines 520
ETL 72, 102
ETLT 73
Europe 509
Executor
– Apache Spark 88
expectation maximization 218
expected value 174
explainability 521, 533, 537
explainable AI 341
exponential growth 257
external validation set 236
eXtreme Programming 507

F

fail-fast approach 426
fairness 520, 521, 525, 526, 529–532, 536
fairness metrics 526, 529–531, 536
FAIR principle 143
feature engineering 251
feature extraction 329
feature vector 329
financial institutions 466
focal length 322
fraud detection 467
F-score 196
function 163
Function as a Service (FaaS) 59

G

Garry Kasparov 255
Gaussian distribution 173, 266
Gaussian Mixture Model (GMM) 220
GDPR 437
– rights 437
General Data Protection Regulation (GDPR) 132
Generalised Linear Model (GLM) 253
generality trap 263
Generative Adversarial Networks (GAN) 232
generative models 220
genetic algorithms 250
geographic profiling 474
Git 51
global features 329
global minimum 267
GOFAI 243, 244, 267
Google Cloud 59
Google Cloud AI Platform 59
Google Compute Engine 62
GoogLeNet 226
government 472
GPT-3 269
GPU 33, 37
gradient 166, 225
gradient descent 167, 225, 267
– stochastic 168
Grafana 136
great engineering 267
grep 48

H

HDR 331
heat map 398
Hessian 166
heterogenous data 392
hidden layer 216
hidden relationship 386
High Dynamic Range imaging *see* HDR
histogram 388
homography 332
homoscedasticity 266
htop 42
human intelligence 268
hybrid cloud 427
hybrid model-decomposition 356
hyperparameter tuning 250

I

IAM (AWS Identity and Access Management) 64
idempotency 55

identity providers 142
image brightness 329
image formats 326
image morphing 333
ImageNet dataset 258
image registration *see* registration
image resolution 323
image retrieval systems 334, 335
image sharpness 322
image stitching 333
image warping 332
independence of random variables 176
independence of sampling 266
inductive biases 247
information security 143
Infrastructure as a Service (IaaS) 59
inpainting 331
input layer 216
integrity 141
interest points 329
I/O 33
iperf3 47
iTerm2 39

J

Jenkins 110
Jeopardy 241
John McCarthy 243
John Snow 388
joint control contract 440
JPEG 327
jps 50
JSON 77
jumbo packets 32

K

Kanban 507
Keiretsu 505
key management system 149
keypoints *see* interest points
KISS 34
K-means 289
k-means clustering 218
Kubernetes
– Pods 128

L

lakehouse 84
lasso 251
laws of physics 250
l-diversity 146
learning
– few-shot 312
– one-shot 312
– zero-shot 311, 312
legacy systems 33
Lego 262
LeNet 226
lense equation 322
lifecycle 353
limited protectable data 436
linear regression 181, 250
linear relationships 266
linear transformations 160
line chart 394
line graph 388
link aggregation 38
LISP 243
loan acceptance prediction 468
local features 329
logistic regression 191
Logstash 53
long-range correlations 251
long short-term memory (LSTM) 229, 251, 266
loss 259
lsof 48
LU decomposition 162

M

machine learning 201, 335, 341–343
machine learning perpetuum mobile 265
macroscopic methods 361
manage data 392
manifolds 219
manufacturing, mass production 478
MapReduce 36
markdown 51
marketing segment analytics 23
Marvin Minsky 255
Maslow's hierarchy of needs 69
master data 141
matrix 154
– diagonal 157
– identity 156
– inverse 156

– positive definite 161
– transpose 156
matrix multiplication 157
matrix-vector multiplication 159
maximum likelihood 220
Mean Absolute Error 187
Mean Squared Error 183
memory 32
Mersenne Twister 265
microscopic methods 361
Minard 389
MIPS 37
MNIST 256
model
– abstract 350
– comparison 359
– concept 349
– conceptual 353, 354
– epidemic 362
– falsification 351
– implementation 358
– lifecycle 352–354
– output 354
– qualitative 356
– reproducible 351
– stochasticity 377
modelling
– and simulation 347
– Black Box 351
– discriminative 310
– dynamic 348
– generative 310
– iteratively 353
– iterative process 351
– railway networks 371
– static 348
– White Box 352
model scaling 128
model specialization 250
model update 127
modular modelling 374
module 356, 374
– policies 374
Moore-Penrose pseudo-inverse 162
mosaicing *see* Image stitching
mosaic plot 391
multilayer perceptrons 216, 224
multiple linear regression 189

N

Nagios 53
Naive Bayes 250
Natural Language Generation 274
– GPT 311
Natural Language Processing (NLP) 273, 480
– Hidden Markov Models 291
– Naive Bayes 287
– phrase-based machine translation 294
– rule-based (symbolic) 284
– sentiment detection 287
– statistical language modelling 292
– statistical machine learning 287
– symbolic meaning representations 286
– text clustering 288
Natural Language ToolKit (NLTK) 275
Natural Language Understanding 274
– domain detection 274
– intent detection 274
– slot filling 274
NDA (Non-Disclosure Agreement) 511
netstat 48, 49
network 32
neurons 216
Nightingale 390
NLP Data Preparation
– Bag-of-Words (BOW) 282
– lemmatization 278
– Named Entity Recognition (NER) 280
– POS Tagging 277
– stemming 278
– stopword removal 279
– TF-IDF 282
– truncating and padding 283
NLP, neural 295
– Convolutional Neural Networks 295
– decoder 297, 310
– encoder 296
– Feedforward Neural Network 299
– Long Short-Term Memory Networks 296
– neural attention 298
– Recurrent Neural Networks 295
– Seq2Seq 296
NLP transfer learning 301
– BERT 308
– ELMO 305
– GloVe 302
– GPT 310
– M2M 100 313
– Masked Language Modelling 309

– multi-headed attention 307
– Next Sentence Prediction 309
– self-attention 307
– transformer 306
– transformer attention 307
– Word2Vec 302
nmap 48
nmon 49
no free lunch theorem 265

O

object identification 334
ODBC 75
omnichannel process 467
one-shot learning 248
ONNX 128
OPC UA 74
OPC Unified Architecture 74
openAI 259
open innovation 19
open source 426
operational applications 415
operationalization 432
optical axis 322
optical illusions 320
optimization 163
– constrained 169
ordinary differential equations 361
output layer 216
overfitting 211, 248
overplotting 403

P

package manager
– apk 41
– apt 41
– brew 41
– yum 41
parameter 354
parametrization 355
Parquet 77
partial differential equations 348, 361
Pearson's coefficient of correlation 253
perceptron 215
performance 30
Personal Identifiable Information (PII) 132
perspective projection 322
physical security 30
pie chart 388, 397
Pinhole 321
Platform as a Service (PaaS) 59
PMML 128
PNG 328
Poisson distribution 172
polar area chart 390
policy 221
polyglot data storage 456
population 374
power law 259
pragmatism 245
precision 196
prediction 201, 393
predictive analytics 24
predictive maintenance 23, 465, 483, 491–493
Prefect 95
prescriptive analytics 453
Presto 84
Principal Component Analysis (PCA) 161
Privacy by Design 144
probability density function 173
probability mass function 171
probability theory 170
processing 391
Prometheus 53, 136
proof of concepts 414
provisioned IOPS 38
provisioning tools 55
PuTTY 39
PyLint 112
Python 112

Q

quadratic loss function 182
qualitative modelling 356
qualitative variables 180
quality assurance 125
quality optimization 480
quantitative variables 180

R

radiometric resolution 323
random forest (RF) 211, 250, 468
random initialization 219
random number generator 265
random variable
– continuous 172
– discrete 171

RDD API
– Apache Spark 87
recall 195
Receiver Operating Characteristic (ROC) 197
recommendation engines 23, 477
Recurrent Neural Networks (RNN) 227
registration 339
regression model 181, 201, 465
regularization 251
regulations 522, 523
reinforcement learning (RL) 221, 372, 378
relevancy 141
ReLu 248, 264
remote work 514
reproducible
– documentation 351, 357
– transparent 351
– verification and validation 351
– visualization 351
ResNets 227
REST API 74
retail 487
reward 221
ridge 251
Risk Analyst 419
Root Mean Squared Error 183
rose diagram 390
R, programming language 112
rule-based decision making 242

S

S3 64
Sarbanes Oxley Act 132
scaled correlation 253
scale out 34
scale up 34
scaling intelligence 255
scaling trap 254
scatter plot 388, 394
scenarios 353
Schema Evolution 78
Scrum 507
secret managers 149
– Ansible Vault 149
– Hashicorp Vault 149
security experts 16
self-driving cars 334, 337
self-service analytics 433
sensitivity 195
sensitivity analysis 360
sensor data 458
sensor resolution 323
serverless 105, 106
serverless computing
– AWS Lambda 67
– Azure Functions 67
– Google Cloud Functions 67
shared-nothing architecture 63
shared responsibility model 143
Sharpe ratio 252
short-term memory 269
sieve diagram 391
SIFT 330
sigmoid 248
Silicon Valley 508
similarity function 218
singular value decomposition 162
skip connections 230
Smart Cities 474
Snowflake 84
Software as a Service (SaaS) 59
SonarCube 112
spam detection 287
specification 358
specificity 195
SSH Key 45
standards 522, 525
standard deviation 175
state 221
storage services 61
– Azure Storage 61
– Google Storage 61
strategy 8–10, 22, 27
strict model 249
Structured Streaming
– Apache Spark 91
study questions 353
subject matter expert (SME) 510
supply chain 22, 378
support vector machine (SVM) 250
symbolic AI 243
symbolic description 349
System Dynamics 362
– causal loop diagram 364
– flow 363
– hypothesized relations 363
– level 363
– top-down approach 365

T

Tagged Image File Format *see* TIFF
target variable 201
t-closeness 146
telecommunication providers 489
Terraform 55
test data 248
tests of data validity 359
Theatrum Orbis Terrarum 386
thorough understanding 246
three Vs of big data 74
TIFF 328
Time Travel 79
timeliness 141
tmux 44
trace 157
traffic analysis 473
training effort 250
transfer learning 226
transparency 520–523, 533–535
Transport Layer Security (TLS) 148
tree diagram 399
triangulation 386
Trustworthy AI 520–522, 525
Tufte 391

U

under-fitting 211
U-nets 230
uniform distribution 173
uniqueness 141
user-friendly 391

V

validation 357
– face 359
– tests of theories 359
validity 141
values
– endogenous 349
– exogenous 349
van Langren 387
variance 175, 210, 211
variety 17
vector 154
vector multiplication 158
velocity 17
vendor lock-in 427, 455
vendor-managed inventory 378
veracity 17
verification 357
– cross-check 358
– double implementation 359
– formal methods 358
– structural analysis 358
– structured code walk-trough 358
– unit testing 359
Vim 51
vision cycle 343
visual 386
visual system 319
visualization
– data analysis 357
– modelling structure 357
volume 16
Von Neumann architecture 35

W

Watson AI 241
weak learners 210, 213
working memory 269

X

XaaS 59
XML 76

Y

YAGNI 34

Z

zsh 40