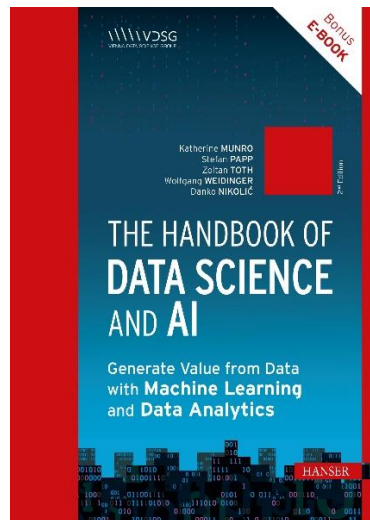


HANSER



Sample Pages

Thermal Analysis in Practice

Katherine Munro, Stefan Papp, Zoltan Toth, Wolfgang Weidinger, Danko Nikolić, Barbora Antosova Vesela, Karin Bruckmüller, Annalisa Cadonna, Jana Eder, Jeannette Gorzala, Gerald Hahn, Georg Langs, Roxane Licandro, Christian Mata, Sean McIntyre, Mario Meir-Huber, György Móra, Manuel Pasieka, Victoria Rugli, Rania Wazir, Günther Zauner

Print ISBN: 978-1-56990-934-8

E-Book ISBN: 978-1-56990-235-6

ePub ISBN: 978-1-56990-411-4

For further information and order see

www.hanserpublications.com (in the Americas)

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

© Carl Hanser Verlag, München

Table of Contents

Preface	XXI
Acknowledgments	XXIII
1 Introduction	1
<i>Stefan Papp</i>	
1.1 About this Book	1
1.2 The Halford Group	2
1.2.1 Alice Halford – Chairwoman	3
1.2.2 Analysts	4
1.2.3 “CDO”	5
1.2.4 Sales	6
1.2.5 IT	7
1.2.6 Security	8
1.2.7 Production Leader	9
1.2.8 Customer Service	10
1.2.9 HR	11
1.2.10 CEO	12
1.3 In a Nutshell	13
2 The Alpha and Omega of AI	15
<i>Stefan Papp</i>	
2.1 The Data Use Cases	16
2.1.1 Bias	16
2.1.2 Data Literacy	19
2.2 Culture Shock	20
2.3 Ideation	24
2.4 Design Process Models	25
2.4.1 Design Thinking	26
2.4.2 Double Diamond	27
2.4.3 Conducting Workshops	28
2.5 In a Nutshell	34

3	Cloud Services	35
	<i>Stefan Papp</i>	
3.1	Introduction	35
3.2	Cloud Essentials	36
3.2.1	XaaS	38
3.2.2	Cloud Providers	39
3.2.3	Native Cloud Services	41
3.2.4	Cloud-native Paradigms	44
3.3	Infrastructure as a Service	45
3.3.1	Hardware	46
3.3.2	Distributed Systems	48
3.3.3	Linux Essentials for Data Professionals	51
3.3.4	Infrastructure as Code	57
3.4	Platform as a Service	61
3.4.1	Cloud Native PaaS Solutions	62
3.4.2	External Solutions	66
3.5	Software as a Service	69
3.6	In a Nutshell	70
4	Data Architecture	71
	<i>Zoltan C. Toth and Sean McIntyre</i>	
4.1	Overview	71
4.1.1	Maslow's Hierarchy of Needs for Data	72
4.1.2	Data Architecture Requirements	73
4.1.3	The Structure of a Typical Data Architecture	74
4.1.4	ETL (Extract, Transform, Load)	78
4.1.5	ELT (Extract, Load, Transform)	79
4.1.6	ETLT	80
4.2	Data Ingestion and Integration	80
4.2.1	Data Sources	80
4.2.2	Traditional File Formats	82
4.2.3	Modern File Formats	84
4.2.4	Which Storage Option to Choose?	86
4.3	Data Warehouses, Data Lakes, and Lakehouses	86
4.3.1	Data Warehouses	86
4.3.2	Data Lakes and Cloud Data Platforms	90
4.4	Data Transformation	93
4.4.1	SQL	95
4.4.2	Big Data & Apache Spark	103
4.4.3	Cloud Data Platforms for Apache Spark	110
4.5	Workflow Orchestration	112
4.5.1	Dagster and the Modern Data Stack	114

4.6	A Data Architecture Use Case	115
4.7	In a Nutshell	119
5	Data Engineering	121
	<i>Stefan Papp</i>	
5.1	Differentiating from Software Engineering	122
5.2	Programming Languages	123
5.2.1	Code or No Code?	123
5.2.2	Language Ecosystem	125
5.2.3	Python	126
5.2.4	Scala	129
5.3	Software Engineering Processes for Data	132
5.3.1	Configuration Management	132
5.3.2	CI/CD	133
5.4	Data Pipelines	135
5.4.1	Common Characteristics of a Data Pipeline	135
5.4.2	Data Pipelines in the Unified Data Architecture	136
5.5	Storage Options	139
5.5.1	File Era	139
5.5.2	Database Era	140
5.5.3	Data Lake Era	142
5.5.4	Serverless Era	143
5.5.5	Polyglot Storage	143
5.5.6	Data Mesh Era	145
5.6	Tooling	146
5.6.1	Batch: Airflow	146
5.6.2	Streaming: Kafka	147
5.6.3	Transformation: Databricks Notebooks	152
5.7	Common challenges	153
5.7.1	Data Quality and Different Standards	154
5.7.2	Skewed Data	155
5.7.3	Stressed Operational Systems	156
5.7.4	Legacy Operational Systems	157
5.7.5	Platform and Information Security	157
5.8	In a Nutshell	157
6	Data Governance	159
	<i>Victoria Rugli, Mario Meir-Huber</i>	
6.1	Why Do We Need Data Governance?	159
6.1.1	Sample 1: Achieving Clarity with Data Governance	160
6.1.2	Sample 2: The (Negative) Impact of Poor Data Governance	161
6.2	The Building Blocks of Data Governance	162
6.2.1	Data Governance Explained	163

6.3	People	165
6.3.1	Data Ownership	165
6.3.2	Data Stewardship	168
6.3.3	Data Governance Board	169
6.3.4	Change Management	170
6.4	Process	172
6.4.1	Metadata Management	173
6.4.2	Data Quality Management	176
6.4.3	Data Security and Privacy	179
6.4.4	Master Data Management	182
6.4.5	Data Access and Search	185
6.5	Technology (Data Governance Tools)	187
6.5.1	Open-Source Tools	187
6.5.2	Cloud-based Data Governance Tools	193
6.6	In a Nutshell	197
7	Machine Learning Operations (ML Ops)	199
	<i>Zoltan C. Toth, György Móra</i>	
7.1	Overview	199
7.1.1	Scope of MLOps	200
7.1.2	Data Collection and Exploration	201
7.1.3	Feature Engineering	201
7.1.4	Model Training	201
7.1.5	Models Deployed to Production	202
7.1.6	Model Evaluation	202
7.1.7	Model Understanding	203
7.1.8	Model Versioning	203
7.1.9	Model Monitoring	204
7.2	MLOps in an Organization	204
7.2.1	Main Benefits of MLOps	204
7.2.2	Capabilities Needed for MLOps	205
7.3	Several Common Scenarios in the MLOps Space	205
7.3.1	Integrating Notebooks	205
7.3.2	Features in Production	207
7.3.3	Model Deployment	209
7.3.4	Model Formats	209
7.4	MLOps Tooling and MLflow	210
7.4.1	MLflow	211
7.5	In a Nutshell	214

8	Machine Learning Security	215
	<i>Manuel Pasieka</i>	
8.1	Introduction to Cybersecurity	216
8.2	Attack Surface	217
8.3	Attack Methods	218
8.3.1	Model Stealing	218
8.3.2	Data Extraction	220
8.3.3	Data Poisoning	222
8.3.4	Adversarial Attack	225
8.3.5	Backdoor Attack	227
8.4	Machine Learning Security of Large Language Models	230
8.4.1	Data Extraction	230
8.4.2	Jailbreaking	232
8.4.3	Prompt Injection	233
8.5	AI Threat Modelling	236
8.6	Regulations	237
8.7	Where to go from here	239
8.8	Conclusion	240
8.9	In a Nutshell	241
9	Mathematics	243
	<i>Annalisa Cadonna</i>	
9.1	Linear Algebra	244
9.1.1	Vectors and Matrices	244
9.1.2	Operations between Vectors and Matrices	247
9.1.3	Linear Transformations	250
9.1.4	Eigenvalues, Eigenvectors, and Eigendecomposition	251
9.1.5	Other Matrix Decompositions	252
9.2	Calculus and Optimization	253
9.2.1	Derivatives	254
9.2.2	Gradient and Hessian	256
9.2.3	Gradient Descent	257
9.2.4	Constrained Optimization	259
9.3	Probability Theory	260
9.3.1	Discrete and Continuous Random Variables	261
9.3.2	Expected Value, Variance, and Covariance	264
9.3.3	Independence, Conditional Distributions, and Bayes' Theorem	266
9.4	In a Nutshell	267

10	Statistics – Basics	269
	<i>Rania Wazir, Georg Langs, Annalisa Cadonna</i>	
10.1	Data	270
10.2	Simple Linear Regression	271
10.3	Multiple Linear Regression	278
10.4	Logistic Regression	280
10.5	How Good is Our Model?	287
10.6	In a Nutshell	289
11	Business Intelligence (BI)	291
	<i>Christian Mata</i>	
11.1	Introduction to Business Intelligence	293
11.1.1	Definition of Business Intelligence	294
11.1.2	Role in Organizations	294
11.1.3	Development of Business Intelligence	295
11.1.4	Data Science and AI in the Context of BI	297
11.1.5	Data for Decision-Making	299
11.1.6	Understanding Business Context	300
11.1.7	Business Intelligence Activities	302
11.2	Data Management Fundamentals	304
11.2.1	What is Data Management, Data Integration and Data Warehousing?	305
11.2.2	Data Load Processes – The Case of ETL or ELT	306
11.2.3	Data Modeling	308
11.3	Reporting and Data Analysis	314
11.3.1	Reporting	314
11.3.2	Types of Reports	317
11.3.3	Data Analysis	318
11.3.4	Visual Analysis	320
11.3.5	Significant Trends	321
11.3.6	Relevant BI Technologies	323
11.3.7	BI Tool Examples	326
11.4	BI and Data Science: Complementary Disciplines	329
11.4.1	Differences	329
11.4.2	Similarities	330
11.4.3	Interdependencies	330
11.5	Outlook for Business Intelligence	331
11.5.1	Expectations for the Evolution of BI	332
11.6	In a Nutshell	333

12	Machine Learning	335
	<i>Georg Langs, Katherine Munro, Rania Wazir</i>	
12.1	Introduction	335
12.2	Basics: Feature Spaces	337
12.3	Classification Models	340
12.3.1	K-Nearest-Neighbor-Classifier	341
12.3.2	Support Vector Machine	342
12.3.3	Decision Trees	342
12.4	Ensemble Methods	344
12.4.1	Bias and Variance	345
12.4.2	Bagging: Random Forests	346
12.4.3	Boosting: AdaBoost	350
12.4.4	The Limitations of Feature Construction and Selection	350
12.5	Unsupervised learning: Learning without labels	351
12.5.1	Clustering	351
12.5.2	Manifold Learning	352
12.5.3	Generative Models	353
12.6	Artificial Neural Networks and Deep Learning	354
12.6.1	The Perceptron	354
12.6.2	Artificial Neural Networks	355
12.6.3	Deep Learning	357
12.6.4	Convolutional Neural Networks	357
12.6.5	Training Convolutional Neural Networks	358
12.6.6	Recurrent Neural Networks	360
12.6.7	Long Short-Term Memory	361
12.6.8	Autoencoders and U-Nets	363
12.6.9	Adversarial Training Approaches	364
12.6.10	Generative Adversarial Networks	365
12.6.11	Cycle GANs and Style GANs	367
12.7	Transformers and Attention Mechanisms	368
12.7.1	The Transformer Architecture	368
12.7.2	What the Attention Mechanism Accomplishes	370
12.7.3	Applications of Transformer Models	370
12.8	Reinforcement Learning	371
12.9	Other Architectures and Learning Strategies	374
12.10	Validation Strategies for Machine Learning Techniques	374
12.11	Conclusion	375
12.12	In a Nutshell	376

13	Building Great Artificial Intelligence	377
	<i>Danko Nikolić</i>	
13.1	How AI Relates to Data Science and Machine Learning	377
13.2	A Brief History of AI	381
13.3	Five Recommendations for Designing an AI Solution	383
13.3.1	Recommendation No. 1: Be Pragmatic	383
13.3.2	Recommendation No. 2: Make it Easier for Machines to Learn – Create Inductive Biases	385
13.3.3	Recommendation No. 3: Perform Analytics	390
13.3.4	Recommendation No. 4: Beware of the Scaling Trap	392
13.3.5	Recommendation No. 5: Beware of the Generality Trap (there is no such a thing as free lunch)	401
13.4	Human-level Intelligence	406
13.5	In a Nutshell	408
14	Signal Processing	411
	<i>Jana Eder</i>	
14.1	Introduction	411
14.2	Sampling and Quantization	413
14.3	Frequency Domain Analysis	416
14.3.1	Fourier Transform	416
14.4	Noise Reduction and Filtering Techniques	422
14.4.1	Denoising Using a Gaussian Low-pass Filter	423
14.5	Time Domain Analysis	425
14.5.1	Signal Normalization and Standardization	425
14.5.2	Signal Transformation and Feature Extraction	425
14.5.3	Time Series Decomposition Techniques	428
14.5.4	Autocorrelation: Understanding Signal Similarity over Time	431
14.6	Time-Frequency Domain Analysis	434
14.6.1	Short Term Fourier Transform and Spectrogram	434
14.6.2	Discrete Wavelet Transform	434
14.6.3	Gramian Angular Field	435
14.7	The Relationship of Signal Processing and Machine Learning	437
14.7.1	Techniques for Feature Engineering	438
14.7.2	Preparing for Machine Learning	438
14.8	Practical Applications	439
14.9	In a Nutshell	441

15	Foundation Models	443
	<i>Danko Nikolić</i>	
15.1	The Idea of a Foundation Model	443
15.2	How to Train a Foundation Model?	446
15.3	How Do we Use Foundation Models?	449
15.4	A Breakthrough: There is no End to Learning	455
15.5	In a Nutshell	456
16	Generative AI and Large Language Models	459
	<i>Katherine Munro, Gerald Hahn, Danko Nikolić</i>	
16.1	Introduction to “Gen AI”	459
16.2	Generative AI Modalities	460
16.2.1	Methods for Training Generative Models	462
16.3	Large Language Models	462
16.3.1	What are “LLMs”?	462
16.3.2	How is Something like ChatGPT Trained?	464
16.3.3	Methods for Using LLMs Directly	465
16.3.4	Methods for Customizing an LLM	475
16.4	Vulnerabilities and Limitations of Gen AI Models	483
16.4.1	Introduction	483
16.4.2	Prompt Injection and Jailbreaking Attacks	484
16.4.3	Hallucinations, Confabulations, and Reasoning Errors	487
16.4.4	Copyright Concerns	488
16.4.5	Bias	491
16.5	Building Robust, Effective Gen AI Applications	494
16.5.1	Control Strategies Throughout Development and Use	494
16.5.2	Guardrails	495
16.5.3	Using Generative AI Safely and Successfully	496
16.6	In a Nutshell	497
17	Natural Language Processing (NLP)	503
	<i>Katherine Munro</i>	
17.1	What is NLP and Why is it so Valuable?	503
17.2	Why Learn “Traditional” NLP in the “Age of Large Language Models”?	505
17.3	NLP Data Preparation Techniques	506
17.3.1	The NLP Pipeline	506
17.3.2	Converting the Input Format for Machine Learning	513
17.4	NLP Tasks and Methods	514
17.4.1	Rule-Based (Symbolic) NLP	515
17.4.2	Statistical Machine Learning Approaches	518
17.4.3	Neural NLP	527
17.4.4	Transfer Learning	532
17.5	In a Nutshell	543

18	Computer Vision	547
	<i>Roxane Licandro</i>	
18.1	What is Computer Vision?	547
18.2	A Picture Paints a Thousand Words	549
18.2.1	The Human Eye	549
18.2.2	Image Acquisition Principle	551
18.2.3	Digital File Formats	556
18.2.4	Image Compression	557
18.3	I Spy With My Little Eye Something That Is... ..	558
18.3.1	Computational Photography and Image Manipulation	560
18.4	Computer Vision Applications & Future Directions	564
18.4.1	Image Retrieval Systems	564
18.4.2	Object Detection, Classification and Tracking	567
18.4.3	Medical Computer Vision	568
18.5	Making Humans See	571
18.6	In a Nutshell	573
19	Modelling and Simulation – Create your own Models	577
	<i>Günther Zauner, Wolfgang Weidinger, Dominik Brunmeir, Benedikt Spiegel</i>	
19.1	Introduction	578
19.2	General Considerations during Modeling	579
19.3	Modelling to Answer Questions	580
19.4	Reproducibility and Model Lifecycle	581
19.4.1	The Lifecycle of a Modelling and Simulation Question	583
19.4.2	Parameter and Output Definition	584
19.4.3	Documentation	587
19.4.4	Verification and Validation	588
19.5	Methods	591
19.5.1	Ordinary Differential Equations (ODEs)	592
19.5.2	System Dynamics (SD)	593
19.5.3	Discrete Event Simulation	596
19.5.4	Agent-based Modelling	599
19.6	Modelling and Simulation Examples	601
19.6.1	Dynamic Modelling of Railway Networks for Optimal Pathfinding Using Agent-based Methods and Reinforcement Learning	602
19.6.2	Agent-Based Covid Modelling Strategies	604
19.6.3	Deep Reinforcement Learning Approach for Optimal Replenishment Policy in a VMI Setting	609
19.6.4	Finding Feasible Solutions for a Resource-constrained Project Scheduling Problem with Reinforcement Learning and Implementing a Dynamic Planing Scheme with Discrete Event Simulation	612
19.7	Summary and Lessons Learned	616
19.8	In a Nutshell	616

20	Data Visualization	621
	<i>Barbora Antosova Vesela</i>	
20.1	History	622
20.2	Which Tools to Use	627
20.3	Types of Data Visualizations	629
20.3.1	Scatter Plot	630
20.3.2	Line Chart	630
20.3.3	Column and Bar Charts	631
20.3.4	Histogram	632
20.3.5	Pie Chart	633
20.3.6	Box Plot	634
20.3.7	Heat Map	634
20.3.8	Tree Diagram	635
20.3.9	Other Types of Visualizations	636
20.4	Select the right Data Visualization	636
20.5	Tips and Tricks	638
20.6	Presentation of Data Visualization	643
20.7	In a Nutshell	643
21	Data Driven Enterprises	647
	<i>Mario Meir-Huber, Stefan Papp</i>	
21.1	The three Levels of a Data Driven Enterprise	648
21.2	Culture	648
21.2.1	Corporate Strategy for Data	649
21.2.2	The Current State Analysis	651
21.2.3	Culture and Organization of a Successful Data Organisation	653
21.2.4	Core Problem: The Skills Gap	660
21.3	Technology	662
21.3.1	The Impact of Open Source	662
21.3.2	Cloud	662
21.3.3	Vendor Selection	663
21.3.4	Data Lake from a Business Perspective	663
21.3.5	The Role of IT	664
21.3.6	Data Science Labs	664
21.3.7	Revolution in Architecture: The Data Mesh	665
21.4	Business	667
21.4.1	Buy and Share Data	667
21.4.2	Analytical Use Case Implementation	668
21.4.3	Self-service Analytics	669
21.5	In a Nutshell	669

22	Creating High-Performing Teams	671
	<i>Stefan Papp</i>	
22.1	Forming	671
22.2	Storming	672
22.2.1	Scenario: 50 Shades of Red	672
22.2.2	Scenario: Retrospective	676
22.3	Norming	678
22.3.1	Change Management and Transition	678
22.3.2	RACI Matrix	680
22.3.3	SMART	682
22.3.4	Agile Processes	683
22.3.5	Communication Culture	685
22.3.6	DataOps	686
22.4	Performing	690
22.4.1	Scenario: A new Dawn	691
22.4.2	Growth Mindsets	692
22.5	In a Nutshell	695
23	Artificial Intelligence Act	697
	<i>Jeannette Gorzala, Karin Bruckmüller</i>	
23.1	Introduction	698
23.2	Definition of AI Systems	700
23.3	Scope and Purpose of the AI Act	701
23.3.1	The Risk-Based Approach	702
23.3.2	Unacceptable Risk and Prohibited AI Practices	703
23.3.3	High-Risk AI Systems and Compliance	705
23.3.4	Medium Risk and Transparency Obligations	707
23.3.5	Minimal Risk and Voluntary Commitments	708
23.4	General Purpose AI Models	708
23.5	Timeline and Applicability	711
23.6	Penalties	711
23.7	AI and Civil Liability	712
23.8	AI and Criminal Liability	712
23.9	In a Nutshell	715
24	AI in Different Industries	717
	<i>Stefan Papp, Mario Meir-Huber, Wolfgang Weidinger, Thomas Tremel</i>	
24.1	Automotive	720
24.1.1	Vision	721
24.1.2	Data	722
24.1.3	Use Cases	722
24.1.4	Challenges	723

24.2	Aviation	725
24.2.1	Vision	725
24.2.2	Data	726
24.2.3	Use Cases	726
24.2.4	Challenges	727
24.3	Energy	727
24.3.1	Vision	728
24.3.2	Data	728
24.3.3	Use Cases	729
24.3.4	Challenges	730
24.4	Finance	730
24.4.1	Vision	730
24.4.2	Data	731
24.4.3	Use Cases	731
24.4.4	Challenges	733
24.5	Health	733
24.5.1	Vision	734
24.5.2	Data	735
24.5.3	Use Cases	735
24.5.4	Challenges	735
24.6	Government	736
24.6.1	Vision	736
24.6.2	Data	737
24.6.3	Use Cases	737
24.6.4	Challenges	740
24.7	Art	740
24.7.1	Vision	741
24.7.2	Data	741
24.7.3	Use cases	741
24.7.4	Challenges	742
24.8	Manufacturing	742
24.8.1	Vision	743
24.8.2	Data	743
24.8.3	Use Cases	743
24.8.4	Challenges	744
24.9	Oil and Gas	745
24.9.1	Vision	745
24.9.2	Data	745
24.9.3	Use Cases	746
24.9.4	Challenges	748
24.10	Retail	748
24.10.1	Vision	748
24.10.2	Data	749

24.10.3 Use Cases	749
24.10.4 Challenges	750
24.11 Telecommunications Provider	750
24.11.1 Vision	751
24.11.2 Data	751
24.11.3 Use Cases	751
24.11.4 Challenges	753
24.12 Transport	753
24.12.1 Vision	754
24.12.2 Data	754
24.12.3 Use Cases	754
24.12.4 Challenges	755
24.13 Teaching and Training	755
24.13.1 Vision	756
24.13.2 Data	757
24.13.3 Use Cases	757
24.13.4 Challenges	758
24.14 The Digital Society	758
24.15 In a Nutshell	760
25 Climate Change and AI	761
<i>Stefan Papp</i>	
25.1 Introduction	761
25.2 AI – a Climate Saver?	763
25.3 Measuring and Reducing Emissions	763
25.3.1 Baseline	763
25.3.2 Data Use Cases	765
25.4 Sequestration	766
25.4.1 Biological Sequestration	768
25.4.2 Geological Sequestration	769
25.5 Prepare for Impact	770
25.6 Geoengineering	771
25.7 Greenwashing	773
25.8 Outlook	774
25.9 In a Nutshell	776
26 Mindset and Community	777
<i>Stefan Papp</i>	
26.1 Data-Driven Mindset	777
26.2 Data Science Culture	780
26.2.1 Start-up or Consulting Firm?	780
26.2.2 Labs Instead of Corporate Policy	781

26.2.3	Keiretsu Instead of Lone Wolf	781
26.2.4	Agile Software Development	783
26.2.5	Company and Work Culture	783
26.3	Antipatterns	786
26.3.1	Devaluation of Domain Expertise	786
26.3.2	IT Will Take Care of It	787
26.3.3	Resistance to Change	787
26.3.4	Know-it-all Mentality	788
26.3.5	Doom and Gloom	789
26.3.6	Penny-pinching	789
26.3.7	Fear Culture	790
26.3.8	Control over Resources	790
26.3.9	Blind Faith in Resources	791
26.3.10	The Swiss Army Knife	792
26.3.11	Over-Engineering	792
26.4	In a Nutshell	793
27	Trustworthy AI	795
	<i>Rania Wazir</i>	
27.1	Legal and Soft-Law Framework	796
27.1.1	Standards	798
27.1.2	Regulations	798
27.2	AI Stakeholders	800
27.3	Fairness in AI	801
27.3.1	Bias	802
27.3.2	Fairness Metrics	805
27.3.3	Mitigating Unwanted Bias in AI Systems	808
27.4	Transparency of AI Systems	809
27.4.1	Documenting the Data	810
27.4.2	Documenting the Model	811
27.4.3	Explainability	812
27.5	Conclusion	814
27.6	In a Nutshell	814
28	Epilogue	815
	<i>Stefan Papp</i>	
28.1	Halford 2.0	815
28.1.1	Environmental, Social and Governance	816
28.1.2	HR	817
28.1.3	Customer Satisfaction	818
28.1.4	Production	819
28.1.5	IT	820
28.1.6	Strategy	822

28.2 Final Words 823

28.3 In a Nutshell 824

29 The Authors 825

Index 833

Preface

This preface was NOT written by ChatGPT (or similar).

As I make this statement, I'm wondering how often it will remain true for text or even other forms of media in the future. Over the last two years, this AI-powered tool has risen to enormous popularity, and has given Data Science and AI an incredible awareness boost. As a result, the expectations for Artificial Intelligence have grown seemingly exponentially, and reached such heights that one might ask, if they can ever be achieved.

AI is following the well-known hype cycle. Some of these high expectations are well-deserved: this powerful technology will change the way we live and work in many ways. To name one example: some universities are considering not to ask their students for seminar papers any longer, as it's not possible to check if it was written by an AI tool.

But we also must brace ourselves for some disappointment in the future, as AI inevitably fails to live up to certain people's inflated expectations.

Even when the vision is reasonable, often the timelines these people and organizations have in mind for implementing AI projects is not. This leads to further disappointment, when the hoped-for impact and value fail to materialize within the desired timeframe.

We're already seeing the beginning of this, with ChatGPT and similar tools generating plenty of eloquent and coherent – yet completely inaccurate – information. This isn't helped by the new wave of 'AI experts', who are making ever more outlandish promises about tools invented by themselves or their companies; promises which will be very hard to keep. They are, essentially, selling digital 'snake oil'.

All of this puts even more pressure on data scientists to deal with these expectations, while continuing to deliver on the same goal they've had for decades:

generating understandable answers to questions, using data.

This is what makes neutral organizations such as the Vienna Data Science Group (VDSG [www.vdsg.at]) – which fosters interdisciplinary and international knowledge exchange between data experts – so necessary and important. We are still highly dedicated to the development of the entire Data Science and AI ecosystem (education, certification, standardization, societal impact study, and so on), across Europe and beyond. This book represents just one of our efforts towards this goal. Because despite all the hype and hyperbole in the AI and data landscape, Data Science remains the same: an interdisciplinary science gathering a very heterogeneous crowd of specialists. It is made up of three major streams, and we are proud to have expert members in each of them:

- Computer Science and IT
- Mathematics and Statistics
- Domain expertise in the industry or field in which Data Science and AI is applied.

As a matter of fact, the VDSG [www.vdsg.at] has always taken a holistic approach to data science, and this book is no different: Starting at Chapter 1 we introduce a fictional company who wants to become more data driven, and we check in with them throughout the book, right up to the end of their data transformation in Chapter 28. Along the way we cover many challenges in their journey, thus providing you with practical insights which were only possible thanks to vibrant exchange among our vast Data Science and AI community.

The result is a greatly expanded edition of our Data Science & AI Handbook, with 10 new chapters covering topics like Building AI solutions (Chapter 13), Foundation Models (Chapter 15), Large Language Models and Generative AI (Chapter 16) and Climate Change and AI (Chapter 25). This is complemented by also tackling the fundamental topics of Data Architecture, Engineering and Governance (Chapters 4, 5 and 6) and topping it off with Machine Learning Operations (MLOps, Chapter 7), which has become a very important discipline in itself.

To provide a firm foundation to help you understand all this, we've again included an introduction to the underlying Mathematics (Chapter 9) and Statistics (Chapter 10) used in Data Science, as well as chapters on the theory behind Machine Learning, Signal Processing and Computer Vision (Chapters 12, 14 and 18). We've also covered topics related to generating value from data, such as Business Intelligence (Chapter 11) and Data Driven Enterprises (Chapter 21), as well as vital information to help you use data safely, including chapters on the new EU AI Act (Chapter 23) and Trustworthy AI (Chapter 27).

This vast expansion of VDSG's Magnum Opus serves one core purpose:

to give a realistic and holistic picture of Data Science and AI.

Data Science and AI is developing at an incredibly quick pace at the moment and so is its impact on society. This means that responsibilities put on the shoulders of data scientists have grown as well, and so has the need for organizations like VDSG [www.vdsg.at] to get involved and tackle these challenges too.

Let's go for it!

Summer 2024

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 lot 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

Thinking about 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 again during the last months while 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

1

Introduction

Stefan Papp

“I want to be CDO instead of the CDO.”

Iznogoud (adjusted)



Questions Answered in this Chapter:

- How could we describe a fictional company before its journey to becoming data-driven?
- What challenges might such a company need to resolve to become data-driven?
- How will the chapters in this book help you, the reader, to recognize and address such challenges in your own organization?

■ 1.1 About this Book

This book takes a practical, experience-led look into various aspects of data science and artificial intelligence. In this, our third edition, the authors also deeply dive into some of the most exciting and rapidly developing topics of our time, including large language models and generative AI.

The authors’ primary goal is to give the reader a holistic approach to the field. For this reason, this book is not purely technical: Data science and AI maturity depends as much on work culture, particularly critical thinking and evidence-based decision-making, as it does on knowledge in mathematics, neural networks, AI frameworks, and data platforms.

In recent years, most experts have come to agree that artificial intelligence will change how we work and live. For a holistic view, we must also look at the status quo, if we want to understand what needs to be done to meet our diverse ambitions with the help of AI. One useful frame for doing this is to explore how people deal with data transformation challenges from an organizational perspective. For this reason, we will shortly introduce the reader to a fictional company at the beginning of its journey to integrate evidence-based decision-making into its corporate identity. We’ll use this fictional company, in which most things could be more data-oriented but aren’t yet, as a model for outlining possible challenges organiza-

tions may encounter when aiming to become more data-driven. By the end of this book, our hypothetical company will also serve as a model of how a data-driven company could look. In the chapters in between, we'll address many of these challenges and provide practical advice on how to tackle them.

Suppose you, as a reader, would rather not read prose about an invented company in order to learn about such typical organizational challenges. In that case, we encourage you to skip this chapter and start with one that fits your interests. As a holistic book on this field, the authors discuss artificial intelligence, machine learning, generative AI, modeling, natural language processing, computer vision, and other relevant areas. We cover engineering-related topics such as data architecture and data pipelines, which are essential for getting data-driven projects into production. Lastly, we also address critical social and legal issues surrounding the use of data. Each author goes into a lot of detail for their specific field, so there's plenty for you to learn from.

We kindly ask readers to contact us directly to provide feedback on how we can do better to achieve our ambitious goal of becoming the standard literature providing a holistic approach to this field. If you feel some new content should be covered in one of the subsequent editions, you can find the authors on professional networks such as LinkedIn.

And with that said, let's get started.

■ 1.2 The Halford Group

Bob entered the office building of the Halford Group, a manufacturer of consumer products, including their best-selling rubber duck. After crossing the office doors, he felt he was thrown back into the eighties. Visitors having to register at the entrance, filling out forms to declare themselves liable in case of an accident, and promising not to take photos, was only the first step. As Bob entered the elevator, with its brass buttons and glossy, mahogany decor, he could have sworn he'd entered the setting of the movie "The Wolf of Wall Street."

The executive office was similar. The brownish carpets showed their age, and the wallpapers looked like they'd inhaled the smoke of many an eighties Marlboro Man. The worn leather couches and the looming wooden desk (mahogany, again), seemed a memory of a great but distant past. Bob could imagine his dad—a man who had always been proud of being in sales and following the teachings of Zig Ziglar—doing business with this company in his younger years.

This image in Bob's imagination was immediately disrupted when a young woman entered the room, and Bob was immediately thrown back into the present time. With an air of determination, she strode forward to reach for Bob's hand. Somewhat taken aback, he took in the shock of platinum blonde hair, and the tattoos that had not been entirely hidden by her tailored suit, and raised his hand in response. The woman smiled.

1.2.1 Alice Halford – Chairwoman

“I’m Alice Halford,” she said, “I am the granddaughter of Big Harry Halford, the founder of this group. He built his empire from the ground up.”

Bob had read all the legends about the old Halford boss. Every article about him made it clear he did not listen to many people. Instead, “Big Harry” was a proud, determined captain; one who set the course and demanded absolute obedience from his team. Business magazines were yet to write much about Alice, as far as Bob knew. However, he had read one article in preparation for this meeting. Alice was different from the grand old family patriarch, it had said. She had won the succession in a fierce battle against three ambitious brothers, and been selected by the board as chairwoman, thanks to her big plans to transition the company into a modern enterprise that could meet the Zeitgeist of the 21st century.

“Although successful, today’s generation would call my granddad a dinosaur who just wanted to leave enough footprints to let the next generation know he had been there,” Alice said. “Especially in his last years, he was skeptical about changes. Many principal consultants from respectable companies came with heads high to our offices, explaining that our long-term existence would depend on becoming a data-driven company. However, my granddad always had a saying: The moment a computer decides, instead of a founder who knows their stuff and follows their gut, it’s over. All the once proud consultants and their supporters from within the company thought they could convince every executive to buy into their ideas of a modern company, but ultimately, they walked out with their tails between their legs.”

Alice smiled at Bob and continued, “my granddad’s retirement was long overdue, but, finally, his exotic Cuban cigars and his habit of drinking expensive whiskey forced him to end his work life. I took over as a chairwoman of the board. I want to eliminate all the smells of the last century. When I joined, I found parts of the company were highly toxic. My strategic consultants advised me that every large organization has some organizational arrogance and inefficiency. They also cautioned me to keep my expectations low. While many enthusiasts claim that AI will change the world forever, every large organization is like a living organism with many different subdivisions and characteristics. Changing a company’s culture is a long process, and many companies face similar challenges. Ultimately, every company is run by people, and nobody can change people over night. Some might be okay with changes, a few may even want them to happen too fast, but most people will resist changes in one way or another.

At the same time, I understand that we are running out of time. We learned that our main competitors are ahead of us, and if we do not catch up, we will eventually go out of business. Our current CEO has a background in Finance and, therefore, needs support from a data strategist. Bob, you have been recommended as the most outstanding expert to transform a company into a data-driven enterprise that disrupts traditional business models. You can talk with everyone; you have all the freedom you need. After that, I am curious about your ideas to change the company from the ground up.”

Bob nodded enthusiastically. “I love challenges. Your secretary already told me I shouldn’t have any other appointments in the afternoon. Can you introduce me to your team? I would love to learn more about how they work, and their requirements.”

“I thought you’d want to do that. First, you will meet David and Anna, the analysts. Then you’ll meet Tom, the sales director. It would be best if you also talked with the IT manager, Peter—” Alice stopped herself, sighed, and continued. “Lastly, I arranged a meeting for you with our production leader, the complaints department, our Head of Security, and finally with our HR. I will introduce our new CEO, who is flying in today to discuss details at dinner. I booked a table in a good restaurant close by. But it makes sense if you first talk to all the other stakeholders. I had my colleagues each arrange a one-on-one with you. You’re in for a busy afternoon, Bob.”

1.2.2 Analysts

As Alice swept out of the room, a bespeckled man apparently in his mid-forties, and a woman of about the same age, appeared in the doorway. It must have been the analysts, David and Anna. When neither appeared willing to enter the room first, Bob beckoned them inside. He was reminded of an empowerment seminar he’d attended some years ago: The trainer had been hell bent on turning everyone in the workshop into strong leaders, but warned that only the energetic would dominate the world. These analysts seemed to be the exact opposite. David laughed nervously as he entered, and Anna kept her eyes lowered as she headed to the nearest seat. Neither seemed too thrilled to be there; Bob didn’t even want to imagine how they would have performed in that seminar’s “primal scream” test.

David and Anna sat down, and Bob tried to break the ice with questions about their work. It took him a while, but finally, they started to talk.

“Well, we create reports for management,” David said. “We aim to keep things accurate, and we try to hand in our reports on time. It’s become something of a reputation,” he added with a weak chuckle.

Bob realized that if he was going to make them talk, he’d need to give his famous speech, summarized as, “your job in this meeting is to talk about your problem. Mine is to listen.” After all, he needed to transform Halford company into a data-driven company, and they were ones working closest with the company’s data.

Bob finished his speech with gusto, but Anna merely shrugged. “The management wants to know a lot, but our possibilities are limited.”

Bob tried his best to look both in the eyes, though Anna turned quickly away. “But what is it that prevents you from doing your work without any limits?”

“Our biggest challenge is the batch process from hell,” David spoke up suddenly. “This notorious daily job runs overnight and extracts all data from the operational databases. It is hugely complex. I lost count of how often this job failed over time.”

Got them, Bob thought, nodding in encouragement.

“And nobody knows why this job fails,” Anna jumped in. “But when it does, we don’t know if the data is accurate. So far, there has never been a problem if we handed in a report with questionable figures. But that’s probably because most managers ignore the facts and figures we provide anyway.”

“Exactly!” David threw up his hands. Bob started to worry he had stirred up a hornet’s nest.

“When a job fails, it’s me who has to go to IT,” David said. “I just can’t hear anymore that these nerds ran out of disk space and that some DevSecOps closed a firewall port again. All I want is the data to create my reports. I also fight often with our security department. Sometimes, their processes are so strict that they come close to sabotaging innovation. Occasionally, I get the impression they cut access to data sources on purpose to annoy us.”

“Often, we are asked if we want something more sophisticated,” Anna said, shaking her head in frustration. “It is always the same pattern. A manager visits a seminar and comes to us to ask us if we can ‘do AI’. If you ask me honestly, I would love to do something more sophisticated, but we are afraid that the whole system will break apart if we change something. So, I am just happy if we can provide the management with the data from the day before.”

Don’t get us wrong, ML and AI would be amazing. But our company must still master the basics. I believe most of our managers have no clue what AI does and what we could do with it. But will they admit it? Not a chance.”

Anna sat back in a huff. Bob did not need to ask them to know that both were applying at other companies for jobs.

1.2.3 “CDO”

At lunch break, a skinny man in a black turtleneck sweater hurled into the office. He seemed nervous, as if someone was chasing him. His eyes darted around the room, avoiding eye contact. His whole body was fidgeting, and he could not keep his hands still.

“I am the CDO. My name is Cesario Antonio Ramirez Sanchez; call me Cesar,” he introduced himself with a Spanish accent.

Bob was surprised that this meeting had not been announced. Meanwhile, his unexpected visitor kept approaching a chair and moving away from it again as if he could not decide whether to sit down or not.

“CDO? I have not seen this position in the org chart,” Bob answered calmly, “I have seen a Cesario Antonio Rami ...”

“No no no ... It’s not my official title. It is what I am *doing*,” Cesar said dramatically. “I am changing the company bottom up, you know? Like guerilla warfare. Without people like me, this company would still be in the Stone Age, you see?”

“I am interested in everyone’s view,” Bob replied, “but I report to Alice, and I cannot participate in any black ops work.”

“No, no, no ..., everything is simple. Lots of imbeciles are running around in this company—” Cesar raised his finger and took a sharp breath, nodded twice, and continued. “I know ... HR always tells me to be friendly with people and not to say bad words. But we have only data warehouses in this company. Not even a data lake. Catastrófica! Its the 21st century, and these dinosaurs work like in Latin America hace veinte años. Increíble!”

He took another breath, and then continued. “Let’s modernize! Everything! Start from zero. So much to do. First, we must toss these old devices into the garbage, you know? And replace them with streaming-enabled PLCs. Then, modern edge computing services streams

everything with Kafka to different data stores. All problems solved. And then we'll have a real-time analytics layer on top of a data mesh."

Bob stared at his counterpart, who seemed unable to keep his eyes or his body still for more than a moment. "I am sorry, I do not understand."

"You are an expert, you have a Ph.D., no? You should understand: modern factory, IoT, Industry 4.0, Factory of the Future."

Bob decided not to answer. Instead, he kept his eyebrows raised as he waited for what Cesar would say next.

"So much potential," Cesar went on. "And all is wasted. Why is HR always talking about people's feelings? Everything is so easy. This old company needs to get modern. We don't need artists, we need people with brains. If I want art, I listen to Mariachi in Cancun. If current people are imbeciles, hire new people. Smart people, with Ph.D. and experience. My old bosses in Latin America, you cannot imagine, they would have fired everyone, including HR. Let's talk later; I'm in the IT department en la cava."

Bob had no time to answer. Cesar left the room as fast as he had entered it.

1.2.4 Sales

A tall, slim, grey-haired man entered the room, took a place at the end of the table, leaned back and presented to Bob a salesman grin for which Colgate would have paid millions.

"I am Tom Jenkins. My friends call me 'the Avalanche'. That's because if I take the phone, nobody can stop me anymore. Back in the nineties, I made four sales in a single day. Can you imagine this?"

I get it; you are a hero. Bob thought. *Let's turn it down a bit.*

"My name is Bob. I am a consultant who has been hired to help this company become more data-oriented."

Tom's winning smile vanished when Bob mentioned 'data.'

"I have heard too much of the data talk," Tom said. "No analysis can beat gut feeling and experience. Don't get me wrong. I love accurate data about my sales records, but you should trust an experienced man to make his own decisions. No computer will ever tell me which potential client I should call. When I sit at my desk, I know which baby will fly."

"With all due respect. I can show you a lot of examples of how an evidence-based approach has helped clients to make more revenue."

"Did you hear yourself just now?" Tom answered, "Evidence-based. You do not win sales with brainy talks. You need to work on people's emotions and relationships. No computer will ever do better sales than a salesman with a winning smile. I'll give you an example: One day, our sales data showed that we sold fewer products in our prime region. Some data analysts told me something about demographic changes. What a nonsense!

So, I went out and talked to the people. I know my folks up there. They are all great people. All amazing guys! Very smart and very hands-on. I love this. We had some steaks and beers, then I pitched our new product line. Guess who was salesman of the month after that?

No computer needs to tell me how to approach my clients. So, as long as we get the sales reports right and we can calculate the commission, all is good. It is the salesman, not the computer, who closes a deal.”

With that, The Avalanche was on his feet. He invited Bob to a fantastic restaurant—“I know the owner and trust me, he makes the best steaks you’ll ever taste!”—and was gone.

1.2.5 IT

Ten minutes past the planned meeting start time, Bob was still waiting for the team member he had heard most about upfront: the IT leader, Peter. His name had been mentioned by various people multiple times, but whenever Bob had asked to know more about him, people were reluctant to answer, or simply sighed and told him, “you’ll see.”

Finally, Peter stormed into the room, breathless and sweating. “This trip from my office in the cellar to this floor is a nightmare,” he said between gasps. “You meet so many people in the elevator who want something. I am constantly under so much stress, you cannot imagine! Here, I brought us some sandwiches. I have a little side business in gastronomy. You need a hobby like this to survive here. Without a hobby in this business, you go mad.”

Peter was a squat, red-faced man, who’d been with Halford since he was a lot younger, and had a lot more hair. He sank a little too comfortably in his chair, with the confidence of a man who’d been around so long, he was practically part of the furniture.

He doesn’t lack confidence, that’s for sure, Bob thought. I wonder how many dirty secrets this man has learned over the years that only he knows.

“Okay, let’s talk about IT then,” Peter sighed after Bob turned down the sandwiches. “My colleagues from the board and the executives still don’t get what it is they’re asking of me daily. When they invite me to meetings, I often do not show up anymore. We are a huge company, but nobody wants to invest in IT. I am understaffed; we hardly manage to keep the company running. Want to go for a cigarette?”

“No, thank you,” Bob said, but Peter was already crumpled pack from his trouser pocket. He rambled all the way to the smoker’s chamber, bouncing around from one topic to another. Bob learned everything about Peter, from his favorite food over his private home to his hernia, which was apparently only getting worse. Once Peter got first cigarette into his mouth, he went back to the topic Bob was really interested in.

“The suits want things without knowing the implications. On the one hand, they want everything to be secure, but then again, they want modern data solutions. Often, they ask me for one thing one day, and then the very next, they prioritize something else. To be blunt, I had my share of talks with these external consultants. If I allowed them to do what they asked me to do, I could immediately put all our data on a file server and invite hackers to download it with the same result. To keep things working, you need to firewall the whole company,” Peter stubbed out his cigarette, and reached for another.

Bob leaped at the chance to interject. “Can you tell me more about your IT department? I was looking for some documentation of the IT landscape. I have not found much information on your internal file shares. Which cloud provider are you currently using?”

Peter laughed and then started coughing. Tears in his eyes, he answered. “I told you, I’m understaffed. Do you really think I have time to document?” He pointed to his head. “Don’t worry, everything is stored in the grey cells up here. And we have a no-cloud strategy. Cloud is just a marketing thing if you ask me. When we build by ourselves, it is safer, and we have everything under control.

If I just had more people ... Did you meet one of my guys, Cesar? He is also okay when he does not talk, which unfortunately doesn’t happen often. I don’t like when people think they are smarter than me. He doesn’t know Peter’s two rules yet. Rule Number 1: Do not get on your boss’s nerves. Rule Number 2. Follow rule number 1.”

Peter laughed, flicked the second cigarette on the ground, and retrieved a bag from his other pocket. It was full of caramels: Peter popped one into his mouth and continued, chewing loudly. “Alice asked me if I could introduce you to Bill, my lead engineer, but I declined. This guy has the brains of a fox but the communication skills of a donkey. He also gets nervous when you look him straight in the eyes. I am always worried that he might wet his pants— Or am I being too politically incorrect again? Our HR keeps telling me that I should be more friendly. But in this looney bin, you learn to let out your stress by saying what you think. So, please excuse my sarcasm. I am the last person standing between chaos and a running IT landscape, the management keeps getting on my nerves with stupid requests, and last but not least, the HR department is more concerned about how I communicate than about finding the people who could help me keep our company running.”

It took a couple of attempts until Bob could finally break free from Peter’s complaining to head to his next meeting. Even as he was leaving, Peter repeatedly called on Bob to visit his food business sometime, where they could have a drink in private, and Peter could share his Halford ‘war stories’ more openly.

1.2.6 Security

While waiting for the HR representative, Bob received a voice message from Suzie Wong, the head of data security. When Bob played it, he heard traffic sounds in the background.

“Apologies for not showing up. School called me in as one of my kids got sick. I hope a voice message is fine. I am Suzie Wong. I have been with Halford for years. They call me the human firewall against innovation. I take this as a compliment because, in some way, it means I am doing my job well. Could any company be happy with a Head of Security who takes her job easy? My predecessor was more laid back than I am. He was in his fifties and got a little too comfortable, thinking he would retire in a secure job. And then one day ... there was this security breach. His kid’s still in private school, he’s suddenly without a job and, well, I’ll spare you the details.

People often think I’m only around to sign off on their intentions to use data, but my real job is protecting our client’s privacy. Data scientists must prove to me that our client’s data is safe when they want to work with it. Unfortunately, too many take that too lightly.

If the requestor follows the process, a privacy impact assessment could be done within a week. I will send you a link to our security portal later so you can review it. You’ll see for yourself that we do not ask for anything impossible.

I am the last line of defense, ensuring that we do not pay hefty fines because someone thought it was just data they were playing around with. Some people also jokingly call me ‘Mrs.No,’ because this is my common answer if you cannot express why I should grant you security exceptions or provide access to data containing clients’ private information. Some people complain that this way, it may take months to get security approval. But so long as engineers and data scientists still don’t get how to address security matters correctly, I don’t care if it takes years before I give my final OK.

Anyway, excuse me now, I’m at the school ...”

1.2.7 Production Leader

Bob had some time before his next meeting and looked up his next meeting partner online. He discovered a middle-aged man with a long history on social media, including some questionable photos of his younger self in a Che Guevara t-shirt. Bob chuckled. That young man could be happy that their interview wasn’t taking place during the times of the Cold War.

Finally, Bob’s interviewee entered the room. He was muscular, and his bushy black beard showed the first signs of greying.

“My name is Hank. Pleased to meet you,” he said with a deep voice.

“I heard you are new in your position,” Bob said.

“Yes. Alice fired my predecessor because he was a tyrant. I am now one of the first of what she calls ‘the new generation.’ I accepted because I can change things here now. Let me get to the point: What are you planning to do?”

Bob smiled and said, “the idea in factories is often to use machine learning for automation. Think of processes where people check the quality of an item manually. Imagine that you can automate all this. A camera screens every piece, and defective items – which we call ‘rejects’ – are filtered out automatically.”

Hank stiffened. “My job is to protect jobs, not support removing them. Some of our factories are often in villages, where they are the only source of work.”

“Almost every country goes through demographic changes. Can you guarantee that you will be able to maintain a strong enough workforce to keep the factories running? How about doing the same with fewer people?”

“But if you remove a few people, they can end up out of work,” Hank said. “What if you don’t need workers at all in a few years? I don’t want to open the door to a system that makes the bourgeoisie richer and put the ordinary proletarian out of work.”

“That is very unlikely,” Bob said.

“I see you are solidary with your employees, Hank. Did you consider exploring use cases to protect them? We can use computer vision to see if factory workers wear helmets, for example.”

Hank looked deeply into Bob’s eyes. Bob couldn’t quite tell if it was a good or bad sign, but he did realize something: this was not a man he’d like to meet on a dark, empty street.

“I understand that there might be benefits for my colleagues,” Hank said. “I just want to open up a trojan horse: I get one IT system in to prevent accidents, and the next one makes the workers obsolete. But I promised Alice I’d support her. She is a good person. I will talk with my colleagues. I need to get them on board, but one thing is not negotiable: We will never tolerate any system that completely replaces people who need the job they have.”

1.2.8 Customer Service

The next interviewee, an elderly woman with perfectly glossy, silver hair, entered the room. She sat down and carefully ran her fingers over classic French bun, ensuring not a hair was out of place.

“I am Annie from the complaints department,” she said with something of an aristocratic tone. She seemed more interested in her neatly manicured nails than Bob as she went on. “I honestly do not know why you want to talk to me.”

“Well, part of a data-driven enterprise is often also a customer-first strategy. We can measure customer churn and other metrics through data. Most of my clients want to use data to maximize success. They even renamed their departments to ‘Customer Satisfaction Department’ to underline this.”

“Aha,” Annie said. There was an uncomfortable silence as she polished the face of her antique watch with her other sleeve.

Bob cleared his throat, anxious to get her attention. “Would you be interested to learn more about your customers through data?”

“Why should I?”

“To serve them better?”

“We have sturdy products. Most complaints have no base. We believe the less money we spend on confused customers, the more we have left to improve our products. This is what I call the real customer value we provide.”

Ah-hah. Bob recognized the famous argument against investing in any domain that doesn’t directly create revenue. *She probably gets a bonus for keeping yearly costs low*, he thought, seeing an opportunity.

“And how do you keep costs small at the moment?”

“We have an offshore call center. They handle about 80% of calls, although a lot of those customers just give up, for some reason. The remaining 20% are forwarded to a small team of more advanced customer support employees. I know it sounds harsh, but you cannot imagine how many confused people try to call us without having a problem at all. Some – it seems – call us just to talk.”

“Right. And have you thought of the possibility to reduce costs by building chatbots backed by generative AI? There are also many ways to use data science to filter customer complaints. If properly trained, your clients get better support, and you reduce costs.”

“Would it be good enough to shut down the offshore center?”

Gotcha. “If done right, yes.”

For what felt like the first time, Annie looked at Bob directly. “How much would it cost?”

“At the moment, it is still difficult to estimate.”

Annie thought a while, then stood up to leave. At the door, she paused. “Once you know, call me immediately.”

1.2.9 HR

“I’m, I’m Pratima,” came a woman’s voice at the door. She approached Bob, looked up at him with a welcoming smile and asked, “how can I help you, Bob?”

“Hi, Pratima. Let’s take a seat. As you know, I’m here to transform this company into a more data-oriented one. I saw on LinkedIn that you have previously worked for very modern companies with a strong data culture. How is it now to work for a company at the beginning of its journey?”

“Alice asked me to be open to you. I took this job as a career step to advance to leadership. However, the Wheel of Fortune led me to more challenges than expected.

In my previous job, we had the vibes to attract new talent. It was an environment primed for excellence: fancy office spaces, a modern work culture with flat hierarchies, cool products to work on, and many talented, diverse colleagues. Recruiting was easy because new candidates felt it the spirit of our community.”

Pratima sighed.

“In this company, though, we cannot hide that we are at the beginning of our transition. Applicants usually have many offers to choose from. Sometimes, we have to watch perfect candidates walk away because we do not yet provide a warm and welcoming environment for data professionals.

When managers discuss AI and data transition, some might oversee the human aspect. What if you create the perfect data strategy but cannot attract enough talent? Many companies face this problem, and an elephant is always in the room. To become a data-driven company, you have to create an environment that attracts people who think differently, and this means changing your culture.

“Do you believe management is scared to promote too much change because it is afraid to lose everything?”

“I understand that some seasoned employees might get disappointed and even resign if their comfortable environment starts to modernize. But at the same time, if you do not change at all, you are stuck in the mud, and your competition will make you obsolete. The Dalai Lama says we should be the change we wish to be.”

“Right. And I believe it was Seneca who once said, ‘It’s not because things are difficult that we dare not venture. It’s because we dare not venture that they are difficult.’”

“True! But I have to go now. I am looking forward to continuing our talks.”

1.2.10 CEO

Alice and Bob met at a fusion restaurant downtown in the evening. Alice introduced Bob to Santiago, the long-time CFO turned new CEO. After an excellent meal, they ordered some famous Armenian cognac, and got down to the real discussion.

“I’ll be honest with you, Bob,” Santiago began. “All your ideas to transform Halford sound fantastic, but as an economist and a numbers person, my first question is, how much will this all cost?”

Oh boy. Bob was prepared for the question, but he knew Santiago wouldn’t like the answer. “It depends,” he said, and Santiago looked about as dissatisfied as Bob would have expected.

“I understand that everyone looks at the costs,” Bob continued, “but history is full of companies that failed to innovate and went bankrupt as their competition moved forward. If you see the full spectrum of artificial intelligence, hardly any company will eventually operate as before.”

“Some companies recommend that we start with data literacy workshops to enable leaders to interpret data and numbers efficiently. Literacy sounds as if they want to teach us to read and write again—and for a huge amount of money, of course. Don’t get me wrong, please. I understand that we need to innovate, but if I approve everything consultants suggest to me, we will soon be broke.”

“But if your leadership team cannot ‘think in data,’” Bob said, making air quotes as he spoke, “how do they expect to attend our planned strategy workshop on exploring specific data science options for our business goals?”

“What is the difference?”

“In the data literacy workshops, we aim to create an understanding of how to interpret data. In the strategy workshop, we’ll create a list of use cases to improve processes in your company, and prioritize them, to integrate new data solutions gradually.”

“I understand that we have some tough nuts to crack. Some of our employees do not believe in becoming data-driven, and we may need to invest hugely in Enablement. We once asked external companies to help us modernize our IT. No consulting company gave me a quote with a fixed price for a transition project. They always said we were facing a hole without a bottom.”

“Leadership is the only way to move forward. If the executive team is convinced and aligned, this culture can spread.

Your operational IT will need to mature and modernize gradually. However, be aware that an analytical layer can be built outside of corporate IT. One risk is to make data transition to an IT problem; IT is part of it, but becoming a data-driven company is far more than giving some engineers a job to build platforms.”

“For me, it’s clear,” Alice said. “Either we modernize, or we gradually fade out of existence. Bob, what do you need to help us?”

Bob looked from one to the other, carefully considering his next words. “Becoming data-driven does not mean hiring a bunch of data scientists who do a bit of magic, and suddenly the company makes tons of money using AI. As I said, the first step is to align the stakeholders. For me, this is the alpha and omega of AI: creating a data culture based on critical thinking and evidence-based decisions.”

“Great,” answered Alice. “Let’s get started with that.”

■ 1.3 In a Nutshell



Expectation Management

Most companies see the need to become data-driven, as they understand that those organizations that ignore technical evolution mostly fail.

Some employees might have unrealistic expectations about how fast a transition can go. We highlight that changing to a data-driven company is not just a change of practices and processes, it is often a cultural overhaul of how the company does its business.

Many employees fear having to give up some of their autonomy, or even losing their jobs to computers entirely, if AI is introduced at their company. An organization that transitions to become data-driven must address this.

Technology Focus and Missing Strategy

Some companies try to find a silver bullet that solves all problems. “We’ll just use this technology, just apply AI in this or that way, and all our problems are resolved,” they think. Being too technology-focused, however, is an anti-pattern that can hinder a company’s evolution to becoming data-driven.

Data Science and AI are about more than just Understanding Frameworks and Methods

While it is essential to have a team of skilled data scientists and AI engineers to pick the right AI frameworks and build complex AI systems, for large organizations, there are many other considerations to watch for. Not being able to understand the needs of an organization and where AI can make a difference is a risk. With the wrong target, every strategy will fail.

Collaboration between Analysts and IT

In some companies, IT provides the platforms that analysts have to use. If these platforms are error-prone or old, it can get frustrating for analysts. In modern environments, not all analytical platforms must be managed by one central IT department. This can give data teams more freedom to operate on their own.

IT

Many IT teams lack the resources to build the data pipelines needed for data science platforms. Often there is a gap between business users and engineers, making it hard for them to communicate with each other.

IT, especially operations, is often focused on preventing problems. As a result, many strive to protect their systems from change. They want to make it difficult to access data in order to keep platforms secure. Data scientists, however, would like to access data easily, to make progress quickly. This can lead to friction between both teams.

Costs

Introducing machine learning, data science and artificial intelligence can be expensive. It is rarely easy to say how much data science will impact the company's results: the relationship between inputs (such as time, effort, and resources) and outputs is anything but deterministic. The alternative, however, is even more grim. A company that is not ready to invest in innovation, will eventually lose its competitiveness and risk bankruptcy.

Data and Privacy Protection

Data and Privacy Protection may slow down some projects and make them more bureaucratic, but they are absolutely necessary. In addition, it's most likely that nobody wants to live in a system where privacy is not respected. In a day-to-day job, privacy protection is process-driven: Making these processes transparent and efficient.

Hiring

Introducing data science may require more significant changes in the corporate structure or culture, which could reveal hidden conflicts and challenges nobody wants to talk about.

Data professionals are a rare breed, and as there are few of them on the job market, how can a company even think about change without the required skills? Attracting engineers and scientists often requires an offer that goes beyond free fruit in the office.

16

Generative AI and Large Language Models

Katherine Munro, Gerald Hahn, Danko Nikolić



Questions Answered in this Chapter:

- What is Generative AI and how does it relate to other AI and Machine Learning techniques?
- Which characteristics define and differentiate Large Language Models?
- How can you use prompt engineering to get the best out of Generative AI models?
- Which design patterns are useful for building LLM-based applications?
- How can you customize an LLM to improve performance on your specific use case?
- Which vulnerabilities and limitations of Generative AI should you be aware of?
- How can you build robust, reliable, and effective Generative AI-powered applications?

■ 16.1 Introduction to “Gen AI”

The first machine learning models were made to be predictive. They were designed to take an input and generate an output, often expressed as a category or numeric value. For example, we might ask a model to identify the category of an object in an image, or to predict the most likely price of a stock a week from now. The goal, ultimately, was to assign new information to some existing piece of data.

Generative models, by contrast, are designed to generate entirely new data. They can create rich outputs in various modalities, including text, images, and video. So, instead of asking a **predictive model** to classify an image as containing a cat or a dog, for example, we can ask a generative model to create an image depicting a cat, or a dog, or just about anything else we can imagine.

Generally, creating such complex outputs is a much more difficult task, due to the great number of intricate dependencies between even the minutest components of each output. To generate an image, for example, a model cannot simply spit out each of the millions of required pixels independently. Instead, the content generated in one part of the image will depend on what has been generated everywhere else. For example, in real-life photographs of building interiors, paintings tend to hang on the walls; they do not lay on floors, nor are they attached to ceilings. Such “rules” are obvious to us, of course, because we’ve learned them through a lifetime of observing different interiors for ourselves. A **Generative AI** model needs to learn such rules in a similar way: by being exposed to thousands of images depicting different types of buildings and rooms.

Learning such a huge number of complex dependencies requires a huge amount of input examples. Hence, a generative model usually requires a lot more training data than a predictive one. The “rules,” or dependencies, can be understood as multi-dimensional distributions that need to be approximated. These distributions indicate the likelihood that a certain combination of features occurs in the data. For example, pixels representing a photo frame will occur more often in combination with pictures representing a wallpapered wall than they do with pixels representing a carpeted floor. Once an image generation model has learned such dependencies between all sorts of possible combinations of all sorts of possible pixel features, it can generate images which obey them. And using this principle, we can create different modalities of generative models by exposing the right machine learning algorithms to different kinds of inputs: texts, programming code, music, biological sequences, and more.

With enough data and enough computational power, it is possible to achieve impressive results with Generative AI (hereafter referred to interchangeably with “Gen AI”). The learned approximations of feature distributions are usually not perfect, of course, but neither is **predictive AI**. Much like predictive AI may mistake a chihuahua for a muffin in an image classification task, for example, a generative model may create images that would be impossible in real life. When generating images of humans, for instance, Gen AI has a hard time sticking to five fingers on each hand, often drawing four or six fingers instead. But despite such glitches, Gen AI has turned out to be immensely useful, and has vastly widened the general applicability of AI in everyday life.

■ 16.2 Generative AI Modalities

As we just mentioned, generative AI algorithms can be used to train all kinds of “content generators”, provided that enough data are available. For example, in the audio domain, we can generate spoken voice or music. In chemistry and biology, AI can generate molecular

structures and protein sequences. Time series and graphs are yet two more interesting possibilities, and so the list goes on. There is no limit to the type of modality to which Gen AI algorithms can be applied.

When we talk of “modality”, we don’t just mean the type of output that an AI model generates. The inputs that trigger the generation are equally relevant, and can be equally varied. The simplest models, known as **uni-modal models**, use the same mode of data for both inputs and outputs. For example, in Large Language Models, which we’ll discuss in detail later, text input is used to generate text outputs.

We can also combine modalities to create **multi-modal models**. Textual inputs can be used to generate images, or, conversely, we can generate text outputs (descriptions) based on image inputs. Information from a piece of text can be converted into a graph, and graphs can be described in natural language. Even the inputs and outputs themselves can be multi-modal. For example, a model may be fed an image and some instructions describing what needs to be done with that image, and it will output an image with the required changes. It could even produce audio or visual content to match the input: it all just depends on how the model was trained.

Much like generative models need to be trained separately for different modalities, it is often useful to separately train them for specific domains within one specific modality. For example, computer code is nothing but text. Nevertheless, we’re likely to get a better code generation model if we train a model specifically for that task, and feed it only inputs that include code (perhaps along with explanations and documentations of code). Similarly, it is a good idea to separately create generative models for videos and images, even though a video is technically nothing more than a series of images.

The landscape of common tools and publicly available models is changing quickly. Below we list a few popular examples (at the time of writing) for the various modalities we’ve just discussed:

- **Text:** ChatGPT, Bing Copilot, Gemini, LLaMA, Claude
- **Images:** Imagen, Stable Diffusion, Midjourney, DALL-E
- **Music:** MusicLM, Soundraw.io, Amper Music, Humtap, Stable Audio
- **Video:** D-ID, Gen-2 from Runway, Pictory, Synthesia, Fliki, Sora
- **Code:** GitHub Copilot, Codey, Tabnine, Polycoder, DeepCode

One may wonder how combining all these diverse modalities into single models is possible. The secret lies in the **encoder-decoder architecture** of multi-modal foundation models (discussed in the previous chapter). The encoder takes the input and transforms it into an intermediate representation, known as an “embedding”, which is then used as input to the decoder. A nice property of this intermediate embedding representation is that it is independent of the modality. The embedding is said to describe the semantics of the input: that is, its meaning. For example, you could provide a multi-modal model with either the text, “a horse is passing near an oak tree,” or with an image depicting just that. In either case, the algorithm may generate very similar embeddings, under the hood. This is how AI abstracts the inputs from different modalities. It uses two different encoder models, one for text and another for images, with both encoders being trained to work with the same embedding space. Similarly, the same AI may pass these intermediate representations to different decoder models, each trained for a different modality. One may generate sounds, the other images,

and yet another one, texts, but all are working in the same embedding space. Therefore, the trick of multi-modal generative AI lies in the encoder-decoder structure.

16.2.1 Methods for Training Generative Models

While we hope to have made the general concept of training generative models clear, we cannot possibly squeeze in a discussion of all types of architectures and algorithms within this chapter: such a task would fill a book in itself! Those who wish to understand more about machine learning foundations can visit Chapter 12, which even deep dives into two types of models commonly used for image generation: Generative Adversarial Networks (12.6.10) and Autoencoders (12.6.8). Large Language Models, including how to train and fine-tune them, will be discussed next. So, as you read on, remember that virtually anything which can be treated like a text sequence can potentially be modelled via LLMs: this includes text, code, protein sequences, and much more.

■ 16.3 Large Language Models

16.3.1 What are “LLMs”?

Since the launch of OpenAI’s **ChatGPT** at the end of 2022, the landscape of Artificial Intelligence text generation has undergone a remarkable transformation. This groundbreaking development marked the rise of accessible AI-powered text generation, captivating the public’s imagination and sparking widespread interest. Built upon the foundation of **Large Language Models** (LLMs), ChatGPT represented a significant leap in the capabilities of AI-generated text. While earlier models (such as ELMo and BERT, which will be discussed in Chapter 17) had certainly exhibited much promise, they were primarily of interest to academics and specialists; ChatGPT, on the other hand, rapidly became a tool for anyone to use and explore.

Since its beginning, the field of **LLMs** has evolved rapidly, with companies racing to develop ever more sophisticated and powerful models. This fast pace of innovation has driven the technology forward at an unprecedented rate, pushing the boundaries of what was once thought possible in the domain of AI-driven text generation.

Conceptually, a language model is an Artificial Intelligence system designed to predict the next word in a sequence based on the preceding words or context. In their early implementations, these models made their predictions based on only a few preceding words; Now, thanks to advancements in machine learning and, in particular, neural network algorithms, language models can consider vast sequences of words, leading to more accurate predictions.

Mathematically, the prediction of the next word is framed as calculating conditional probabilities. This involves assessing the likelihood of a specific word occurring given the context provided by the preceding words. The word with the highest probability within the model’s

vocabulary is then chosen, in a process known as **sampling**. The chosen word is then appended to the existing text, and the process repeats iteratively until a predetermined stopping condition is met. This **autoregressive mechanism**, also known as **causal generation**, ensures that the model only considers previously generated words and not those predicted afterward, as would be the case in bidirectional approaches.

Over time, language model methodologies have undergone significant changes. Initial approaches relied on simple n-gram models before transitioning to **neural network architectures** such as feedforward networks [1], convolutional [2] and recurrent neural networks [3], including variants like long-short term memory networks [4]. However, the most significant advancement came with the introduction of **Transformer models** [5], which revolutionized the field with their self-attention mechanism. Their architecture allows models to consider a broad context within a sentence, significantly enhancing their predictive capabilities. Chapter 17, Natural Language Processing (NLP), presents all these algorithms in detail, showing how each new development built upon previous successes to bring us to the revolutionary moment NLP is enjoying today.

More recently, newer architectures like **Eagle** [6] and **Mamba** [7] have demonstrated competitive performance without relying on attention mechanisms or employing state-based models. These models promise comparable performance to traditional Transformer-based LLMs of similar size, while being more computationally efficient and enabling faster inference.

The ability of language models to predict the next words and generate coherent text stems from their training on vast amounts of data sourced from the internet. Through learning the statistical probabilities inherent in language patterns, these models become adept at generating meaningful responses to queries and crafting creative pieces of text, such as emails or stories. With recent advancements, they've also gained limited capability to reason through complex tasks (see Section 16.3.3.3).

Large Language Models vary in several key aspects, which contribute to differences in their capabilities:

Performance on benchmark tasks: LLMs are often evaluated based on their performance on benchmark tasks, which serve as standardized tests to measure their effectiveness. Examples of benchmark tasks include language understanding tasks like question answering and text classification. LLMs may excel in certain tasks while performing less optimally in others, depending on their design and training.

Open source vs. closed source: LLMs can be further classified as either open source or closed source. Open-source models provide access to their architecture and parameters, allowing researchers and developers to modify and fine-tune them for specific applications. Closed-source models, on the other hand, restrict access to their internal teams and are typically only provided to the public as pretrained models through APIs or licensed software.

Number of parameters: LLMs come in different sizes, usually quantified by the number of parameters they possess. Small LLMs ("SLMs") might contain a few billion parameters, whereas larger models can encompass hundreds of billions. Generally, the parameter count correlates with the model's complexity and capacity to capture nuances in language (though much research is attempting to test the limits of this relationship).

Algorithms: LLMs leverage different algorithms for text generation, with the Transformer architecture being the most prevalent. Newly developed variations of the original attention mechanism (such as [8]) can enhance a model's ability to capture long-range dependencies and contextual information. Additionally, some LLMs may utilize alternative architectures, such as recurrent neural networks or state space models.

Training data: The size and quality of the training data significantly impact the performance of LLMs. Models trained on larger and more diverse datasets tend to exhibit superior performance due to their exposure to a broader range of linguistic patterns and contexts. Even models with the same parameter count can demonstrate substantial differences in performance based on the quality and quantity of their training data. Fine-tuning existing models on specialized datasets (see Section 16.3.4.2) can further enhance their performance for specific tasks, boosting it beyond that of larger, general-purpose models.

16.3.2 How is Something like ChatGPT Trained?

Pre-training

Off-the-shelf Large Language Models undergo a comprehensive training process before they are made available to users through APIs or downloadable via platforms like HuggingFace. This process starts with an initial **pre-training** phase, where the goal is to train the model to predict words in a sequence. Often this is done by exposing the model to an extensive amount of text data scraped from the internet, randomly masking out words, and having the model fill the gaps. Initially, the model's predictions will be quite random. But as training progresses, guided by an objective to minimize errors in the model's next-word predictions, the model's parameters are iteratively adjusted until it has learned to capture the intricate syntactic and semantic relationships in the text. In other words, through exposure to natural language data, the model gradually gains a statistical “understanding” of which words make sense in different contexts, and in combination with which other words.

Such an approach is known as **self-supervised learning** [9] because it enables the model to learn from data without requiring external labels, relying instead on the inherent structure of the text itself. Unlike in supervised learning, no explicit labels are provided during this process, as the missing words are already known from the text. The quantity and quality of this data are paramount, however, as they directly impact the model's ability to learn and generalize from the information provided.

Instruction Fine-tuning

Pretraining LLMs is a fundamental step in their development, providing them with solid foundational knowledge (hence the term “Foundation Models”, the title of the previous chapter). However, simply predicting missing words in a text doesn't necessarily serve much purpose in the real world, which is why we next conduct a supervised learning step, known as **instruction fine-tuning** [10]. The purpose of this phase is to train the model to follow human instructions more precisely. This is achieved by providing the model with labeled data consisting of examples that illustrate how it should respond to specific questions or instructions. Through exposure to explicit examples provided by humans, the model learns to better understand and execute tasks according to human expectations.

Reinforcement Learning from Human Feedback

This phase, though effective, still leaves one problem unsolved: the potential for LLMs to generate outputs that violate human values, resulting in toxic, harmful, or biased responses. To mitigate this problem, reinforcement learning techniques are used in combination with human feedback [11]. In this approach, known as **Reinforcement Learning from Human Feedback** (RLHF, see also Section 16.3.4.2.3), the model receives feedback on the appropriateness of its responses, and adjusts its behavior during training accordingly. This helps ensure that the model's outputs align with societal norms and values.

This combination of pre-training and fine-tuning via instructions and human feedback have been crucial to the success of models like ChatGPT and its successors. The first stage provides the model with broad, general language knowledge, and the latter two help make it particularly adept at enacting human requests. If you're thinking that all these training stages sound complex and resource intensive, however, you're right. Pretraining demands significant computational resources and cutting-edge hardware infrastructure, including high-performance computing clusters and specialized processing units optimized for deep learning tasks. As a result, only a handful of companies worldwide have the infrastructure and expertise necessary to pretrain LLMs. Instruction fine-tuning and RLHF are more feasible for smaller organizations, but nevertheless require carefully curated datasets and significant manual effort. Fortunately, many off-the-shelf LLMs have already been pretrained and fine-tuned, making them useful for many applications straight out of the box.

In the following section, we'll discuss ways to use LLMs directly. Then, in Section 16.3.4, we'll discover how to further customize them for your specific needs.

16.3.3 Methods for Using LLMs Directly

For many people, their first interaction with LLMs was directly through a chat interface, such as ChatGPT. Even development teams may start an LLM initiative by testing their idea directly with a publicly available chatbot, which, given the right instructions, can already achieve an impressive number and variety of tasks. When more is required of the LLM, however, two key design patterns often come into play: augmenting the LLM with additional information to help broaden its knowledge base, and providing it with access to tools, with which to execute more complex tasks. The following three subsections explore at each of these options in turn.

16.3.3.1 Direct Interaction via Prompting

16.3.3.1.1 Zero- to Few-Shot-Inference

The public success of LLMs is not solely attributed to their performance on NLP benchmarks, but also, to the way they enable human interaction. With traditional machine learning models, programmers wrote the code for training the model and directing it to perform tasks. Consequently, proficiency in programming languages was essential. However, LLMs operate differently. They allow direct interaction for everyone. These AI systems can learn from and respond to instructions, known as “**prompts**,” given in everyday language, whether spoken or written. Their responses, called “**completions**,” are also given in natural language,

as if the user and the “bot” were engaging in ordinary human conversation. This significantly improves accessibility, reaching a broad audience beyond just experts.

Pretraining the model serves not only to help it comprehend natural language instructions, but also to tackle problems out-of-the-box, without requiring the user to worry about model training. For instance, the model could be prompted to categorize a message as positive, negative, or neutral, without the need for explicit training examples. This capability, called **zero-shot inference**, essentially relies on the model’s pre-training knowledge alone to fulfill the assigned task.

When zero-shot inference is not enough, such as with smaller language models and/or more intricate tasks, users can provide the model with a few examples illustrating how to tackle a task. This method has multiple names, including **few-shot learning**, **few-shot inference** [12], and **in-context learning**. To instruct the model to classify customer enquiries by service line, for example, one might present sentences such as, “I would like to extend my payment deadline”, “I need to reset my password”, and “I want to make a purchase”, along with their corresponding labels: “billing”, “assistance” and “sales”. Through these examples, the model acquires knowledge, enabling it to generalize and address similar problems. Note that the term “learning” here is somewhat ambiguous since the model’s internal parameters – its weights – remain unchanged, and once the examples are removed from the prompt, the model will forget them.

While zero- and few-shot-inference can be very powerful, examples alone are not guaranteed to get the best out of an LLM. The way the examples and any other instructions and context are provided to the model are also vitally important, and bring us to the art of “**prompt engineering**”.

16.3.3.1.2 Effective Prompt Engineering

The discovery that language models can be instructed using everyday language has led to much excitement, hearsay, and research, on the most effective ways to craft such instructions. Finding “universal rules” to suit all LLMs is a challenge: a highly effective prompt for one LLM might yield subpar results from another. Consequently, a specialized role has emerged: the **prompt engineer**. They have the expertise needed to write effective prompts, honed through experience and a deep understanding of LLM behavior.

If you dream of becoming a prompt engineer, or just want to improve your results with LLMs, we recommend you practice the following techniques, using a variety of models, and stay on the lookout for newly emerging advice in this field.

Ask the question you actually want answered:

It’s easy to just start chatting with an LLM, making requests quite spontaneously, and getting outputs that don’t quite meet your needs. Perhaps you framed the question badly, or perhaps you asked the wrong thing entirely: you may realize you wanted help with a different problem altogether.

The following prompt engineering best practices can help with this, as they each relate to how you frame the problem the LLM should solve.

- First and foremost, always **be clear and concise**. Read your prompt back to yourself before you submit it, ensuring that related ideas are close to one another and there is no

redundant or duplicated information. Rethink and rephrase, if necessary. This will force you to get specific about the problem you need solved. It will also help the LLM pay attention to the core issue, and not get sidetracked by unnecessary details. And note that a concise prompt need not be short. It should be as thorough and detailed as is needed to clearly convey your requirements.

- Another best practice is to **simplify the problem**. Try turning open-ended requests for information into closed classification questions: Instead of *“How should I set up a machine learning monitoring tech stack?”*, ask, *“Which of the following tools are most appropriate for machine learning monitoring, given that I currently use Kubernetes and Google Cloud Platform: Tool X, or Tool Y?”* You can also have the LLM choose from a provided list of possible answers, instead of coming up with its own interpretations: Instead of *“Which topics can you identify in this customer inquiry email?”*, try, *“Which of the following products and services are mentioned in the provided customer inquiry email? Products & services: {}. Customer Email: {}.”* Of course, here you would insert your products or services within the {} placeholder, and the customer email within the {}.
- **Provide constraints**, to ensure an LLM’s output is actually useful to you. This is particularly relevant when using it as an ideation tool or brainstorming partner. Say you want help drafting product descriptions for a new advertising campaign: specify the desired text length, provide certain banned keywords (such as those associated with your competitors), and tell the model to only use product features you’ll provide within the prompt.
- Another tip is to **specify the context and target audience**. One way to do this is implicitly, simply by changing the style of your own prompt, and letting the model adapt its style to match. Thus, a question like *“What are the benefits of taking cold showers?”* could result in an academic, yet impersonal response, whereas asking *“Why and how should I incorporate cold showers into my daily routine?”* will likely generate a much more casual, personal output. Instead of changing your own tone, you can also simply state your desired context or target audience directly. This is particularly convenient for documents with well-defined and well-known formats, as you don’t need to spend a lot of time specifying those formatting rules: Ask an LLM to create an Instagram Post, LinkedIn newsletter or Google Ads text, for example, and it will adapt its language, output length, and use of hashtags and emojis accordingly. Adding the target audience will also help ensure that the resulting text is appropriate and appealing for the desired final reader.
- This leads us to the next best practice, which is to **describe the input format** so that the LLM knows how to handle it, such as being able to differentiate between instructions and additional context information. For example, try something like, *“You will be provided with a company document, denoted in hashtags, and an employee question, denoted by angular brackets. Use information from the document to answer the question. The document: ##. The question: <>.”* Again, here you would insert the document between the ## symbols, and the question between the <>. As this example shows, it can also help to use special characters to structure the prompt more clearly for the model.
- Similarly, **describe the output format** exactly as you need it, keeping in mind the downstream task you need the LLM’s output for. Turn unstructured documents into structured data, for example, by having an LLM extract certain expected entities into a JSON string featuring {Entity A: Value, Entity B: Value}. Such an output can easily be fed to code tools or programs, or saved as a table.

Guide the model through the task:

While the above tips will help you frame your problem and desired output clearly and accurately, it can also be very helpful to instruct the LLM on how it should work through your request. These additional best practices can help you do just that:

- First, **give the model examples** showing how to complete the task (as we discussed in the previous section). This may even be quicker, easier, and just as effective as writing a detailed prompt: provided your examples are good, and the task can easily be inferred from them. For example, if your problem involves taking a lot of similar inputs and systematically driving some result from them, you might not need any instruction at all: You can simply provide a few input-output pairs, and then start providing inputs only, for the LLM to complete.
- Given the nature of LLMs to people-please and hallucinate (more on that, later), it's also important to **give the model an escape route**, should it need one. That is, tell it what to do if it can't solve a problem: it could defer to other tools (assuming it's part of an agent system), return a consistent null value such as "NA" or "not specified", or simply state that it doesn't know.
- Speaking of challenging problems: you can also **tell the model to work step-by-step**. This has been shown to help LLMs execute logical and mathematical problems more effectively (though they're still no master mathematicians, yet!). You can also describe the control flow that the overall interaction should take. For example, when crafting a chatbot assistant application, you could specify the order in which the model should ask for specific information from the user, including constraints so that it only attempts to solve the task once all required pieces of information have been requested and received.
- Another simple yet effective practice is to **ask for multiples and variations**. Inexperienced users may write a prompt like, "*Generate a title for a blog post about subject X,*" get disappointed by the result, and give up. A far better strategy is to ask for N different variations, pick the best one(s), and then iterate, explaining to the model what worked well and what it should try more of.
- Finally, you can **use structured prompts, or templates**. This is an effective way to use LLMs inside larger applications, but even everyday users who predominantly interact with public chatbots can use this technique. Imagine you've just spent a reasonable amount of time conversing with a chatbot, refining and rephrasing your prompt, and adding additional requirements and constraints as you observed the shortcomings of the model's answers. Once you've gotten the output you really needed, ask yourself, will you ever have to solve a similar task again? If so, it can be wise to pack the most effective parts of your conversation into a new prompt, try it out, and if it works, save it somewhere (popular chatbots such as ChatGPT and Google's Gemini include a history function for this purpose). You can also share with colleagues, so that everyone benefits from your fantastic prompt engineering skills!

While these prompt engineering techniques can help you get the best out of a pretrained model, for some tasks, more sophisticated techniques can be used. One option, which we'll discuss next, is to provide the model with additional documents from which it can retrieve information required for a given task. Another is to fine-tune the model with customized data, which we'll cover in Section 16.3.4.2.

16.3.3.2 Retrieval-Augmented Generation

16.3.3.2.1 Introduction to RAG Systems

Relying only on prompting to solve a task can cause the LLM to give incorrect or made-up answers, known as “hallucinations” or “confabulations” [13] (see Section 16.4.3). This can happen because the model does not have enough specific knowledge for those tasks, or because its knowledge has a date cutoff based on whenever its pre-training data was collected. Another concern is that even if an answer is correct, we might not know where it came from because the information it used is hidden within the pre-training data. But sometimes we want or even need to know the source of a model’s responses, either to help us understand and improve it, or to be certain we are building explainable, trustworthy AI.

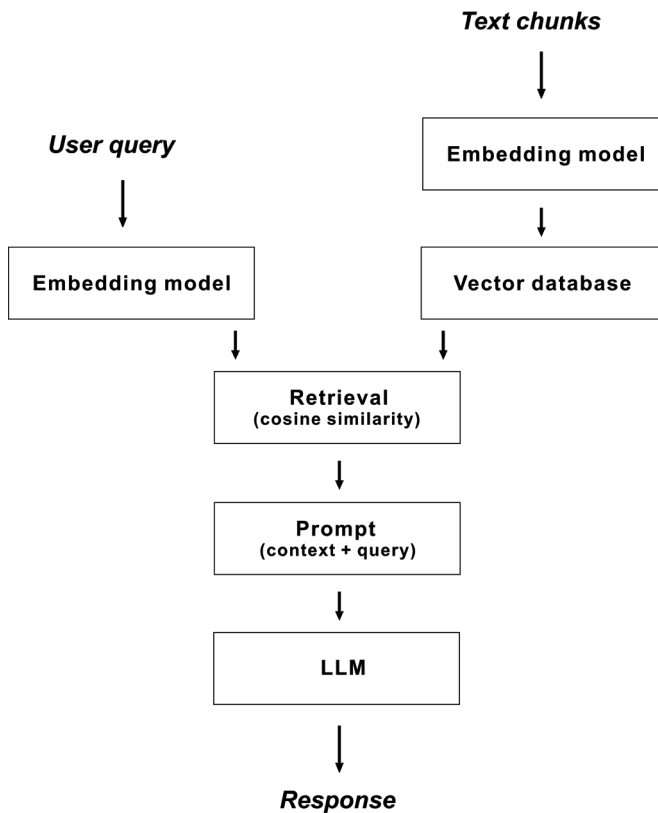


Figure 16.1 Flow of retrieval-augmented generation. User queries and text chunks from relevant documents are embedded using an embedding model. Relevant chunks are retrieved from a vector database based on a similarity measure between the query and chunks. These relevant chunks form the context, which is then added to the prompt, together with the query. The prompt is subsequently fed to the LLM, resulting in the generation of a response.

Theoretically, we could tackle both issues by giving the LLM a lot of information directly in the prompt, such as all of our company's internal documentation. This would provide it with the knowledge it needed, and make its responses more transparent. However, in practice, an LLM's small context window might not fit all the needed documents. Even if it could, we have the token-based processing costs to think about. Finally, this may also lead to **positional bias**: When you load the prompt with lots of information, the LLM tends to focus more on the beginning and end of the window, possibly missing important details in-between [14].

Retrieval-Augmented Generation (RAG) [15] aims to solve these challenges by only adding additional *task-related* information from an external database into the prompt, rather than trying to fit all the needed knowledge into the limited context window (Figure 16.1). A RAG system works like a search engine, finding and retrieving relevant documents to help the LLM with the task at hand. A notable advantage of RAG is that it can easily adapt to changes in the external database. If the database is updated, the LLM can use the new information without needing to retrain its weights. This flexibility allows the LLM to quickly adapt to new data. Also, by focusing on the information in the context window, RAG can reduce problems like hallucinations. Moreover, users can see what information the LLM used to make its decisions, which improves transparency and interpretability.

The RAG process starts when a user asks a question (writes a query) and ends when the LLM gives an answer or solution. This process happens in a few steps: indexing, retrieval, and generation.

Indexing: During indexing, documents are collected, parsed, and stored in a database. However, for a specific question, only a few paragraphs of a specific document might be relevant. Thus, documents are segmented into chunks, usually with some degree of overlap. Next, the text chunks are transformed into embeddings using an embedding model. These are then stored in a vector database, also called index, designed specifically to store and efficiently query embeddings.

Retrieval: The user question is transformed into an embedding with the same model used to embed the text chunks during the indexing stage. The resulting embedding is then compared with all embeddings in the database using similarity measures such as cosine similarity. This method is also known as “semantic search” because it considers the meaning and context of words, unlike a basic search using keywords only. The top-*k* documents with the highest similarity scores, typically about 3 to 5 text chunks, are selected.

Generation: The chosen top-*k* retrieved chunks, which we call the “context”, are integrated into a structured prompt, together with the original query. The LLM uses this context to generate an answer which is based less on its own internal knowledge than would have been, and instead, is more grounded in the information in the retrieved texts.



Katherine Munro is a Data Scientist, Data Science Ambassador and Computational Linguist, conducting research and development and corporate training in AI, Natural Language Processing and Data Science. Katherine began her tech career specializing in user interfaces and Natural Language Understanding, with roles at Mercedes-Benz and the Fraunhofer Institute. She then transitioned to data science in the e-commerce and insurance domains, before landing her current role building smart conversational AI systems using NLP techniques and Large Language Models. In her free time, Katherine is an avid tech writer on X and her own newsletters, and is a volunteer for diverse initiatives helping women and girls start their own tech careers.



Stefan Papp is an entrepreneur who works with Fortune 500 companies to build data platforms and helps them to become more data-driven. Living with his family in Armenia, he is also involved in the Armenian startup ecosystem, and he acts there as an advisor and investor.

Although he has a background in computer science, he strongly believes that the transition to AI and data science is more about culture than about technology. As a libertarian, he sees the benefits and opportunities of data to address current global challenges such as climate change or demographic change.



Zoltan C. Toth is a data engineering architect, lecturer and entrepreneur. With a background in Computer Science and Mathematics, he has taught data architectures, big data technologies and machine learning operations to Fortune 500 companies worldwide. In the past two decades he has worked with several large enterprises as a Solutions Architect, implementing data analytics infrastructures and scaling them up to processing petabytes of data. He is also a lecturer at the Central European University. He founded Datapao, a data engineering consultancy that became Databricks's European professional services center and a Microsoft Gold Partner in Data Science.



Wolfgang Weidinger is a Data Scientist and AI professional. He has worked in a wide variety of industries and sectors such as start-ups, finance, consulting, wholesale and insurance. There he led Data Science & AI teams and drove their role as spearheads in digital and data-driven transformation.

He is President of the Vienna Data Science Group (www.vdsg.at), a non-profit association of and for Data Scientists and all other Data & AI professionals. This brings together both research and practice across a wide range of industries. The VDSG is a rapidly growing international community whose goal is to educate about Data Science and its subfields such as Machine Learning and Artificial Intelligence, as well as their impact on society.

Wolfgang is particularly interested in the societal impact of Data Science and AI, as well as the establishment of interdisciplinary Data Science & AI teams in companies and their disruptive impact on business models. He enjoys lecturing and developing Data Science & AI solutions end2end.



Dr. Danko Nikolić is an expert in both brain research and AI. For many years he has run an electrophysiology lab at the Max-Planck Institute for Brain Research. Also, he is an AI and machine learning professional heading a Data Science team and developing commercial solutions based on AI technology. He invented AI Kindergarten—a concept for training AI of the future for achieving near human-level intelligence. He also pioneered using machine learning to read “minds” from the brain’s electrical signals; he and his team were able to reconstruct what an animal was seeing solely by analyzing the brain signals. He introduced the concept of ideasthesia (“sensing concepts”) into neuroscience and is the author of a theory called practopoiesis describing how biological systems achieve intelligence. He has a degree in Psychology and Civil Engineering from the University of Zagreb, Croatia and a PhD from the University of Oklahoma, USA. He was an honorary professor at the university of Zagreb from 2014 to 2019.



Barbora Antosova Vesela is a data scientist and software engineer working at Frequentis, which operates in a safety critical communication and information environment. Her background is a study of biophysics at Masaryk University in Brno and biomedical engineering both at FH Technikum Wien and Brno University of Technology. She is interested in various topics combining data science and signal and image processing applied in multiple environments, such as medicine, research and air traffic management.



Dr. Karin Bruckmüller studied law and is a criminal lawyer at the Sigmund Freud Private University in Vienna and the Johannes Kepler University in Linz. In both research and teaching, she focuses on medical and nursing criminal law in connection with ethics. She is a regular speaker at relevant nursing congresses and conferences such as the Nursing Congress 2019 in Vienna.



Dr. Annalisa Cadonna is a statistician and data science consultant. She received her Ph.D. in Applied Mathematics and Statistics from University of California, Santa Cruz. Annalisa has applied statistical and machine learning methods to deliver projects in the financial, energy and medical industries. Currently, her professional goal is bridging the gap between time series research and industry applications, by using probabilistic programming and cloud technologies. Annalisa strives to use statistics and machine learning as means for the achievement of the Sustainable Development Goals and to be active in the development of tools and frameworks for responsible artificial intelligence. She is also one of the organizers of R-ladies Vienna.



Dr. Jana Eder is an expert in machine learning and information technology. She earned her Ph.D. in AI and medical imaging through a collaborative effort between Paracelsus Medical University and ETH Zurich, exploring the cutting-edge intersection of AI and healthcare. Following her Ph.D., she completed a habilitation on the use of AI in imaging diagnostics for osteoarthritis. Jana worked as a lecturer at Paracelsus Medical University, Paris Lodron University Salzburg, and the University of Vienna. Currently she holds a position as Senior Key Expert for AI focusing on data efficient learning and multi-source learning at Siemens Technology.



Dr. Jeannette Gorzala, BSc is an attorney at law and AI governance expert working with renowned startups, companies and public organizations to implement and design trustworthy AI systems and business models. As an AI literacy coach, she helps to build up AI know-how and design roles and processes allowing organizations to safely and successfully unlock the potential of AI. She has a background in law and international business administration with more than ten years of experience in investment banking and international law firms with a focus on highly regulated industries and the technology sector.

Jeannette is dedicated to positively shaping the AI ecosystem via her involvement in the independent think tank AI Austria and the European AI Forum, via which she brought together nine international AI associations to give European founders a voice in law and policy making.

As Vice President of the European AI Forum, Jeannette represented more than 2,000 AI entrepreneurs in the legislative process for the Artificial Intelligence Act, advocating for a collaborative and innovation-friendly approach while adequately addressing risks and challenges. Further, Jeannette acts as the Deputy Chair of the AI Advisory Board to the Austrian Government



Dr. Gerald Hahn is proficient in Python, setting up and training deep learning models for NLP and computer vision, pytorch, transformers, classic machine learning algorithms, supervised and non-supervised methods, Bayesian approaches, implementing state of the art literature in machine learning, +10 years of experience in analyzing and modeling neuroscience data.



Dr. Georg Langs is a full professor for Machine Learning in Medical Imaging at the Medical University of Vienna, where he heads the Computational Imaging Research Lab at the Department of Biomedical Imaging and Image-guided Therapy. He is co-founder and chief scientist of the spin-off contextflow GmbH, which develops software for AI-based image search. Georg Langs studied mathematics at the Vienna University of Technology and computer science at the Graz University of Technology, and was a Research Scientist at MIT's Computer Science and Artificial Intelligence Lab, where he is still a Research Affiliate.



Dr. Roxane Licandro is a postdoctoral research associate at the Medical University of Vienna and a research fellow at the Massachusetts General Hospital and Harvard Medical School. She graduated from her medical computer science studies at TU Wien, where she worked as a university assistant at the Computer Vision Lab. She was awarded a Marie Skłodowska-Curie Fellowship and completed research stays at Charité Berlin, Children's Hospital Zurich and University College London. She worked at the Kunsthistorisches Museum Wien and at Agfa Healthcare. Her research focus lies on finding new ways to computationally model and predict dynamic processes in space and over time, paediatric and fetal brain development, statistical pattern analysis in cancer research and geometric shape analysis of anatomical and cultural objects.



Christian Mata is a Business intelligence & Analytics Consultant with more than two decades of BI & reporting project experience for well-known international companies. He leads the development of information solutions and acts as a business analyst bridging business and IT.

He is passionate about leveraging data in organizations and empowering business people to make relevant data visible and accessible. As a trainer, he teaches practical data analysis, visualization and data literacy skills to close the gap between data and decisions.

Christian Mata founded Matadata e.U. in 2014 to strengthen the value-added use of data in organizations. He supports decision-makers at all levels in the cultural shift towards data-based management and in making informed decisions with confidence. Accelerating the transition to a sustainable economy and the implementation of the mobility and energy transition is his current focus.



Sean McIntyre is a Solutions Architect helping teams across Europe to deliver reliable analytics faster using the Modern Data Stack. Over his career, he has worked with software and data in the finance, life sciences, manufacturing, and retail industries. A former International Collegiate Programming Contest world finalist, he has been involved in organizing programming contests and coaching university programming teams for 10 years. He holds a B.Sc. degree in Computer Science and Pure Mathematics from the University of Calgary, and a Masters degree from the Interactive Telecommunications Program at New York University.



Mario Meir-Huber is an experienced senior executive working in different large international organizations. He leads data teams and fosters the cultural change for digitalization and data in these organizations. Before that, he worked in solution architecture roles in leading technology providers such as Microsoft and Teradata. In addition to his job, he is a featured speaker at various international events such as GITEX, WeAreDevelopers or London Tech Week and teaches MBA students on Data Strategy. Mario has already published several books on the topic of the Cloud and (Big) Data.



Gyorgi Mora is a data scientist with strong engineering skills. He designed and built large scale data infrastructures, search systems and machine learning applications.



Manuel Pasioka is an AI Solution Architect and Entrepreneur that helps companies make the right AI strategy and technology decisions, leveraging AI in a way that works for them.

He has a background in distributed and high-performance computing and has been working in several technology startups and research institutes in Austria and Spain.

Since 2021 he has been working as an independent AI consultant and in 2024 started his own AI consultancy and development company.

In addition, he is the host of the “Austrian Artificial Intelligence Podcast” that has the goal to highlight the Austrian AI landscape and its protagonists from academia and private industry.

Manuel is specially interested in the cyber security aspects of machine learning methods and their application in cybersecurity.



Victoria Rugli is a data consultant with a focus on analytics. She received her Master's degree in Data Science and Digital Marketing from Emlyon University in Paris, where she was a student research fellow at the Institute for artificial intelligence in management (AIM) as well as a bachelor's degree in International Business at WU in Vienna. Victoria has delivered projects in the financial, logistics and real estate industries. Currently, her professional focus is data governance and artificial intelligence strategy within cloud technologies. Victoria strives to accelerate the maturity of companies' data by improving the quality and use of data by developing and implementing frameworks for a sustainable data governance implementation. She co-hosts a podcast for artificial intelligence called AI Literacy.



Dr. Rania Wazir is a mathematician and data scientist focussing on Trustworthy AI, Natural Language Processing and Social Media Monitoring. She is a vice chair of Austria's Standards Committee on AI, and Austrian delegate to the ISO working group on Trustworthy AI; she is also coordinator of the VDSC's data4good initiative, which works with non-profits on data-based projects. She lead a consortium of machine learning, legal, and social science experts that recently completed an investigation into bias in algorithms for the EU Fundamental Rights Agency, and is currently tech lead in a three-year project to create a fair by design AI development process, funded by the Austrian Research Agency. Dr. Wazir is co-founder, together with open innovation expert Dr. Gertraud Leimüller, of the recent start-up leiwand.ai, whose goal is to provide companies and organizations involved in the development or use of AI systems with the tools and know-how necessary to ensure their systems are trustworthy.



Günther Zauner is a long-time employee at dwh GmbH, a mathematician and expert in the field of modeling and simulation, parametrization and forecast modeling. He is working on industrial projects as well as on research projects (e.g. EU FP7 CEPHOS-LINK, Horizon 2020 RheumaBuddy). He specializes in the development of modeling concepts, integration of routine data and population behavior. He is a member of VDSC, Society of Medical Decision Making (SMDM) and a member of the board of International Society for Pharmacoeconomics and Outcomes Research Austria (ISPOR Austria). Furthermore, he is reviewer of several journals, and he is doing a PhD study in the field of Public Health under the lead of Professor Majdan at the University of Trnava.

Index

Symbole

1644, Michael Florent 623

A

access control 180
accountability 162
accuracy 177, 285
ACID 141
actions 371
Adam optimizer 404
additive model 429
additive PEFT *see* Parameter-Efficient
Fine-tuning
ad-hoc decision 649
administrative metadata 175
adversarial examples 218
Adversarial Robustness Toolbox 240
adversarial training 365
adversary 364
agent 371
agent-based modelling 599, 604
– actor 599
– agent 599
– COVID-19 604
– emerging behavior 600
– flexibility 600
– natural description 600
agents 474
agent systems *see* agents
AGI 406
agile analytics 662
Agile Manifesto 783
AIAAIC 698
AI Act 700
AI for health care 392
AI Liability Directive 712
AI-powered search 187
Airflow 146
AI stakeholders 796–797, 800, 808–811
AI strategy 28
Alan Turing 393
Albedo 768
AlexNet 359
aliasing 413–415
Aliasing effect 554
alphaGo 382
Amazon Athena 93
Amazon Redshift 42
amortization 19
amount of data 387
analytical competence 649
analytical reporting 317
analytics 390
analytics department 664
antipatterns 786
Antropic 231
Apache Airflow 113
Apache Hadoop 719
Apache Kafka 689
Apache Nifi 123
Apache Spark 103
– Amazon Athena 110
– architecture 104
– Cloud Platforms 110
– DataSet API 105
– driver 105
– executors 105
– RDD API 104
– structured streaming 108
aperture 552

apparent generalization 402
 architecture 379
 – cloud 8
 area chart 631
 Area Under the Curve 287
 Armenia 784
 art 740
 Artificial General Intelligence 406
 Artificial Intelligence (AI) 35, 335, 377, 629
 artificial muse 742
 artificial neural networks (ANN) 355
 association plot 627
 attack
 – causative 216
 – exploratory 216
 – indiscriminate 216
 – targeted 216
 Attack Model 217
 attack surface 217
 attention 453, 530
 attract attention 627
 audio analysis 421
 audio signal 411
 augmented analytics 322
 autocorrelation 431
 autoencoder 363, 447
 automation 158, 651, 820
 automation bias *see* bias
 automotive 721
 autonomous driving 378, 451, 722
 autonomous vehicles 379
 autoregressive mechanism 463
 availability 45, 177
 availability bias *see* bias
 average revenue per user (ARPU) 161
 aviation 725
 AWS 40
 AWS Glue 196
 AWS Lambda 45
 AWS Redshift Spectrum 92
 Azure 40
 Azure Artifacts 135
 Azure Databricks 135
 Azure Data Factory 112–113, 117, 135
 Azure DevOps 134
 Azure Functions 45
 Azure Pipelines 135
 Azure Purview 193
 Azure Repos 132, 135
 Azure SQL Database 135

B

backdoor attacks 218
 bagging 348
 Bag-of-Words (BOW) Input Representation 513
 balanced scorecard 732
 band-pass 422, 429
 bar chart 624, 631
 baseline 641
 Bayer pattern 556
 Bayes' theorem 266
 Bayesian approaches 383
 Bayes rule 519
 Bernoulli distribution 262
 BERT *see* Bidirectional Encoder Representations from Transformers
 bias 345–346, 355, 491, 797–798, 801–803, 808–810–812
 biases 449
 BI department 655
 Bidirectional Encoder Representations from Transformers 539
 BI Engineer 654
 big O notation 417
 Bill Gates 763
 Bi-modal IT 787
 binary classification 223
 Binomial distribution 262
 biological neurons 354
 biometric categorization 704
 black-box access 216
 Bletchley Declaration 699
 blue carbon sequestration 767
 bootstrapping 288
 bottlenecks 448
 boundaries 583
 box plot 426, 634
 brain 380
 Broaden and Build theory 27
 brute-force 380
 brute-force search 405
 bullwhip effect 593, 609
 business analyst 654
 business context 300
 Business Data Owner 654
 Business intelligence 138, 294, 719
 Business Objects 327
 business vault 313

C

calculus 253
 calibration 586
 camera obscura *see* pinhole camera
 cap and trade 762
 capital allocation line 390
 CAP theorem 178
 carbon accounting 763
 carbon capture and storage (CCS) 770
 carbon capture and utilization (CCU) 770
 carbon credits 762
 Carbon Engineering 769
 carbon sequestration 767
 catastrophic forgetting 479
 Central Limit Theorem 264
 central organization 655
 Chain of Thought prompting 474
 change ambassadors 171
 Change Data Capture (CDC) 137, 156
 change management 171, 678
 Charge-Coupled Device (CCD) 554
 Charter of Fundamental Rights 701
 ChatGPT 454, 462, 708
 Chief Data Officer 654
 China 784
 Cholesky decomposition 253
 churn rate 751
 CIA triad 216
 CI/CD 133
 classification model 271, 336
 Climeworks 769
 cloud 662
 cloud-based BI 332
 cloud provider 39
 cluster 351
 Code Injection *see* LLM Attacks
 Cognos Analytics 327
 Color Filter Array (CFA) 554
 column-based security 184
 column chart 631
 commodity price 743
 completeness 178
 compression 557
 computational photography 560
 computerized model 584
 computer science 627
 Computer Vision 547–549, 559–561, 564
 conceptual data model (CDM) 308
 confabulation 487
 configuration drift 57

configuration management 61
 confirmation bias 17 *see* bias
 conflicts of responsibility 650
 confusion matrix 284
 consistency 178
 container 687
 Contextual Word Embeddings 536
 continuous bag-of-words 533
 convolution 416, 441
 Convolutional Neural Networks (CNN) 357, 436, 441, 527
 convolution layers 450
 Conway's law 145
 copyright 488
 Corner detector 560
 correlation 265, 390
 CosmosDB 41
 covariance 265
 COVID-19 604
 – model calibration 608
 – model implementation 609
 – parametrization 608
 – structure and scheduling 606
 C#, programming language 126
 CPU 46
 critical thinking 1
 cross-validation 288
 crypto currencies 390
 CSV 83
 cumulative variables 391
 customer churn 161
 customer journey 731
 customer satisfaction 726
 cybersecurity 215

D

DAG (Directed Acyclic Graph) 112, 146
 Dagster 113–114
 daily standup 676
 Dartmouth conference 381
 dashboard 318
 data
 – availability 580
 data access policies 185
 data analysis 318
 data architecture
 – maintainability 119
 – scalability 89
 database management systems (DBMS) 323

- databases 140
- data bias *see* bias
- Databricks 93, 111, 211
- Databricks Notebooks 152
- data catalog 161
- data classification 181, 688
- data democratization 332
- data-driven decision-making 159
- data engineering
 - ETL 4
 - firewall 5
- data extraction 218
- DataFrame API 104
- data governance 159
- data governance board 162
- DataHub 189
- data ingestion 80
- data integration 305
- data lake 90, 142
- DataLakeHouse 145
- data lineage 160
- data literacy workshop 12
- data management 304
- Data Management Association (DAMA) 162
- Data Management Body of Knowledge (DMBOK) 162
- data maturity 30
- data mesh 145
- data modeling 308
- data monetization 723
- DataOps 678
- data owner 168
- data ownership 165
- data pipeline 135
 - data cleaning 154
- data poisoning 218
- data privacy 182
- data program 717
- data protection 45, 785
- data quality management 172
- data science
 - hypothesis 15
- data science lab 665
- data science life cycle 18
- Data Science Notebooks 132
- data science use cases 24
- data scientist 654
- data source authentication 225
- data sources 80
- data steward 165, 168
- data storytelling 304
- data strategy 651
 - factory of the future 6
- data swamp 664
- data type 637
- data validation 225
- data vault 312
- data visualization 325, 426
- Data Warehouse (DW) 86, 91, 305, 663, 719
- de-biasing 808
- decentralized teams 655
- decision tree 381, 388
- decoder 447
- Deep Blue 393
- Deep Fakes 572
- deep learning 35, 388, 445, 568
- deep neural network 227
- DenseNets 359
- Depth of Field 552
- derivative 254
- descriptive analysis 319
- descriptive metadata 174
- descriptive statistics 390
- design process models 25
- design thinking 26
- determinant 251
- devil's cycle 408
- diagnostic analysis 319
- digital image 556
- digital photography 554
- digital society 778
- digital transformation 717
- dimensionality reduction 438
- dimensional modeling 311
- dimension table 311
- Direct Air Capture (DAC) 769
- directories 43
- discrete event simulation 596
 - dynamic discrete systems 596
 - event list 596
 - priorities 597
- discriminator 364
- disease 605
- diversity 779
- diversity guardrails *see* Guardrails
- Docker 51, 148
- documentation 587
 - source code 587
 - textual 587
- domain adaptation 570

Domain Expert 654
double-counting 774
double diamond 27
dropout 389
durability 45
DynamoDB 42

E

EC2 (Elastic Compute Cloud) 43
edge detection 559
effect ordering for data display 623
efficient frontier 390
eigendecomposition 251
eigenvalues 251
eigenvectors 251
elasticity 45
elections 706
ELMO *see* Embeddings from Language Models
ELT 79, 307
embeddings 478, 533
Embeddings from Language Models 536
embedding space 352
emotion-driven company 28
emotion recognition in the workplace 704
encoder 447
encoder-decoder 461
encoder-decoder architecture 528
encryption 181, 689
energy sector 727
ensemble methods 348
enterprise reporting 325
entropy 426
environment 371
Epic 683
ESG 773, 816
ethics guidelines 796
ETL 78, 138, 307
ETLT 80
EU Artificial Intelligence Act (AIA) 237
EU Cyber Resilience Act (CRA) 237
Europe 785
European Union (EU) 698
evidence-based decision-making 1
evidence-based thinking 28
expectation maximization 351
expected value 264
Experiment Tracking 211
explainability 797, 809, 813
explainable AI 571

exploding-gradient problem 527
exploitation of vulnerabilities 703
exponential growth 395
external validation set 375
eXtreme Programming 783

F

facial images 704
fact table 311
fail-fast approach 662
fairness 796–797, 801–802, 805–808, 812
fairness metrics 802, 805–807, 812
fair use 489
Falcon 230
Fast Fourier Transform 417
Fear of Change 21
feature engineering 201, 389
feature extraction 425, 436, 559
feature selection 438
feature store 207
– Real-time 208
feature vector 559
feed-forward neural network 531
few-shot inference 466
few-shot learning *see* few-shot learning
filter 422–423, 429
financial institutions 730
fine-tuning 231, 477, 540
fixed mindset 692
focal length 552
foundation model 443
Fourier Transform 416
fraud detection 221, 731
frequency domain 416, 441
F-score 285
full fine-tuning 479
function 253
Function as a Service (FaaS) 39

G

Game of Thrones 676
Garry Kasparov 393
Gaussian distribution 263, 404
Gaussian Mixture Model (GMM) 353
GDPR 164
geek culture 22
General Data Protection Regulation 700
Generalised Linear Model (GLM) 391

generality trap 401
 Generative Adversarial Networks (GAN) 365
 Generative AI 1, 322, 460, 505
 generative models 353, 460
 Generative Pre-Trained Transformer 541
 genetic algorithms 388
 geoengineering 771
 geographic profiling 738
 GHG Protocol 765
 git-flow 133
 GitHub 132
 GitLab 132
 global features 559
 global minimum 405
 Global Vectors 534
 GloVe *see* Global Vectors
 GOFAI 381–382, 405
 ‘golden record’ 183
 Google BigQuery 92
 Google Cloud 40
 Google Cloud Functions 45
 Google Data Catalog 194
 GoogLeNet 359
 government 736
 GPAIM 709
 GPT *see* Generative Pre-Trained Transformer
 GPT-2 231
 GPT-3 407
 GPU 46
 gradient 256, 358
 gradient descent 257, 358, 405, 450
 – stochastic 258
 Gradle 134
 gramian angular field 435
 grammar parsers 516
 great engineering 405
 greenwashing 762, 773
 grep 54
 growth mindset 690–692
 guardrails 232, 495
 Guided Transfer Learning 451

H

Habana Labs 46
 hallucination 487
 Hard disks 47
 HDR 561
 heat map 634

Hessian 256
 heterogeneous data 628
 hidden layer 355
 Hidden Markov Model 522
 hidden relationship 622
 High Dynamic Range imaging *see* HDR
 high-pass 422, 429
 Hinton, Geoffrey 763
 HIPAA 164
 histogram 426, 624
 homography 562
 homoscedasticity 404
 HTAP 140
 htop 53
 human intelligence 406
 human in the loop 496
 hybrid cloud 663
 hybrid model-decomposition 586
 hyperparameter tuning 388

I

IAM (AWS Identity and Access Management) 43
 ideation workshops 25
 idempotency 58
 image brightness 559
 image classification model 228
 image formats 556
 image morphing 563
 ImageNet dataset 396
 image processing 412, 421
 image registration *see* registration
 image resolution 553
 image retrieval systems 564–565
 image sharpness 552
 image stitching 563
 image warping 562
 in-context learning 466
 independence of random variables 266
 independence of sampling 404
 individual predictive policing 704
 inductive biases 385
 Informatica 123
 information competing 454
 Infrastructure as a Service (IaaS) 36
 inpainting 561
 input layer 355
 instruction fine-tuning 464, 481
 interest points 559

intrusion detection system 228
iTerm2 51

J

jailbreaking *see* LLM Attacks
Jailbreaking 232
Java 125
Jenkins 133
Jeopardy 379
John McCarthy 381
John Snow 624
JPEG 557
JSON 84

K

Kanban 683, 783
Keiretsu 781
kernel 418, 423
Kestra 114
keypoints *see* interest points
KISS 48
K-Means 520
k-means clustering 351
KNIME Analytics 328
Kotlin, programming language 125–126
Kryoserializer 130
Kubeflow 210
Kubernetes 687

L

lakehouse 91
language model 525
Language Modelling *see* Statistical Language Modelling
large action model 456
Large Language Models 1, 215, 447, 462, 505
lasso 389
laws of physics 388
l-diversity 689
Lego 400
lemmatising 510
LeNet 359
lense equation 552
Library of Alexandria 444
lifecycle 583
linear regression 271, 388
linear relationships 404

linear transformations 250
line chart 630
line graph 624
LISP 381
Llama 230
LLM Attacks 484
LLMs *see* Large Language Models
loan acceptance prediction 732
local features 559
logical data model (LDM) 309
logistic regression 280
long-range correlations 389
long short-term memory (LSTM) 361, 389, 404
Long Short-Term Memory Networks 527
loss 397
low-pass 422–424, 429
Low Rank Adapters 480
Isof 54
LU decomposition 252

M

machine learning 35, 138, 335, 565, 571–573
machine learning perpetuum mobile 403
machine translation 515
macroscopic methods 591
manage data 628
management reporting 315
manifolds 352
manufacturing, mass production 742
MapReduce 49
Marvin Minsky 393
Masked Language Modelling 540
Maslow's hierarchy of needs 72
master data management 172, 182
matrix 244
– diagonal 247
– identity 246
– inverse 246
– positive definite 251
– transpose 246
matrix multiplication 247
matrix-vector multiplication 249
Maven 134
maximum likelihood 353
Mean Absolute Error 276
Mean Squared Error 272
Mersenne Twister 403
metadata 323

Metadata 360 191
 metadata management 172
 microscopic methods 591
 MicroStrategy 328
 Minard 625
 MIPS 46
 MITRE-ATLAS 239
 MLflow 211
 MLOps 199
 ML pipelines 199
 MLSecOps community 240
 MNIST 394
 MNIST dataset 223
 model
 – abstract 580
 – comparison 590
 – concept 580
 – conceptual 584–585
 – epidemic 592
 – falsification 582
 – implementation 588
 – lifecycle 583–584
 – output 584
 – qualitative 586
 – reproducible 581
 – stochasticity 608
 Model Deployment 209
 modelling
 – Black Box 582
 – dynamic 578
 – iteratively 583
 – iterative process 580
 – railway networks 602
 – static 578
 – White Box 582–583
 modelling and simulation 578
 Model Monitoring 204
 Model Registry 212
 Model Serving 213
 model specialization 388
 model stealing 218
 Model Versioning 203
 modular modelling 604
 module 586, 605
 – policies 605
 Moore-Penrose pseudo-inverse 252
 Morris 235
 mosaicing *see* Image stitching
 mosaic plot 627
 MS SQL 41

Multi-Headed Attention *see* Transformer
 Attention
 multilayer perceptrons 355–357
 Multi-modal attacks *see* LLM Attacks
 multi-modal models 461
 multiple linear regression 278
 multiplicative model 430
 multi-purpose backdoor 229

N

Naive Bayes 388, 519
 Naive Bayes Classifier 518
 Named Entity Recognition 512
 Natural Language Generation 504
 Natural Language Processing 503
 Natural Language Processing (NLP) 187, 744
 Natural Language ToolKit 507
 Natural Language Understanding 504
 NDA (Non-Disclosure Agreement) 787
 need to know principle 182
 NER *see* Named Entity Recognition
 netstat 54–56
 network security 180
 net zero 763
 net-zero goal 769
 Neural Networks 527
 neurons 355
 Next Sentence Prediction 540
 Nightingale 626
 NLG *see* Natural Language Generation
 NLP *see* Natural Language Processing
 NLP Pipeline 506
 NLTK *see* Natural Language ToolKit
 NLU *see* Natural Language Understanding
 nmap 54
 no free lunch theorem 403
 noise reduction 441
 Non-Contextual Embeddings 532
 normalization 425, 441
 Noun Chunking 512
 nucleus sampling *see* Sampling
 Nvidia 46
 Nyquist frequency 415

O

Obfuscation *see* LLM Attacks
 object identification 564
 ODBC (Open Database Connectivity) 91

- omnichannel process 731
- one-shot learning 386
- one-shot prompting 453
- online analytical processing (OLAP) 325
- ONNX 202
- OPC UA (OPC Unified Architecture) 81
- OpenAI 231, 397
- OpenMetadata 187
- open source 662
- operational applications 651
- operationalization 668
- operational reporting 316
- OPEX 46
- OPT 230
- optical axis 552
- optical illusions 550
- optimization 253
 - constrained 259
- ordinary differential equations 592
- output layer 355
- overfitting 346, 386
- overplotting 639
- OWASP 237

P

- package manager
 - apk 52
 - apt 52
 - brew 52
 - yum 52
- parameter 584
- Parameter-Efficient Fine-tuning 479
- parameter space 453
- parametrization 586
- Parquet 85
- partial differential equations 578, 592
- Part-of-Speech Tagging 508
- patch management 180
- pattern detection 447
- Payload Splitting *see* LLM Attacks
- Pearson's coefficient of correlation 391
- PEFT *see* Parameter-Efficient Fine-tuning
- perceptron 354
- performance 45
- performance monitoring 316
- personally identifiable information (PII) 688
- perspective projection 552
- Phrase-Based Machine Translation *see* Machine Translation
- physical data model (PDM) 310
- physical security 45, 180
- pie chart 624, 633
- Pinhole 551
- Platform as a Service (PaaS) 36, 39, 61
- PNG 558
- Poisson distribution 262
- polar area chart 626
- policy 371
- polyglot data storage 720
- polyglot persistence 42
- population 605
- POS tagging *see* Part-of-Speech Tagging
- Power BI 327
- power law 397, 456
- pragmatism 383
- precision 285
- prediction 335, 629
- predictive AI 460
- predictive maintenance 729, 747, 753–754
- predictive models 459
- pre-emptive governance 164
- Prefect 113
- Prefix Tuning 480
- prescriptive analytics 717
- pre-trained model 450
- pre-training 464, 532
- Principal Component Analysis (PCA) 251
- privacy by default 688
- privacy by design 688
- Privacy Impact Assessment (PIA) 8, 689
- proactive governance 164
- probability density function 263
- probability mass function 261
- probability theory 260
- processing 627
- Product Liability Directive 712
- Product Owner 683
- prompt encryption 486
- prompt engineering 466
- prompt extraction *see* LLM Attacks
- prompting 451
- prompt injection 233
- Prompt Leaking *see* LLM Attacks
- prompts 465
- prompt sanitization 486
- Prompt Tuning 480
- proof of concept 19, 650
- provisioned IOPS 47
- provisioning tools 58

proximal policy optimization 483
 Proxy Model 217
 PuTTY 51
 PyLint 134
 pythonic 126
 Python, programming language 126

Q

quadratic loss function 272
 qualitative modelling 586
 qualitative variables 270
 quality optimization 744
 quantitative variables 270
 quantization 414, 441, 480

R

RACI matrix 681
 radiometric resolution 553
 RAG *see* Retrieval-Augmented Generation
 RAM 47
 random forest (RF) 346, 388, 732
 random initialization 351
 random number generator 403
 random variable
 – continuous 262
 – discrete 261
 raw vault 313
 ReAct framework 474
 reactive governance 164
 real-time remote biometric identification 704
 recall 285
 Receiver Operating Characteristic (ROC) 286
 recommendation engines 741
 recommender systems 225
 Recurrent Neural Networks 360, 527
 red-teaming 486
 redundancy and backup 180
 registration 569
 regression model 271, 336, 729
 regularization 389
 regulations 798–799
 Reinforcement Learning from
 Human Feedback 465, 482
 reinforcement learning (RL) 371, 602, 609
 ReLu 386, 402
 remote work 790
 reparameterization methods 480

reporting 314
 reproducible
 – documentation 582, 587
 – transparent 581
 – verification and validation 582
 – visualization 582
 reserved instance 46
 ResNets 359
 REST API 81
 retail 748
 Retrieval-Augmented Generation (RAG) 224, 470
 retrospective 20, 684
 return on investment 19
 reward 371
 reward model 482
 ridge 389
 Risk Analyst 655
 RLHF *see* Reinforcement Learning from
 Human Feedback
 role-based access control (RBAC) 180
 Root Mean Squared Error 273
 rose diagram 626
 R, programming language 125
 rule-based decision making 380
 Rule-Based (Symbolic) NLP 515
 Rust, programming language 125–126

S

S3 43
 safety guardrails *see* Guardrails
 Sam Altman 36
 sampling 413, 441, 476
 sandwich defense 486
 SaTML 240
 Scala, programming language 130
 scaled correlation 391
 Scaled Dot-Product Attention *see* Transformer
 Attention
 scale-out 48
 scale-up 48
 scaling intelligence 393
 scaling trap 392
 scatter plot 624, 630
 scenarios 584
 Schema Evolution 91
 schema-on-read 142
 scorecards 318
 Scrum 683, 783

- Scrum Master 676, 683–684
 - SDG goals 762
 - secret managers 689
 - security guardrails *see* Guardrails
 - security policies 180
 - selection bias *see* bias
 - selective PEFT *see* Parameter-Efficient Fine-tuning
 - Self-Attention *see* Transformer Attention
 - self-driving cars 564, 567
 - self-fulfilling bias *see* bias
 - self-service analytics 669
 - self-supervised learning 464
 - self-supervised pre-training 540
 - semantic search 187
 - sensitivity 285
 - sensitivity analysis 590
 - sensor data 722
 - sensor resolution 553
 - sentiment detection 518
 - Sequence-to-Sequence Learning 528
 - serverless 143
 - serverless computing 45
 - Shadow Model 217
 - shared-nothing architecture 42
 - Sharpe ratio 390
 - short-term memory 407
 - Short-Time Fourier Transform 434
 - sieve diagram 627
 - SIFT 560
 - sigmoid 386
 - Silicon Valley 784
 - similarity function 351
 - Single Responsibility Principle 44
 - singular value decomposition 252
 - skewed data 155
 - skip connections 363
 - skip-gram 533
 - SMART 28, 682
 - Smart Cities 738
 - Snowflake 93
 - social credit mechanisms 225
 - social scoring for public and private purposes 703
 - Soft Prompts 479
 - Software as a Service (SaaS) 39
 - SonarCube 134
 - spam detection 518
 - Spark MLlib 109
 - specification 588
 - specificity 285
 - spectral leakage 419–420
 - Spot instance 46
 - Sprint Reviews 684
 - SQL 96
 - standards 798, 801
 - standard deviation 265
 - standardization 425, 441
 - star schema 311
 - state 371
 - Statistical Language Modelling 524
 - Statistical Machine Learning 518
 - Statistical Machine Translation *see* Machine Translation
 - Stemming 509
 - stochastic approaches 522
 - Stoicism 680
 - Stopword Removal 511
 - storage services 41
 - strict model 387
 - STRIDE 237
 - study questions 584
 - subject matter expert (SME) 786
 - subliminal techniques 703
 - supply chain 609
 - support vector machine (SVM) 388
 - surveillance systems 226
 - SWOT 29
 - symbolic
 - description 579
 - symbolic AI 381
 - Symbolic Meaning Representations 517
 - synthetic data 689
 - System Dynamic
 - level 594
 - System Dynamics 593
 - causal loop diagram 594
 - flow 594
 - hypothesized relations 593
 - Top-Down approach 596
- ## T
- Tableau 327
 - tactics 32
 - Tagged Image File Format *see* TIFFF
 - Target Model 217
 - target variable 336
 - t-closeness 689
 - technical metadata 175

telecommunication providers 750
 temperature parameter 476
 Term-Frequency Inverse-Document-Frequency 514
 Terraform 58
 terrestrial sequestration 767
 test data 386
 tests of data validity 590
 text classification 518
 text clustering 520
 TF-IDF *see* Term-Frequency Inverse-Document-Frequency
 Theatrum Orbis Terrarum 622
 The IT crowd 684
 thorough understanding 384
 Three Vs 80
 TIFF 558
 time domain 425
 timeliness 178
 time-series decomposition 428
 Time Travel 85
 tmux 53
 Tokenizing 508
 Token Smuggling *see* LLM Attacks
 Tony Robbins 690
 topical guardrails *see* Guardrails
 top-k sampling *see* Sampling
 top-p sampling *see* Sampling
 TOWS 30
 trace 247
 traffic analysis 737
 training effort 388
 transfer learning 359, 445, 478, 532
 transformation manager 679
 Transformer 463, 537
 Transformer Attention *see* Transformer
 transformers 453
 transition 678
 translation model 525
 transparency 796–797–799, 809–811
 Transport Layer Security (TLS) 689
 tree diagram 635
 triangulation 622
 Trustworthy AI 796–798, 801
 Tufte 627
 two-factor authentication 689

U

under-fitting 346
 understandability 179
 understanding 447
 U-nets 363
 uniform distribution 263
 uni-modal models 461
 use case
 – customer relation 10, 16
 – education 22
 – manufacturing 9
 user-friendly 627

V

validation 588
 – face 590
 – tests of theories 590
 value proposition 19
 values
 – endogenous 579
 – exogenous 579
 vanishing-gradient problem 527
 van Langren 623
 variance 265, 345–346
 vector 244
 vector multiplication 248
 vendor lock-in 663, 719
 vendor-managed inventory 609
 verification 588
 – cross-check 589
 – double implementation 589
 – formal methods 589
 – structural analysis 589
 – structured code walk-trough 589
 – unit testing 589
 Vim 56
 violin plot 427
 virtualization security 180
 virtual network 47
 vision cycle 573
 visual 622
 visual system 549
 visual analysis 320
 visual analytics 321
 visualization
 – data analysis 587
 – modelling structure 587
 Visualization Attack *see* LLM Attacks

voice-controlled systems 226
Von Neumann architecture 48

W

Watson AI 379
wavelet transform 434
Wayne Gretzky 691
weak learners 344, 348
web application firewalls (WAFs)\ 181
weights 449
white-box access 216
window function 419
Word2Vec 533
working memory 407
workshop, “Dr. Evil” 236

X

XaaS 38
XML 83

Y

YAGNI 48

Z

Zero redundancy optimization 481
zero-shot inference 466
zero-shot learning 451
zsh 52