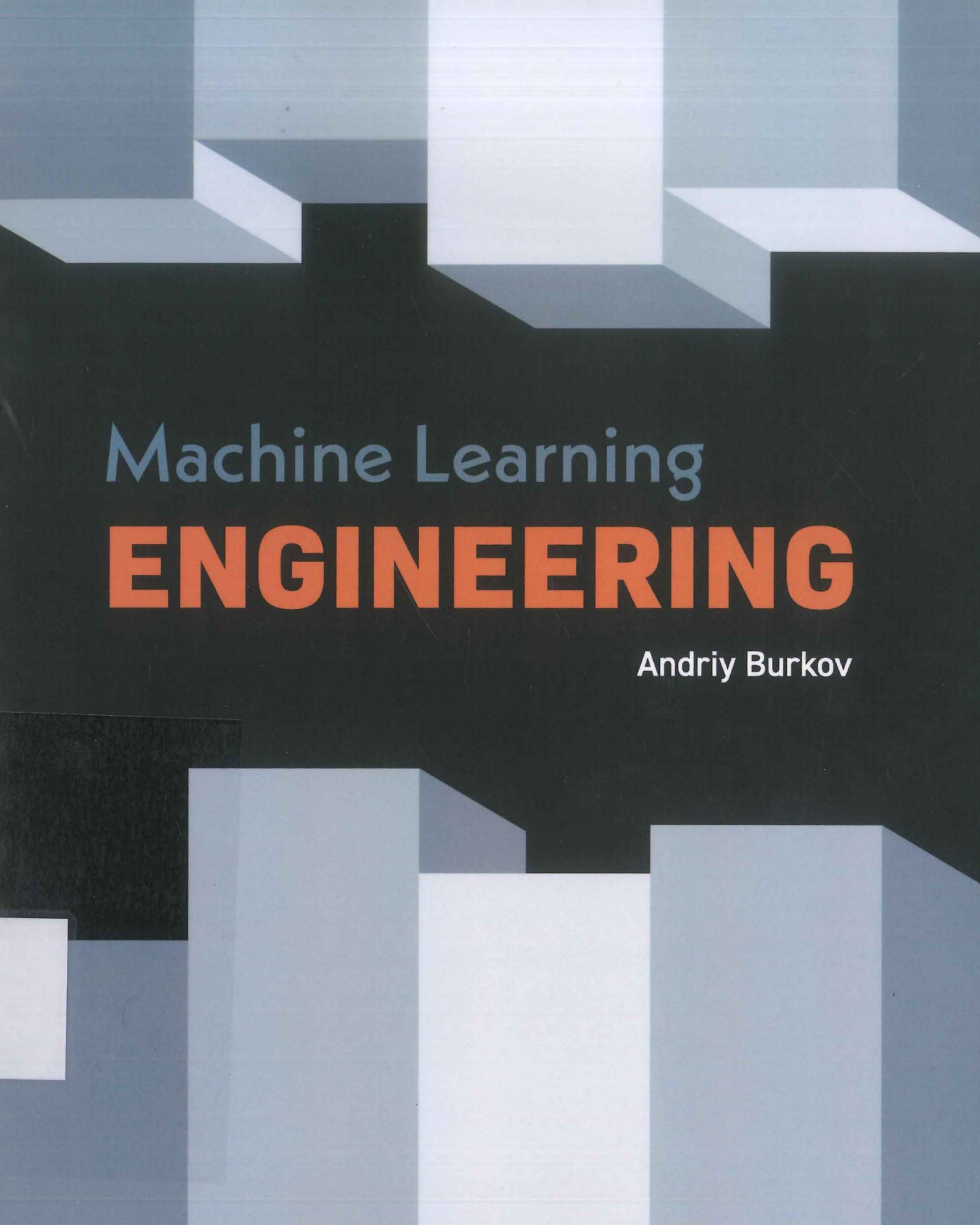




Machine Learning **ENGINEERING**

Andriy Burkov

สำนักหอสมุดมหาวิทยาลัยเชียงใหม่

616477960
012492711
122386452



Machine Learning Engineering

Andriy Burkov

Contents

Foreword	xxiii
Preface	xxv
Who This Book is For	xxv
How to Use This Book	xxvi
Should You Buy This Book?	xxvi
1 Introduction	1
1.1 Notation and Definitions	1
1.1.1 Data Structures	2
1.1.2 Capital Sigma Notation	3
1.2 What is Machine Learning	4
1.2.1 Supervised Learning	4
1.2.2 Unsupervised Learning	5
1.2.3 Semi-Supervised Learning	6
1.2.4 Reinforcement Learning	6
1.3 Data and Machine Learning Terminology	7
1.3.1 Data Used Directly and Indirectly	7
1.3.2 Raw and Tidy Data	7

1.3.3	Training and Holdout Sets	9
1.3.4	Baseline	10
1.3.5	Machine Learning Pipeline	10
1.3.6	Parameters vs. Hyperparameters	10
1.3.7	Classification vs. Regression	11
1.3.8	Model-Based vs. Instance-Based Learning	11
1.3.9	Shallow vs. Deep Learning	12
1.3.10	Training vs. Scoring	12
1.4	When to Use Machine Learning	12
1.4.1	When the Problem Is Too Complex for Coding	12
1.4.2	When the Problem Is Constantly Changing	13
1.4.3	When It Is a Perceptive Problem	13
1.4.4	When It Is an Unstudied Phenomenon	14
1.4.5	When the Problem Has a Simple Objective	14
1.4.6	When It Is Cost-Effective	14
1.5	When Not to Use Machine Learning	15
1.6	What is Machine Learning Engineering	15
1.7	Machine Learning Project Life Cycle	16
1.8	Summary	18
2	Before the Project Starts	21
2.1	Prioritization of Machine Learning Projects	21
2.1.1	Impact of Machine Learning	21
2.1.2	Cost of Machine Learning	22
2.2	Estimating Complexity of a Machine Learning Project	23

2.2.1	The Unknowns	23
2.2.2	Simplifying the Problem	24
2.2.3	Nonlinear Progress	25
2.3	Defining the Goal of a Machine Learning Project	25
2.3.1	What a Model Can Do	25
2.3.2	Properties of a Successful Model	26
2.4	Structuring a Machine Learning Team	27
2.4.1	Two Cultures	27
2.4.2	Members of a Machine Learning Team	27
2.5	Why Machine Learning Projects Fail	29
2.5.1	Lack of Experienced Talent	29
2.5.2	Lack of Support by the Leadership	29
2.5.3	Missing Data Infrastructure	30
2.5.4	Data Labeling Challenge	30
2.5.5	Siloed Organizations and Lack of Collaboration	30
2.5.6	Technically Infeasible Projects	31
2.5.7	Lack of Alignment Between Technical and Business Teams	31
2.6	Summary	32
3	Data Collection and Preparation	35
3.1	Questions About the Data	36
3.1.1	Is the Data Accessible?	36
3.1.2	Is the Data Sizeable?	36
3.1.3	Is the Data Useable?	38
3.1.4	Is the Data Understandable?	40

3.1.5	Is the Data Reliable?	40
3.2	Common Problems With Data	41
3.2.1	High Cost	41
3.2.2	Bad Quality	44
3.2.3	Noise	44
3.2.4	Bias	44
3.2.5	Low Predictive Power	50
3.2.6	Outdated Examples	50
3.2.7	Outliers	51
3.2.8	Data Leakage	52
3.3	What Is Good Data	52
3.3.1	Good Data Is Informative	53
3.3.2	Good Data Has Good Coverage	53
3.3.3	Good Data Reflects Real Inputs	53
3.3.4	Good Data Is Unbiased	54
3.3.5	Good Data Is Not a Result of a Feedback Loop	54
3.3.6	Good Data Has Consistent Labels	54
3.3.7	Good Data Is Big Enough	54
3.3.8	Summary of Good Data	55
3.4	Dealing With Interaction Data	55
3.5	Causes of Data Leakage	56
3.5.1	Target is a Function of a Feature	56
3.5.2	Feature Hides the Target	57
3.5.3	Feature From the Future	57

3.6	Data Partitioning	58
3.6.1	Leakage During Partitioning	60
3.7	Dealing with Missing Attributes	60
3.7.1	Data Imputation Techniques	61
3.7.2	Leakage During Imputation	62
3.8	Data Augmentation	62
3.8.1	Data Augmentation for Images	63
3.8.2	Data Augmentation for Text	64
3.9	Dealing With Imbalanced Data	66
3.9.1	Oversampling	66
3.9.2	Undersampling	67
3.9.3	Hybrid Strategies	67
3.10	Data Sampling Strategies	68
3.10.1	Simple Random Sampling	69
3.10.2	Systematic Sampling	70
3.10.3	Stratified Sampling	70
3.11	Storing Data	70
3.11.1	Data Formats	71
3.11.2	Data Storage Levels	72
3.11.3	Data Versioning	74
3.11.4	Documentation and Metadata	75
3.11.5	Data Lifecycle	76
3.12	Data Manipulation Best Practices	76
3.12.1	Reproducibility	76

3.12.2	Data First, Algorithm Second	77
3.13	Summary	77
4	Feature Engineering	79
4.1	Why Engineer Features	80
4.2	How to Engineer Features	81
4.2.1	Feature Engineering for Text	81
4.2.2	Why Bag-of-Words Works	84
4.2.3	Converting Categorical Features to Numbers	84
4.2.4	Feature Hashing	86
4.2.5	Topic Modeling	88
4.2.6	Features for Time-Series	91
4.2.7	Use Your Creativity	94
4.3	Stacking Features	95
4.3.1	Stacking Feature Vectors	95
4.3.2	Stacking Individual Features	95
4.4	Properties of Good Features	97
4.4.1	High Predictive Power	97
4.4.2	Fast Computability	97
4.4.3	Reliability	98
4.4.4	Uncorrelatedness	98
4.4.5	Other Properties	98
4.5	Feature Selection	99
4.5.1	Cutting the Long Tail	99
4.5.2	Boruta	100

4.5.3	L1-Regularization	102
4.5.4	Task-Specific Feature Selection	103
4.6	Synthesizing Features	103
4.6.1	Feature Discretization	103
4.6.2	Synthesizing Features from Relational Data	106
4.6.3	Synthesizing Features from the Data	107
4.6.4	Synthesizing Features from Other Features	107
4.7	Learning Features from Data	108
4.7.1	Word Embeddings	108
4.7.2	Document Embeddings	109
4.7.3	Embeddings of Anything	111
4.7.4	Choosing Embedding Dimensionality	111
4.8	Dimensionality Reduction	112
4.8.1	Fast Dimensionality Reduction with PCA	113
4.8.2	Dimensionality Reduction for Visualization	113
4.9	Scaling Features	114
4.9.1	Normalization	114
4.9.2	Standardization	115
4.10	Data Leakage in Feature Engineering	116
4.10.1	Possible Problems	116
4.10.2	Solution	116
4.11	Storing and Documenting Features	116
4.11.1	Schema File	117
4.11.2	Feature Store	118

4.12	Feature Engineering Best Practices	120
4.12.1	Generate Many Simple Features	120
4.12.2	Reuse Legacy Systems	121
4.12.3	Use IDs as Features when Needed.	121
4.12.4	... But Reduce the Cardinality When Possible	121
4.12.5	Use Counts with Caution	122
4.12.6	Make Feature Selection When Necessary	122
4.12.7	Test the Code Carefully	123
4.12.8	Keep Code, Model, and Data in Sync	123
4.12.9	Isolate Feature Extraction Code	123
4.12.10	Serialize Together Model and Feature Extractor	123
4.12.11	Log the Values of Features	124
4.13	Summary	124
5	Supervised Model Training (Part 1)	127
5.1	Before You Start Working on the Model	128
5.1.1	Validate Schema Conformity	128
5.1.2	Define an Achievable Performance Level	128
5.1.3	Choose a Performance Metric	129
5.1.4	Choose the Right Baseline	129
5.1.5	Split Data Into Three Sets	131
5.1.6	Preconditions for Supervised Learning	132
5.2	Representing Labels for Machine Learning	132
5.2.1	Multiclass Classification	133
5.2.2	Multi-label Classification	133

5.3	Selecting the Learning Algorithm	134
5.3.1	Main Properties of a Learning Algorithm	134
5.3.2	Algorithm Spot-Checking	137
5.4	Building a Pipeline	137
5.5	Assessing Model Performance	138
5.5.1	Performance Metrics for Regression	139
5.5.2	Performance Metrics for Classification	140
5.5.3	Performance Metrics for Ranking	145
5.6	Hyperparameter Tuning	148
5.6.1	Grid Search	149
5.6.2	Random Search	150
5.6.3	Coarse-to-Fine Search	152
5.6.4	Other Techniques	152
5.6.5	Cross-Validation	152
5.7	Shallow Model Training	153
5.7.1	Shallow Model Training Strategy	153
5.7.2	Saving and Restoring the Model	154
5.8	Bias-Variance Tradeoff	155
5.8.1	Underfitting	155
5.8.2	Overfitting	156
5.8.3	The Tradeoff	157
5.9	Regularization	158
5.9.1	L1 and L2 Regularization	159
5.9.2	Other Forms of Regularization	160

5.10 Summary	160
6 Supervised Model Training (Part 2)	163
6.1 Deep Model Training Strategy	163
6.1.1 Neural Network Training Strategy	164
6.1.2 Performance Metric and Cost Function	164
6.1.3 Parameter-Initialization Strategies	167
6.1.4 Optimization Algorithms	168
6.1.5 Learning Rate Decay Schedules	171
6.1.6 Regularization	173
6.1.7 Network Size Search and Hyperparameter Tuning	173
6.1.8 Handling Multiple Inputs	175
6.1.9 Handling Multiple Outputs	176
6.1.10 Transfer Learning	177
6.2 Stacking Models	179
6.2.1 Types of Ensemble Learning	179
6.2.2 An Algorithm of Model Stacking	180
6.2.3 Data Leakage in Model Stacking	180
6.3 Dealing With Distribution Shift	181
6.3.1 Types of Distribution Shift	182
6.3.2 Adversarial Validation	182
6.4 Handling Imbalanced Datasets	183
6.4.1 Class Weighting	183
6.4.2 Ensemble of Resampled Datasets	183
6.4.3 Other Techniques	185

6.5	Model Calibration	185
6.5.1	Well-Calibrated Models	185
6.5.2	Calibration Techniques	187
6.6	Troubleshooting and Error Analysis	187
6.6.1	Reasons for Poor Model Behavior	188
6.6.2	Iterative Model Refinement	188
6.6.3	Error Analysis	189
6.6.4	Error Analysis in Complex Systems	190
6.6.5	Using Sliced Metrics	191
6.6.6	Fixing Wrong Labels	192
6.6.7	Finding Additional Examples to Label	192
6.6.8	Troubleshooting Deep Learning	193
6.7	Best Practices	194
6.7.1	Deliver a Good Model	195
6.7.2	Trust Popular Open Source Implementations	195
6.7.3	Optimize a Business-Specific Performance Measure	195
6.7.4	Upgrade From Scratch	195
6.7.5	Avoid Correction Cascades	196
6.7.6	Use Model Cascading With Caution	196
6.7.7	Write Efficient Code, Compile, and Parallelize	197
6.7.8	Test on Both Newer and Older Data	198
6.7.9	More Data Beats Cleverer Algorithm	198
6.7.10	New Data Beats Cleverer Features	199
6.7.11	Embrace Tiny Progress	199

6.7.12 Facilitate Reproducibility	199
6.8 Summary	200
7 Model Evaluation	203
7.1 Offline and Online Evaluation	204
7.2 A/B Testing	206
7.2.1 G-Test	207
7.2.2 Z-Test	210
7.2.3 Concluding Remarks and Warnings	212
7.3 Multi-Armed Bandit	212
7.4 Statistical Bounds on the Model Performance	215
7.4.1 Statistical Interval for the Classification Error	216
7.4.2 Bootstrapping Statistical Interval	217
7.4.3 Bootstrapping Prediction Interval for Regression	218
7.5 Evaluation of Test Set Adequacy	218
7.5.1 Neuron Coverage	219
7.5.2 Mutation Testing	219
7.6 Evaluation of Model Properties	220
7.6.1 Robustness	220
7.6.2 Fairness	221
7.7 Summary	222
8 Model Deployment	223
8.1 Static Deployment	224
8.2 Dynamic Deployment on User's Device	224

8.2.1	Deployment of Model Parameters	225
8.2.2	Deployment of a Serialized Object	225
8.2.3	Deploying to Browser	225
8.2.4	Advantages and Drawbacks	225
8.3	Dynamic Deployment on a Server	226
8.3.1	Deployment on a Virtual Machine	226
8.3.2	Deployment in a Container	227
8.3.3	Serverless Deployment	229
8.3.4	Model Streaming	230
8.4	Deployment Strategies	232
8.4.1	Single Deployment	232
8.4.2	Silent Deployment	233
8.4.3	Canary Deployment	233
8.4.4	Multi-Armed Bandits	234
8.5	Automated Deployment, Versioning, and Metadata	234
8.5.1	Model Accompanying Assets	234
8.5.2	Version Sync	235
8.5.3	Model Version Metadata	235
8.6	Model Deployment Best Practices	236
8.6.1	Algorithmic Efficiency	236
8.6.2	Deployment of Deep Models	239
8.6.3	Caching	239
8.6.4	Delivery Format for Model and Code	240
8.6.5	Start With a Simple Model	243

8.6.6	Test on Outsiders	243
8.7	Summary	243
9	Model Serving, Monitoring, and Maintenance	245
9.1	Properties of the Model Serving Runtime	246
9.1.1	Security and Correctness	246
9.1.2	Ease of Deployment	246
9.1.3	Guarantees of Model Validity	247
9.1.4	Ease of Recovery	247
9.1.5	Avoidance of Training/Serving Skew	248
9.1.6	Avoidance of Hidden Feedback Loops	248
9.2	Modes of Model Serving	249
9.2.1	Serving in Batch Mode	249
9.2.2	Serving on Demand to a Human	249
9.2.3	Serving on Demand to a Machine	251
9.3	Model Serving in Real World	252
9.3.1	Being Ready for Errors	252
9.3.2	Dealing With Errors	253
9.3.3	Being Ready for, and Dealing With, Change	254
9.3.4	Being Ready for, and Dealing With, Human Nature	256
9.4	Model Monitoring	257
9.4.1	What Can Go Wrong?	257
9.4.2	What and How to Monitor	258
9.4.3	What to Log	260
9.4.4	Monitor for Abuse	261

9.5	Model Maintenance . . .	262
9.5.1	When to Update . . .	262
9.5.2	How to Update . . .	263
9.6	Summary . . .	266
10	Conclusion	269
10.1	Takeaways . . .	269
10.2	What to Read Next . . .	273
10.3	Acknowledgements . . .	274
	Index	275