

Machine Learning Theory and Applications

**Hands-on Use Cases with Python
on Classical and Quantum Machines**

Xavier Vasques

WILEY

Contents

Foreword *xiii*

Acknowledgments *xv*

General Introduction *xvii*

1	Concepts, Libraries, and Essential Tools in Machine Learning and Deep Learning	1
1.1	Learning Styles for Machine Learning	2
1.1.1	Supervised Learning	2
1.1.1.1	Overfitting and Underfitting	3
1.1.1.2	K-Folds Cross-Validation	4
1.1.1.3	Train/Test Split	4
1.1.1.4	Confusion Matrix	5
1.1.1.5	Loss Functions	7
1.1.2	Unsupervised Learning	9
1.1.3	Semi-Supervised Learning	9
1.1.4	Reinforcement Learning	9
1.2	Essential Python Tools for Machine Learning	9
1.2.1	Data Manipulation with Python	10
1.2.2	Python Machine Learning Libraries	10
1.2.2.1	Scikit-learn	10
1.2.2.2	TensorFlow	10
1.2.2.3	Keras	12
1.2.2.4	PyTorch	12
1.2.3	Jupyter Notebook and JupyterLab	13
1.3	HephAIstos for Running Machine Learning on CPUs, GPUs, and QPUs	13
1.3.1	Installation	13
1.3.2	HephAIstos Function	15
1.4	Where to Find the Datasets and Code Examples	32
	Further Reading	33
2	Feature Engineering Techniques in Machine Learning	35
2.1	Feature Rescaling: Structured Continuous Numeric Data	36
2.1.1	Data Transformation	37
2.1.1.1	StandardScaler	37
2.1.1.2	MinMaxScaler	39
2.1.1.3	MaxAbsScaler	40
2.1.1.4	RobustScaler	40
2.1.1.5	Normalizer: Unit Vector Normalization	42
2.1.1.6	Other Options	43
2.1.1.7	Transformation to Improve Normal Distribution	44

- 2.1.1.8 Quantile Transformation 48
- 2.1.2 Example: Rescaling Applied to an SVM Model 50
- 2.2 Strategies to Work with Categorical (Discrete) Data 57
 - 2.2.1 Ordinal Encoding 59
 - 2.2.2 One-Hot Encoding 61
 - 2.2.3 Label Encoding 62
 - 2.2.4 Helmert Encoding 63
 - 2.2.5 Binary Encoding 64
 - 2.2.6 Frequency Encoding 65
 - 2.2.7 Mean Encoding 66
 - 2.2.8 Sum Encoding 68
 - 2.2.9 Weight of Evidence Encoding 68
 - 2.2.10 Probability Ratio Encoding 70
 - 2.2.11 Hashing Encoding 71
 - 2.2.12 Backward Difference Encoding 72
 - 2.2.13 Leave-One-Out Encoding 73
 - 2.2.14 James-Stein Encoding 74
 - 2.2.15 M-Estimator Encoding 76
 - 2.2.16 Using HephAistos to Encode Categorical Data 77
- 2.3 Time-Related Features Engineering 77
 - 2.3.1 Date-Related Features 79
 - 2.3.2 Lag Variables 79
 - 2.3.3 Rolling Window Feature 82
 - 2.3.4 Expanding Window Feature 84
 - 2.3.5 Understanding Time Series Data in Context 85
- 2.4 Handling Missing Values in Machine Learning 88
 - 2.4.1 Row or Column Removal 89
 - 2.4.2 Statistical Imputation: Mean, Median, and Mode 90
 - 2.4.3 Linear Interpolation 91
 - 2.4.4 Multivariate Imputation by Chained Equation Imputation 92
 - 2.4.5 KNN Imputation 93
- 2.5 Feature Extraction and Selection 97
 - 2.5.1 Feature Extraction 97
 - 2.5.1.1 Principal Component Analysis 98
 - 2.5.1.2 Independent Component Analysis 102
 - 2.5.1.3 Linear Discriminant Analysis 110
 - 2.5.1.4 Locally Linear Embedding 115
 - 2.5.1.5 The *t*-Distributed Stochastic Neighbor Embedding Technique 123
 - 2.5.1.6 More Manifold Learning Techniques 125
 - 2.5.1.7 Feature Extraction with HephAistos 130
 - 2.5.2 Feature Selection 131
 - 2.5.2.1 Filter Methods 132
 - 2.5.2.2 Wrapper Methods 146
 - 2.5.2.3 Embedded Methods 154
 - 2.5.2.4 Feature Importance Using Graphics Processing Units (GPUs) 167
 - 2.5.2.5 Feature Selection Using HephAistos 168
 - Further Reading 170
- 3 Machine Learning Algorithms 175
 - 3.1 Linear Regression 176
 - 3.1.1 The Math 176
 - 3.1.2 Gradient Descent to Optimize the Cost Function 177

- 3.1.3 Implementation of Linear Regression 182
 - 3.1.3.1 Univariate Linear Regression 182
 - 3.1.3.2 Multiple Linear Regression: Predicting Water Temperature 185
- 3.2 Logistic Regression 202
 - 3.2.1 Binary Logistic Regression 202
 - 3.2.1.1 Cost Function 203
 - 3.2.1.2 Gradient Descent 204
 - 3.2.2 Multinomial Logistic Regression 204
 - 3.2.3 Multinomial Logistic Regression Applied to Fashion MNIST 204
 - 3.2.3.1 Logistic Regression with scikit-learn 205
 - 3.2.3.2 Logistic Regression with Keras on TensorFlow 208
 - 3.2.4 Binary Logistic Regression with Keras on TensorFlow 210
- 3.3 Support Vector Machine 211
 - 3.3.1 Linearly Separable Data 212
 - 3.3.2 Not Fully Linearly Separable Data 214
 - 3.3.3 Nonlinear SVMs 216
 - 3.3.4 SVMs for Regression 217
 - 3.3.5 Application of SVMs 219
 - 3.3.5.1 SVM Using scikit-learn for Classification 220
 - 3.3.5.2 SVM Using scikit-learn for Regression 222
- 3.4 Artificial Neural Networks 223
 - 3.4.1 Multilayer Perceptron 224
 - 3.4.2 Estimation of the Parameters 225
 - 3.4.2.1 Loss Functions 225
 - 3.4.2.2 Backpropagation: Binary Classification 226
 - 3.4.2.3 Backpropagation: Multi-class Classification 227
 - 3.4.3 Convolutional Neural Networks 230
 - 3.4.4 Recurrent Neural Network 232
 - 3.4.5 Application of MLP Neural Networks 233
 - 3.4.6 Application of RNNs: LST Memory 242
 - 3.4.7 Building a CNN 246
- 3.5 Many More Algorithms to Explore 249
- 3.6 Unsupervised Machine Learning Algorithms 251
 - 3.6.1 Clustering 251
 - 3.6.1.1 K-means 253
 - 3.6.1.2 Mini-batch K-means 255
 - 3.6.1.3 Mean Shift 257
 - 3.6.1.4 Affinity Propagation 259
 - 3.6.1.5 Density-based Spatial Clustering of Applications with Noise 262
- 3.7 Machine Learning Algorithms with HephAistos 264
 - References 270
 - Further Reading 270
- 4 Natural Language Processing 273
 - 4.1 Classifying Messages as Spam or Ham 274
 - 4.2 Sentiment Analysis 281
 - 4.3 Bidirectional Encoder Representations from Transformers 286
 - 4.4 BERT's Functionality 287
 - 4.5 Installing and Training BERT for Binary Text Classification Using TensorFlow 288
 - 4.6 Utilizing BERT for Text Summarization 294
 - 4.7 Utilizing BERT for Question Answering 296
 - Further Reading 297

5	Machine Learning Algorithms in Quantum Computing	299
5.1	Quantum Machine Learning	303
5.2	Quantum Kernel Machine Learning	306
5.3	Quantum Kernel Training	328
5.4	Pegasos QSVC: Binary Classification	333
5.5	Quantum Neural Networks	337
5.5.1	Binary Classification with EstimatorQNN	338
5.5.2	Classification with a SamplerQNN	343
5.5.3	Classification with Variational Quantum Classifier	348
5.5.4	Regression	351
5.6	Quantum Generative Adversarial Network	352
5.7	Quantum Algorithms with HephAistos	368
	References	372
	Further Reading	373
6	Machine Learning in Production	375
6.1	Why Use Docker Containers for Machine Learning?	375
6.1.1	First Things First: The Microservices	375
6.1.2	Containerization	376
6.1.3	Docker and Machine Learning: Resolving the “It Works in My Machine” Problem	376
6.1.4	Quick Install and First Use of Docker	377
6.1.4.1	Install Docker	377
6.1.4.2	Using Docker from the Command Line	378
6.1.5	Dockerfile	380
6.1.6	Build and Run a Docker Container for Your Machine Learning Model	381
6.2	Machine Learning Prediction in Real Time Using Docker and Python REST APIs with Flask	389
6.2.1	Flask-RESTful APIs	390
6.2.2	Machine Learning Models	392
6.2.3	Docker Image for the Online Inference	393
6.2.4	Running Docker Online Inference	394
6.3	From DevOps to MLOPS: Integrate Machine Learning Models Using Jenkins and Docker	396
6.3.1	Jenkins Installation	397
6.3.2	Scenario Implementation	399
6.4	Machine Learning with Docker and Kubernetes: Install a Cluster from Scratch	405
6.4.1	Kubernetes Vocabulary	405
6.4.2	Kubernetes Quick Install	406
6.4.3	Install a Kubernetes Cluster	407
6.4.4	Kubernetes: Initialization and Internal Network	410
6.5	Machine Learning with Docker and Kubernetes: Training Models	415
6.5.1	Kubernetes Jobs: Model Training and Batch Inference	415
6.5.2	Create and Prepare the Virtual Machines	415
6.5.3	Kubeadm Installation	415
6.5.4	Create a Kubernetes Cluster	416
6.5.5	Containerize our Python Application that Trains Models	418
6.5.6	Create Configuration Files for Kubernetes	422
6.5.7	Commands to Delete the Cluster	424
6.6	Machine Learning with Docker and Kubernetes: Batch Inference	424
6.6.1	Create Configuration Files for Kubernetes	427
6.7	Machine Learning Prediction in Real Time Using Docker, Python Rest APIs with Flask, and Kubernetes: Online Inference	428
6.7.1	Flask-RESTful APIs	428
6.7.2	Machine Learning Models	431

6.7.3	Docker Image for Online Inference	432
6.7.4	Running Docker Online Inference	433
6.7.5	Create and Prepare the Virtual Machines	434
6.7.6	Kubeadm Installation	434
6.7.7	Create a Kubernetes Cluster	435
6.7.8	Deploying the Containerized Machine Learning Model to Kubernetes	437
6.8	A Machine Learning Application that Deploys to the IBM Cloud Kubernetes Service: Python, Docker, Kubernetes	440
6.8.1	Create Kubernetes Service on IBM Cloud	440
6.8.2	Containerization of a Machine Learning Application	443
6.8.3	Push the Image to the IBM Cloud Registry	446
6.8.4	Deploy the Application to Kubernetes	448
6.9	Red Hat OpenShift to Develop and Deploy Enterprise ML/DL Applications	452
6.9.1	What is OpenShift?	453
6.9.2	What Is the Difference Between OpenShift and Kubernetes?	453
6.9.3	Why Red Hat OpenShift for ML/DL? To Build a Production-Ready ML/DL Environment	454
6.10	Deploying a Machine Learning Model as an API on the Red Hat OpenShift Container Platform: From Source Code in a GitHub Repository with Flask, Scikit-Learn, and Docker	454
6.10.1	Create an OpenShift Cluster Instance	455
6.10.1.1	Deploying an Application from Source Code in a GitHub Repository	457
	Further Reading	463
	Conclusion: The Future of Computing for Data Science?	465
	Index	477