Index

Note: Page numbers referring to figures should be *italics* and those referring to tables should be **bold**

a

absolute error loss 8
accelerators 452
accuracy score 234
ACF (autocorrelation function) 80, 81
activation function 223, 224, *224,* 227
aeronautics 474
affinity propagation 252, 259–262
agglomerative algorithms 252
AI (artificial intelligence) 1, 273, 375, 465
alternative hypothesis 132
analysis of variance (ANOVA) F statistic 137–140, 168
ANN. *see* artificial neural networks
ANOVA (analysis of variance) F statistic 137–140, 168
api.py file 428, 443, 445, 458
apiVersion 423
artificial intelligence (AI) 1, 375, 465
artificial neural networks (ANN) 175, 223–249
  backpropagation 227–230
  convolutional neural networks 230–232
  estimation of parameters 225–230
  multilayer perceptron 224–225
  multilayer perceptron neural networks 233–242
  recurrent neural networks 232–233, *233*
  schematic representation of *223*
autocorrelation function (ACF) 80, 81

b

backpropagation
  binary classification 226–227
  multi-class classification 227–230
backward difference encoding method 72–73
backward elimination 151–153
backward propagation of errors 226
bag-of-words method 274, 278, 280, 296
bash command 378
batch gradient descent 181
batch inference 389, 415, 424–428
Bayesian approach 76
Bayesian encoders 58
Bayesian machine learning models 175
BCSS (between-cluster sum of squares) 253
Bernoulli random variables 132
BERT. *see* Bidirectional Encoder Representations from Transformers
between-cluster sum of squares (BCSS) 253
bias 4
Bidirectional Encoder Representations from Transformers (BERT) 11, 251, 286–287
  bidirectional training 287
  for binary text classification using tensorflow 288–294
  functionality 287–288
  language models 287
  objective 287
  for question answering 296–297
  for text summarization 294–295
bidirectional training approach 287
binary classification 2, 225, 333–337
  with EstimatorQNN 338–343
  regression 351
  with SamplerQNN 343–348
  with variational quantum classifier 348–351
binary cross-entropy 225
  as loss functions 8, 352
binary encoding method, *64,* 64–65
binary logistic regression 202–204
  cost function 203–204
  gradient descent 204
  with Keras on TensorFlow 210–211
binary systems 466–467
biologically inspired systems 468–469

Machine Learning Theory and Applications: Hands-on Use Cases with Python on Classical and Quantum Machines, First Edition. Xavier Vasques.

bottom-up clustering 252
Box-Cox transformation 46, 47

c

Caffe2 12
California Cooperative Oceanic Fisheries Investigations (CalCOFI) dataset 185
Captum 12
categorical cross-entropy loss 8
categorical (discrete) data 17–18, 57–77
  encoding methods 57–58, *58*. *see also* encoding methods
  nominal 57
  types of *57*
  using hephAIstos to encode 77
  workflow with continuous and *58*
CD (continuous delivery) 389, 396, 397
CentOS 378, 397
CERN 304
ChatGPT 286
Chebyshev norm 43
chi-squared test 134–137, 168
  for feature selection 135
  percentiles of *136*
CI (continuous integration) 389, 396, 397
class-dependent LDA 110
classical computing 465
classical information theory 300
classical kernel method 307
classical neural network 352
classical physics 299, 470
classical support vector machines 303
classification accuracy 6, *327*
classification algorithms 23–31, 274
  on CPUs 14
  on GPUs 14, 24–25, 266
  support vector machine using 220–222
class-independent LDA 110
clustering 252–264
  Kubernetes 407–409, 416–418, 426, 435–437
  objective of 175
  techniques 252
clustering algorithm 252–264
  affinity propagation 259–262
  DBSCAN 252, 262–264
  k-means 252–255
  mean shift clustering 252, 257–259
  mini-batch k-means 252, 255–256
CNNs (convolutional neural networks) 12, 32, 223, 230–232, 246–249, 269, 274
CNOT (Controlled-NOT) gate 302
"cocktail party problem" 102
conditional maximum likelihood estimation 203
configuration files, for Kubernetes 422–423, 427–428
confusion matrix *5,* 5–7
containerization 376
  of machine learning application 443–446
continuous delivery (CD) 389, 396, 397
continuous integration (CI) 389, 396, 397
continuous uniform distribution 48
continuous variables 57, 134
Controlled-NOT (CNOT) gate 302
Control Plane 405
conventional support vector machines 303
convolutional layers 230
convolutional neural networks (CNNs) 12, 32, 223, 230–232, 246–249, 269, 274
CoreferenceHandler class 294
correlation matrix 143
cost function 8, 176, 203–204
  gradient descent to 177–181
covariance 98
covariance matrix 98
covariant quantum kernels 309
CPUs
  classification algorithms on 14
  machine learning on 13–32
  regression algorithms on 14–15, 267
critical value 132
cross-entropy loss function 8, 203, 225
cross-validation score 4, *5,* 234
cryptography 473
CUDA toolkit 11
cuDNN libraries 11
cumulative distribution function 43, 44, 48
curl command 395, 433, 439
cybersecurity 473

d

DataFrame 15
data points 91
data preprocessing 35
  for machine learning 36
data rescaling method 18–19
data scaling 36
datasets 35
  standardization 37
data transformation 37–50
  logistic data transformation 43
  lognormal transformation 43–44
  MaxAbsScaler 40
  MinMaxScaler 39
  to normal distribution 44–48
  normalization 42–43

quantile transformation 48–50
RobustScaler 40–41
StandardScaler 37–38
using unsupervised learning 98
data visualization 128
DBSCAN (density-based spatial clustering of applications with noise) 252, 262–264
decision tree algorithms 4, 154
deep learning (DL) 1, *3*
algorithms 88, 168
approaches 274
machine learning and. *see* machine learning
Red Hat OpenShift to 452–454
deep residual learning 11
default data mapping 308
density-based spatial clustering of applications with noise (DBSCAN) 252, 262–264
deployment.yaml file 438, 448
descriptive analysis 35
deviation encoding, effect of 68
DevOps tools 396–405, 454
diff() method 85
discrete values 57, 202
distance-based models 36
distribution 69
divisive clustering algorithms 252
Docker Engine 377
Dockerfile 380–381, 383–384, 390, 393, 418, 424, 432, 443, 459
Docker, for machine learning 375–389
batch inference 424–440
build and run 381–389
containerization 376
Dockerfile 380–381
GitHub repository with 454–463
Cloud Kubernetes service 440–452
install 377–378
integrate machine learning models using 396–405
and Kubernetes. *see* Kubernetes
machine learning prediction in real time using 428–440
microservices 375–376
for online inference 393–394, 432–434
and Python APIs with Flask 389–396
training models 415–424
use of 378–379
Docker Hub 377
dummy variable 61

e

EEGs (electroencephalograms) 102
eigenvalues 99, 110
eigenvectors 99, 110
elastic net method 157
electroencephalograms (EEGs) 102
Electronic Numerical Integrator and Computer (ENIAC) 300
element-wise multiplication *232*
ELIZA program 273
embedded methods 22–23, 154–167, 169
elastic net method 157
lasso regression 154–156
ridge regression 156–157
tree-based algorithms 161–165
embedded-type feature selection 131
encoding functions 308, 310
encoding methods 57–58, *58*
backward difference encoding method 72–73
binary encoding method *64,* 64–65
frequency encoding method 65–66
hashing encoding 71–72
Helmert encoding method 58, 63–64
hephAIstos encoding method 77
James-Stein encoding method 74–75
label encoding method 62
leave-one-out encoding 73–74
mean encoding 66–67, *67*
M-estimator encoding 76
one-hot encoding method *61,* 61–62
ordinal encoding method 59–60
probability ratio encoding 70–71
sum encoding method 68
weight of evidence method 68–70
ENIAC (Electronic Numerical Integrator and Computer) 300
equivalent docker command 423
error function 218
error rate 6
EstimatorQNN 338–343
Euclidean distance 116
Euclidian norm 43
exhaustive feature selection 153–154
expending window feature 84
explained variance ratio 102

f

feature engineering techniques 35
feature extraction 97–130
feature rescaling 36–57
time-related features engineering 77–88
work with categorical data. *see* categorical (discrete) data
feature extraction method 19–20, 97–130
advantage 97
with hephAIstos 130
independent component analysis 102–109
linear discriminant analysis 110–115
locally linear embedding method 115–123
manifold learning techniques 125–130

feature extraction method (*cont'd*)
principal component analysis 98–102
t-Distributed Stochastic Neighbor Embedding technique *123,* 123–125
feature rescaling 36–57
data transformation. *see* data transformation
and feature selection techniques 369
SVM model 50–57
feature selection method 20–23, 97, 131–169
embedded methods 22–23, 131, 154–167
filter methods 132–146
flow of *131*
permutation feature importance 165–167
tree-based algorithms for 165
using GPUs 167–168
using hephAIstos 168–169
wrapper methods 21–22, 146–154
feature space 306
filter methods 132–146
advantage 146
ANOVA F-value 137–140
chi-squared test 134–137
disadvantage 146
many more possibilities 146
Pearson correlation coefficient 140–146
using statistical tests 132–134
variance threshold 132
fine needle aspirate (FNA) 138
Fisher's formula 110
.fit_transform() method 63, 64
Flask
GitHub repository with 454–463
Python APIs with 389–396, 428–440
Flask-RESTful APIs 390–392, 428–431
fMRI (functional magnetic resonance imaging) analysis 102
FMs (foundation models) 286
FNA (fine needle aspirate) 138
forward stepwise selection *146,* 146–150
foundation models (FMs) 286
frequency encoding method 65–66
F-score 6–7
F-test statistic 137–140
fully connected layers 230
functional magnetic resonance imaging (fMRI) analysis 102

g

GABAergic interneurons 307
Gambetta's law 465
Gaussian distribution 44–48
quantile transformation with 49
Gaussian scaling 37
Generative Pre-Training (GPT) 286
get_dummies function 61
Gibbs sampling 92
Gini index 161
GitHub repository 33, 294, 402
source code in 454–463
glial cells 220
GPT (Generative Pre-Training) 286
GPUs
classification algorithms on 14, 24–25, 266
feature selection method using 167–168
machine learning on 13–32
regression algorithms on 15, 267–268
gradient descent 177–181, 204

h

Hadamard gate 302, 308
ham messages 274–281
hashing encoding methods 71–72
head() method 255
HelmertEncoder function 63
Helmert encoding method 58, 63–64, 72
hephAIstos
categorical data 17–18
classification algorithms 23–31
data rescaling method 18–19
encoding method 77
feature extraction method 19–20, 130
feature selection method using 20–23, 168–169
function 15–32
installation 13–15
for machine learning 13–32
machine learning algorithms with 264–269
quantum algorithms with 368–372
time series transformation 16–17
time series with 88
training and testing datasets 15–16
Hessian locally linear embedding (HLLE) 121, 122
hierarchical clustering 251, 252
Higgs boson 304
high-performance computing 467
Hilbert–Schmidt space 307
Hilbert space 306
hinge loss function 8
HLLE (Hessian locally linear embedding) 121, 122
Huber loss 8
HuggingFace Pytorch transformers 294
hybrid cloud environment 286
hybrid quantum-classical algorithm 352
hyperparameter optimization 303
hyperplane 306
hypothesis testing 132

i

IBM Cloud Container Registry 446, 447, 450
IBM Cloud Kubernetes service 440–452
IBM Power System 452
IBM Quantum framework 310, 318
ICA (independent component analysis) 102–109
identity function 223
independent component analysis (ICA) 102–109
independent variables 70
inference.py script 382–384, 387, 388–389, 424–426
information value (IV) 69
integer encoding method 59
integrate machine learning models 396–405
intercept-only model 137
internal network 410–414
interneuron phenotypes 307
interneurons 219–220
interquartile range (IQR) 40, *41*
inverse document frequency 278
inverse linear correlation *140*
IQR (interquartile range) 40, *41*
IV (information value) 69

j

James-Stein encoding method 74–75
Jenkins 389, 400
  installation 397–399
  integrate machine learning models using 396–405
  scenario implementation 399–405
JupyterLab 13
Jupyter Notebooks 13, 33, 77, 98, 182, 353, 379

k

KDE (kernel density estimation) 257
Keras 12, 167
  binary logistic regression with 210–211
  logistic regression with 208–210
kernel density estimation (KDE) 257
kernel functions 212, 217, 303, 306
kernel trick. *see* kernel function
k-fold cross-validation 4
k-means clustering algorithm 252–255
k-nearest neighbors (KNN) 274
  imputation method 93–97
  permutation feature importance with 165–166
KNN. *see* k-nearest neighbors
Kubeadm installation 434–435
kubectl 453
kubelet 405
kube-proxy 405
Kubernetes
  application to 448–452
  batch inference 424–428
  CLI 442
  cluster 407–409, 416–418, 424, 426, 435–437
  commands to delete 424
  configuration files for 422–423, 427–428
  containerized machine learning model to 437–440
  distribution 376
  initialization and internal network 410–414
  installation 406–407, 415–416
  internal network 417
  Jobs 415
  kubectl 453
  machine learning with 405–414
  OpenShift and 453–454
  Python APIs ith 428–440
  service on Public Cloud 440–452
  training models 415–424
  virtual machines 415
  vocabulary 405–406
Kubernetes Dashboard 452
Kullback–Leibler divergence 123
kustomization.yaml file 439, 450

l

label encoding method 62, 130
Lagrange multipliers 213
lag variables 79–82
language modeling 287
Large Models, Large Language Models (LMs/LLMs) 286
LDA (linear discriminant analysis) 7, 110–115, 381, 430, 431
learning algorithm 177
learning styles, for machine learning 2–9
  methods *9*
  reinforcement learning 9
  semi-supervised learning 9
  supervised learning. *see* supervised learning
  unsupervised learning 9
least absolute shrinkage and selection operator (lasso) regression 154–156, 169
leave-one-out encoding method 73–74
linear algorithms 131
linear discriminant analysis (LDA) 7, 110–115, 381, 430, 431
linear interpolation 91–92
linear regression 68, 137, 176–202
  gradient descent to cost function 177–181
  implementation 182–202
  math 176–177
  multiple 185–202
  univariate 182–185
linear support vector classification algorithm 157
LLE (locally linear embedding) method 115–123
L-Measure 219

LMs/LLMs (Large Models, Large Language Models) 286
l2 norm 43
locally linear embedding (LLE) method 115–123
local tangent space alignment (LTSA) 121
loc method 255
logistic data transformation 43
logistic function 202
logistic regression 202–211
  binary 202–204, 210–211
  with Keras on TensorFlow 208–210
  multinomial. *see* multinomial logistic regression
  with sklearn 205–208
log loss 8
lognormal cumulative distribution functions 43, *44*
lognormal transformation 43
log transformation 44
long short-term memory (LSTM) 233, 242–246
loss functions 7–9, 203, 225–226
  binary cross-entropy as 352
L2 regularization (ridge regression) 156–157
LSTM (long short-term memory) 233, 242–246
LTSA (local tangent space alignment) 121

m

machine learning (ML) 1. *see also* machine learning algorithms
  application 443–446
  data preprocessing for 36
  Docker. *see* Docker, for machine learning
  feature engineering. *see* feature engineering techniques
  goal 4
  handling missing values in 88–97
  hephAIstos for running. *see* hephAIstos, for machine learning
  learning styles for 2–9
  models 392–393, 431–432, 454–463
  production. *see* production, machine learning in
  Python tools for 9–13
  quantum advantage for 303
  quantum computing and. *see* quantum computing
  Red Hat OpenShift to 452–454
  workflow *2*
machine learning algorithms 94
  artificial neural networks 223–249
  with hephAIstos 264–269
  linear regression 176–202
  logistic regression 202–211
  many more algorithms to explore 249–251
  in quantum computing. *see* quantum machine learning
  rule-based 176
  support vector machine 211–222
  unsupervised 251–264
machine learning operations (MLOps) 396–405
MAE (mean of the absolute errors) 8
magnetoencephalograms (MEGs) 102
Mahalanobis distance transformation 318, **327**
make_blobs() function 253
manifold learners *116*
MAR (missing at random) 92
Masked Language Modeling (MLM) 287, 288
math 176–177
matplotlib 10, 101
MaxAbsScaler 40, 57
maximum-margin hyperplane 212, 306
max-norm 43
MCAR (missing completely at random) 92
mean 37, *37*
  imputation 90
mean encoding methods 66–67, *67*
mean of the absolute errors (MAE) 8
mean shift clustering 252, 257–259
mean squared error (MSE) 8
median imputation 90
MEGs (magnetoencephalograms) 102
M-estimator encoding methods 76
method(s)
  backward difference encoding 72–73
  bag-of-words 274, 278, 280, 296
  binary encoding *64,* 64–65
  data rescaling 18–19
  diff() 85
  elastic net 157
  embedded. *see* embedded methods
  encoding. *see* encoding methods
  feature extraction. *see* feature extraction method
  feature selection. *see* feature selection method
  filter 132–146
  head() 255
  KNN imputation 93–97
  loc 255
  quantum kernel 307
  regularization 154
  shift() 79
  weight of evidence 68–70
  wrapper. *see* wrapper methods
MICE (multivariate imputation by chained equation) imputations 92–93, *93*
microservice approach 375–376
mini-batch k-means clustering algorithm 252, 255–256
Minkowski norm 43
MinMaxScaler 39, 57, 120
misclassification rate 6
missing at random (MAR) 92
missing completely at random (MCAR) 92

missing_method 97
missing not at random (MNAR) 92
missing values handling, in machine learning 88–97
  KNN imputation method 93–97
  linear interpolation 91–92
  multivariate imputation by chained equation 92–93, *93*
  row or column removal 89–90
  statistical imputation 90–91
ML. *see* machine learning
MLLE (modified locally linear embedding) 121, 122
MLM (Masked Language Modeling) 287, 288
MLOps (machine learning operations) 396–405
MLP 224–225
MLP neural networks 233–242, 395, 430, 433–434
MNAR (missing not at random) 92
MNIST dataset 12
mode imputation 90
model-building process 131
modified locally linear embedding (MLLE) 121, 122
monolithic approach 375
Monte Carlo simulations 473
Moore's law 465, 467
MSE (mean squared error) 8
multi-class classification 2, 227–230
multinomial logistic regression 154, 204
  applied to fashion MNIST 204–210
multiple linear regression 176, 181, 185–202
  with Keras on TensorFlow 196–202
  with scikit-learn 188–191
  with *statsmodels,* 192–193
  with TensorFlow 193–196
multivariate imputation by chained equation (MICE) imputations 92–93, *93*

n

namespace.yaml file 437
NaN (Not a Number) 80
natural language generation (NLG) 274
natural language processing (NLP) 97, 251, 273
  BERT. *see* Bidirectional Encoder Representations from Transformers
  deep learning approaches 274
  domain of 296
  messages as spam or ham 274–281
  neural approaches 274
  sentiment analysis 281–286
  statistical approaches 274
  symbolic approach 274
  tools and libraries for 273
Natural Language Toolkit (NLTK) 273, 274
natural language understanding (NLU) 274
negative log-likelihood loss 203
NeuralNetworkClassifier 337, 338, 343, 348
neural network multilayer perceptron 233–242, 381, 395, 430, 431
NeuralNetworkRegressor 337
neural networks 27, 175, 224
  layers *224*
neural NLP approaches 274
NeuroM 219
neuronal classification 307
neuron morphology 219, 307, **327**
new-app command 462
Next Sentence Prediction (NSP) technique 288
NLG (natural language generation) 274
NLP. *see* natural language processing
NLTK (Natural Language Toolkit) 273, 274
NLU (natural language understanding) 274
nodes 405
nominal categorical data 57
non-Gaussian processes 102, 103, 105
nonlinear support vector machines *216,* 216–217, *217*
nonlinear SVM 306
nonparametric quantile transformation 48
normal distribution (Gaussian distribution) 44–48
  quantile transformation with 49
normalization 36, 42–43
normal probability distribution
  right-tailed test with *133*
  two-sided test with *133*
norms, concept of 42
Not a Number (NaN) 80
not fully linearly separable data 214–216
NSP (Next Sentence Prediction) technique 288
null hypothesis 132
numerical variable 57
Numpy 10, 242

o

objective function 36
one-hot encoding method *61,* 61–62
one-hot vector 204
one-to-one process 220
one-to-rest process 220
online inference 393–394, 428–440
Open Java Development Kit (OpenJDK) 397
OpenShift 376, 453
  cluster instance 455–463
  and Kubernetes 453–454
  Red Hat 452–454
  web console 460
OpenShift Enterprise 457

open-source world 12
optimal fitting *3*
optimal hyperplane 212
ordinal categorical data 57
OrdinalEncoder class 60
ordinal encoding method 59–60
ordinal logistic regression 204
ordinary least squares linear regression 160
overfitting *3,* 3–4, 131

p

PACF (partial autocorrelation function) 80, 81
pandas 10, 58, 59, 242
  one-hot encoding method 61
  ordinal encoding method 59
partial autocorrelation function (PACF) 80, 81
partitional clustering 251
PauliFeatureMap 308
Pauli-X gate 302
Pauli-Y gate 302
Pauli-Z gate 302
PCA (principal component analysis) 9, 62, 98–102, 104, 106
Pearson correlation coefficient 20, 140–146, 168
pegasos algorithms 28, 333, 369
pegasos QSVC 333–337
penalization methods 154
p-norm 43
Pods 405, 411, 414, 438
Poisson model 93
polylogarithmic function 306
pooling layers 230
port 8888, 379
precision score 6, 234
pre-trained models 286
primal optimization problem 213
principal cells 219
principal component analysis (PCA) 9, 62, 98–102, 104, 106
probability distribution 225
probability ratio encoding 70–71
probability value (*p*-value) 132, 137
production, machine learning in
  DevOps to MLOPS 396–405
  Docker containers for 375–389
  Kubernetes. *see* Kubernetes
  in real time using Docker and Python 389–396
punctuation 274
p-value (probability value) 132, 137
Python APIs
  with Flask 389–396, 428–440
  with Kubernetes 428–440
Python ecosystem 380
Python language 472
Python tools 9–13
  data manipulation with 10
  Cloud Kubernetes service 440–452
  JupyterLab 13
  Jupyter Notebook 13
  Keras 12
  libraries 10–13
  PyTorch 12
  scikit-learn 10
  TensorFlow 10–12, 288
PyTorch 12, 168, 352

q

qGAN (quantum generative adversarial network) 352–368
Qiskit 304, 305
Qiskit Runtime 305–306
QKA (quantum kernel alignment) 27, 309, 328
QKE (quantum kernel estimation) 307
QNNs (quantum neural networks) 303, 337–351
QPUs. *see* quantum processing units
QSVC 333–337
QSVM 304, 306
qualitative variables 35
quantile function 48
quantile transformation 48–50
quantitative variables 35
quantum computing 474
  algorithms with hephAIstos 368–372
  characteristics of **305**
  mechanics 299
  127-qubit superconducting *305*
  Pegasos QSVC 333–337
  potential use cases for 304
  QNN. *see* quantum neural networks
  quantum Generative Adversarial Network 352–368
  quantum kernel machine learning 306–327
  quantum kernel training 328–333
  quantum machine learning 303–306
quantum feature mapping 311
quantum gates 302
quantum generative adversarial network (qGAN) 352–368
quantum information theory 300
quantum kernel alignment (QKA) 27, 309, 328
quantum kernel estimation (QKE) 307
quantum kernel machine learning 306–327
quantum kernel methods 307
QuantumKernelTrained.fit method 309, 328
QuantumKernelTrainer 309
quantum kernel training 328–333
quantum machine learning 303–306

quantum mapping function 307
quantum mechanics 470
  development of 300
  fundamental equation of 300
  notion of 299
quantum neural networks (QNNs) 337–351
quantum processing units (QPUs) 13–32
  classification algorithms 27
quantum state space 303
quantum systems 469–474
qubits 300, 302, 469–474, 470
  connectivity, *305*

r

radial basis function (RBF) kernel 14
random forest algorithm 161–162
RBAC (role-based access control) 453
RBF (radial basis function) kernel 14
recall score 6, 234
receiver operating characteristic curve (ROC AUC) 147
reciprocal transformation 46
rectified linear unit (ReLU) 223
recurrent neural networks (RNNs) 29, 30–31, 87, 232–233, *233,* 268–269, 274
  long short-term memory (LSTM) 242–246
Red Hat 378, 397
Red Hat OpenShift 452–454
  container platform 454–463
regression algorithms
  binary classification 351
  on CPUs 14–15, 267
  on GPUs 15, 267–268
regression models 2, 93
  with an ϵ-insensitive tube, *218*
  support vector machine using 222
regularization methods 154
reinforcement learning 9
ReLU (rectified linear unit) 223
residual error 93
residual variance 93
responsibility matrix 260
RGB model 230, *230*
ridge regression 156–157
right-tailed test *133*
RMSE (root-mean-squared error) 177, 183
RNNs. *see* recurrent neural networks
RobustScaler 40–41, 57
role-based access control (RBAC) 453
rolling window features 82–83
root-mean-squared error (RMSE) 177, 183
R-squared score 183
rule-based machine learning algorithms 176

s

SamplerQNN 338, 343–348
scaling 36
Schrödinger equation 300, 471
scikit-learn 10, 21, 36, 43, 58, 112
  digits dataset using 123
  Gaussian distribution using 49
  GitHub repository with 454–463
  locally linear embedding using 118
  logistic regression with 205–208
  multiple linear regression with 188–191
  naive Bayes with 280
  support vector machines using 220–222
SciPy 10, 134
Seaborn 10
self-supervised learning 287
semi-supervised learning 9
sentiment analysis 281–286
sequential forward selection (SFS) 148
service_port.yaml file 449–450
service.yaml file 438, 449
SFS (sequential forward selection) 148
SGD optimizer 26, 267
shift() method 79
Shor algorithm 473
sigmoid function 202, *202,* 223, 227
simple linear regression model 137, 177
simple word count 274
sklearn. *see* scikit-learn
skorch 12
slack variable *215*
softmax function 204
softmax loss 8
SOTA (State-of-the-Art) models 286
spam messages 274–281
Spearman's rank correlation 134
.spec.template 423
squared deviation 102
squared error loss 8
square root transformation 45
standard deviation 37, *37,* 98
standardization 36, *37,* 110
StandardScaler 37–38, 57, 100, 130
State-of-the-Art (SOTA) models 286
statistical imputation 90–91
statistical learning 1, 223
statistical NLP approaches 274
statistical tests 132–134
stochastic gradient descent 181, 226, 227, 232
storage systems 466
sudo command 400
sum encoding method 68

summation 223
superposition 300, 301
supervised learning 2–9, 110
  algorithms 2
  confusion matrix *5,* 5–7
  k-fold cross-validation 4
  loss functions 7–9
  objectives 2
  overfitting and underfitting *3,* 3–4
  train/test split 4–5
support vector machines (SVM) 36, 50–57, 211–222, *212*
  application 219–222
  classical 303
  conventional 303
  linearly separable data *212,* 212–214
  nonlinear *216,* 216–217, *217*
  not fully linearly separable data 214–216
  for regression 217–219, *218*
  using linear kernel 280
  using sklearn for regression 222
support vectors 306
SVCLoss 309
SVM. *see* support vector machines
Swiss Roll manifold *116*
symbolic NLP approaches 274
target encoding 66

t

t-Distributed Stochastic Neighbor Embedding (t-SNE) technique *123,* 123–125
TensorFlow 10–12, 167
  BERT for binary text classification using 288–294
  Keras on. *see* Keras
  multiple linear regression with 193–196
  Python 288
term frequency–inverse document frequency (TF-IDF) 278
text summarization, BERT for 294–295
TF-IDF (term frequency-inverse document frequency) 278
three-dimensional space 115
time-related features engineering 77–88
  date-related features 79
  expending window feature 84
  lag variables 79–82
  rolling window features 82–83
  time series data 85
time series 77
  with hephAIstos 88
  transformation 16–17
  trends and seasonal components 85
time transformation 88
tokens 274
top-down clustering 252
TorchConnector 352
"TotalGrayVol" data 36
TrainableFidelityQuantumKernel 309
training Job 415
training model 376
  Kubernetes Cluster 424
  Python application 418–422
train.py file 381–382, 387, 390, 392, 418, 420, 428, 431, 443, 445, 457
train/test split 4–5
tree-based algorithms 36, 57, 161–165
t-SNE (*t*-Distributed Stochastic Neighbor Embedding) technique *123,* 123–125
two-dimensional complex vector space 301
two-sided test *133*

u

Ubuntu 377
underfitting *3,* 3–4
uniform distribution 49
unit vector normalization 42–43
univariate linear regression 176, 182–185, 196
univariate tests 131
universal approximation theorem 225
unrestricted model 137
unsupervised machine learning 9
  algorithms 110, 251–264
  clustering algorithms 252–264
  data transformation using 98
  methods 1

v

Vagrantfile 407, 408, 409
variance 4, 98
variance threshold 132, 168
variational quantum classifier (VQC) 337, 338, 348–351
variational quantum regressor (VQR) 337
vector norms 42
video games, types of 57
virtual machines (VMs) 376, 415, 434
VMs (virtual machines) 376, 415, 434
VQC (variational quantum classifier) 337, 338, 348–351
VQR (variational quantum regressor) 337

w

Ward 252
wave mechanics 299, 300, 470
wave-particle duality 299
"wave–particle duality" 299
WCSS (within-cluster sum of squares) 253

weight of evidence (WoE) method 68–70
within-cluster sum of squares (WCSS) 253
WoE (weight of evidence) method 68–70
wrapper methods 21–22, 131, 146–154, 168–169
  backward elimination 151–153
  exhaustive feature selection 153–154
  forward stepwise selection *146,* 146–150

y

YAML files 437–439
Yeo-Johnson transformation 47

z

ZFeatureMap 308
Z-score normalization 37
ZZFeatureMap 308, 318