

Index

Note: Page numbers followed by “b” “f” and “t” refer to boxes, figures and tables, respectively.

A

- “ABCs” of DL, 324
- Accuracy, 83, 266, 266*t*
- ACF chart. *See* AutoCorrelation
 - Function chart (ACF chart)
- Activation function, 320, 320*f*
- “Actual Class”, 267
- AdaBoost, 157–158, 158*f*, 159*f*
- Additive time series, 401–402, 403*f*
- Advanced statistical techniques, 450–451
- Aggregate operator, 404
- AGI. *See* Artificial general intelligence (AGI)
- AI. *See* Artificial intelligence (AI)
- AI Winter (1960s–2006), 310–311
 - RapidMiner XOR example, 311*f*
- Algorithms, 14–18
- Andrews curves, 61–63, 62*f*
- Andrews plot, 61
- ANNs. *See* Artificial neural networks (ANNs)
- Anomaly detection, 13*t*, 17–18, 372, 447–453, 526
 - causes of outliers, 448–449
 - LOF technique, 460–464
 - outlier detection
 - using data science, 451–453
 - density-based, 457–460
 - distance-based, 453–457
 - using statistical methods, 450–451
- Anscombe’s quartet, 48, 49*f*
- Append operator, 508
- Apply forecast operator, 424–425
- Apply Model operator, 364, 372, 382, 418
 - Apriori algorithm, 206–211. *See also* Naïve Bayesian algorithm; FP-Growth algorithm
 - frequent itemset generation using, 207*f*, 208–211, 208*f*
 - rule generation, 209–211
 - Apriori algorithm, 525
 - Area under curve (AUC), 83–84, 263, 266–268
 - ARIMA. *See* Auto Regressive Integrated Moving Average (ARIMA)
 - Artificial general intelligence (AGI), 314
 - Artificial intelligence (AI), 2–4, 3*f*, 307
 - to engineering, 308*b*
 - models using deep learning, 335
 - spring and summer of, 314–315
 - Artificial neural networks (ANNs), 124–135, 316, 523
 - implementation, 130–132
 - performance vector, 134*f*
 - works, 128–130
 - Assembling known ratings, 349
 - Assimilation, 35–36
 - Association analysis, 11, 13*t*, 16, 35–36, 351, 525
 - Apriori algorithm, 206–211
 - concepts of mining association rules, 201–205
 - FP-Growth algorithm, 211–219
 - Association analysis, 348
 - Association rules
 - creation, 217–218, 218*f*, 219*f*
 - learning, 199
 - Attributes, 23, 374, 505–506
 - and examples, 495
 - independence, 120
 - naming for De-Pivot, 382
 - understanding relationship between, 64
 - weighting, 467
 - AUC. *See* Area under curve (AUC)
 - Australian Beer Production time series dataset, 412
 - Auto Regressive Integrated Moving Average (ARIMA), 398–399
 - algorithm, 528
 - ARIMA(*p,d,q*) model, 423–424
 - trainer, 424
 - Autocorrelation, 398–399, 419–420
 - AutoCorrelation Function chart (ACF chart), 420
 - Autoencoders, 334, 337*f*
 - Automatic iterative methods, 6
 - Automatic Multilayer Perceptron (AutoMLP), 130–131
 - Autoregressive integrated moving average, 418–425
 - AutoRegressive models, 420–421
 - differentiation, 422
 - implementation, 424–425
 - moving average of error, 423
 - stationary data, 421–422
 - Average method, 408
 - Average pooling, 327–328
 - Axon, 125–126

B

- Backpropagation, 128, 313, 316, 329
 - need for, 321–322
- Backward elimination, 484–489, 529

Bagging technique, 154–155
 Balanced accuracy, 511
 Balanced dataset, 509
 Bayes' theorem, 113
 predicting outcome, 115–117
 Bayesian belief network, 120
 BI. *See* Business intelligence (BI)
 Biased matrix factorization (BMF),
 371–372
 Binning technique, 27
 Binomial
 classification, 273
 operator, 463–464
 variables, 196
 Biological neurons, 125*b*
 Bivariate plots, 502–503
 Blog authors gender prediction,
 294–304
 data preparation, 295–297
 applying trained models to
 testing data, 302–303
 builds model, 298–302
 identifying key features,
 297–298
 preparing test data for model
 application, 302
 gathering unstructured data, 295
 raw data for blog classification
 study, 296*t*
 training and testing predictive
 models, 301*f*
 BMF. *See* Biased matrix factorization
 (BMF)
 Boosting, 156–157
 Bootstrap aggregating technique,
 154–155
 Bootstrapping, 154–155
 Boston housing dataset
 attributes, 171*t*
 sample view, 171*t*
 Bot, 448
 Box whisker plot, 50–51
 Bubble chart, 57–58, 59*f*
 Business intelligence (BI), 4, 7

C

CAE. *See* Computer-aided
 engineering (CAE)
 Cartesian space, 54
 Categorical data, 43
 Causation, 24
 Center-based clustering.
 See Prototype-based clustering
 Center-based density, 239–240
 Centered cosine
 coefficient metric, 359–360
 similarity, 356–357
 Central data point, 46
 Central point for each attribute, 63
 Central tendency measure, 44–45
 Centroid clustering. *See* Prototype-
 based clustering
 Centroid prototype, 226–227
 Chi square-based filtering, 467–468,
 480–483
 converting golf example set into
 nominal values for, 481*f*
 process to rank attributes of Golf
 dataset by, 482*f*
 Chi-square algorithm, 529
 Chi-square test, 120
 Class conditional probability
 calculation, 115
 of humidity, outlook, and wind,
 116*t*
 of temperature, 115*t*
 Class label. *See* Output variable
 Class selection, 91
 Classic golf dataset, 69*t*
 Classical decomposition, 403–404
 Classification, 13*t*, 14–16
 model, 353, 383–384, 523
 performance, 264, 271
 tasks, 29
 techniques, 11, 452
 trees. *See* Decision trees
 Click fraud detection in online
 advertising, 449*b*
 Clustering, 11, 13*t*, 16, 221, 452,
 525–526
 k-means clustering, 226–238
 working, 227–234
 implementation, 234–238
 DBSCAN clustering, 238–247
 working, 240–243
 implementation, 240–243
 self-organizing map (SOM),
 247–259
 working, 249–252
 implementation, 252–259
 for object reduction, 223
 to reduce dimensionality,
 222–223
 techniques, 447
 types, 223–225
 Cold start problem, 349, 358
 Collaborative filtering, 351–373, 354*f*
 neighborhood-based methods,
 354–366
 Collaborative filtering algorithm,
 527
 Comma-separated values (CSV),
 78–79, 497–498, 498*f*
 Competitive SOMs, 248–249
 Computer-aided engineering (CAE),
 308–309, 308*f*
 Confidence
 of response, 267
 rule, 203–204
 Confusion matrix, 263–266,
 273–274
 Consumer Price Index (CPI),
 505–506
 Content recommendation, 345–346
 Content-based filtering, 352–353,
 373–389, 374*f*
 algorithm, 527
 Content-based recommendation
 engines, 388
 Content-based recommenders, 378,
 389
 dataset, 378
 implementation, 378
 Predicted rating using content-
 based filtering, 383*f*
 recommender process using
 content-based filtering, 379*f*
 supervised learning models,
 383–389
 “Contextually adaptive” system, 315
 Contingency tables, 120, 120*t*, 481
 Continuous attributes, 118–119
 Golf dataset with, 119*t*
 mean and deviation, 119*t*
 Continuous data, 42
 Convex hull, 139, 139*f*
 Conviction rule, 204–205
 Convolution, 324–325, 325*f*, 326*f*
 combining convolution with
 activation function, 329*f*
 multiple filters of convolution,
 330*f*
 Convolutional neural networks, 330*f*
 convolution, 324–325
 dense layer, 331, 331*f*
 dropout layer, 331, 334*f*
 Correlation, 24
 Correlation, 47–48, 47*f*
 Cosine similarity, 107, 364

Cost function, 329–330
 CPI. *See* Consumer Price Index (CPI)
 Crawl Web operator, 291
 Credit rating, 77
 Credit scoring, 77
 Cross Industry Standard Process for
 Data Mining (CRISP-DM),
 19–20, 20*f*
 Cross selling, 200*b*
 Cross-entropy cost function, 318
 Cross-sectional data, 395–396, 397*f*
 CSV. *See* Comma-separated values
 (CSV)
 Cycle, 401

D

DARPA, 314
 Data
 cleansing practices, 25
 engineering, 7
 errors, 448
 exploration, 25, 39, 502
 datasets, 40–43
 descriptive statistics, 43–48
 objectives, 40
 roadmap, 63–64
 importing and exporting tools,
 497–501
 point, 23
 preparation, 25–29, 40, 77–81,
 94, 108, 121, 131, 141, 144,
 172–173, 193, 215–216,
 363–364, 378–379,
 504–505
 feature selection, 28
 missing values, 26–27
 outliers, 27
 transformation, 27
 quality, 25
 sampling, 28–29
 series, 407
 set or data table, 495
 splitting, 68–73, 70*f*
 transformation tools, 504–509
 types and conversion, 27
 understanding, 40
 warehouses, 25
 Data mining, 1, 4, 199
 algorithm comparison
 anomaly detection, 526
 association analysis, 525
 classification, 523
 clustering, 525–526
 deep learning, 526
 feature selection, 529
 recommenders, 527
 regression, 524
 time series forecasting, 528
 framework, 19–20
 Data science, 1–7, 5*f*
 algorithms, 12, 26–27
 case for, 8–9
 complex questions, 9
 dimensions, 8–9
 volume, 8
 classification, 10–11, 10*f*
 getting started with, 12–18
 outlier detection using, 451–453
 process, 19, 467, 501
 application, 34–36
 data mining process, 21*f*
 prior knowledge, 21–24
 knowledge, 36–37
 modeling, 29–34, 29*f*
 reducing uncertainty, 67*b*
 tasks, 40
 and examples, 13*t*
 Data visualization, 48–63, 450
 high-dimensional data, 60–63
 multivariate visualization, 53–59
 tools, 501–503, 501*f*
 bivariate plots, 502–503
 finishing data import, 500*f*
 metadata visible under statistics
 tab, 502*f*
 multivariate plots, 503
 univariate plots, 502
 univariate visualization, 50–53
 Data-driven approaches, 407–413.
 See also Model-driven
 forecasting methods
 exponential smoothing, 409–412
 simple forecasting methods,
 407–409
 Data-driven forecasting methods,
 398
 Datasets, 23, 24*t*, 40–43, 361–362,
 378, 385
 attributes, 501
 dividing into training and testing
 samples, 81
 MovieLens datasets, 362*t*
 organization, 63
 preparation, 336

types of data, 42–43
 categorical or nominal data, 43
 numeric or continuous data, 42
 Davies–Bouldin index, 234
 DBSCAN. *See* Density-Based Spatial
 Clustering of Applications
 with Noise (DBSCAN)
 DBSCAN clustering, 238–247.
 See also *k*-Means clustering
 algorithm, 525–526
 implementation, 234–238
 clustering operator and
 parameters, 244
 data preparation, 244
 evaluation, 244
 execution and interpretation,
 245–247
 optimizing parameters, 242
 varying densities, 242–243
 working principle, 240–243
 classification of data points,
 241–242
 clustering, 242
 defining epsilon and MinPoints,
 241
 De-Pivot operator, 382, 508
 Decision trees, 66–87, 74*f*, 75*f*, 82*f*
 algorithm, 523
 approach, 15
 for Golf data, 72*f*
 for Golf dataset, 76*f*, 90*f*
 implementation, 73–86
 model, 81–84
 operator, 152
 path, 66–67
 works, 66–73
 Decision-making model, 34
 Decomposed data, forecasting using,
 406–407
 Decomposition algorithm, 528
 Deep learning (DL), 11, 17, 307,
 337*f*, 339*f*. *See also* Machine
 learning (ML)
 AI Winter (1960s–2006),
 310–311
 algorithm, 526
 convolutional neural networks,
 324–331, 330*f*
 deep architectures
 AI models using, 335
 autoencoders, 334, 337*f*
 RNN, 332–334, 333*f*

Deep learning (DL) (*Continued*)
 implementation, 335–341
 applying model, 340–341
 dataset preparation, 336
 modeling, 338–340
 results, 341
 Mid-Winter Thaw (1980s),
 311–314
 spring and summer of AI,
 314–315
 systems, 315
 working principle, 315–335
 adding hidden layers and need
 for backpropagation,
 321–322
 gradient descent, 317–321
 regression models as neural
 networks, 316–317, 317*f*
 softmax, 323–324
 Define, Measure, Analyze, Improve,
 and Control (DMAIC), 19–20
 Demographic attributes, 271
 Dendrite, 125–126
 Dendrogram, 225
 Dense information comprehension,
 48
 Dense layer, 331, 331*f*
 Density chart, 58–59, 59*f*
 Density-based algorithm, 526
 Density-based outlier detection, 452,
 457–460. *See also* Distance-
 based outlier detection
 implementation, 459–460
 data preparation, 459
 detect outlier operator,
 459–460
 execution and interpretation,
 460
 working principle, 458–459
 Density-Based Spatial Clustering of
 Applications with Noise
 (DBSCAN), 225–226
 Density-clustering, 225
 algorithm, 238–239
 mode, 246*f*
 visual output, 246*f*
 Descriptive Analytics Technique,
 293–294
 Descriptive data science, 11
 Descriptive modeling, 20–21
 Descriptive statistics, 6–7, 25, 39,
 43–48, 46*f*
 multivariate exploration, 46–48

 univariate exploration, 44–45
 Deviation, 45
 chart, 60–61, 62*f*
 Dichotomization process, 504
 Dimension reduction method, 467,
 475–477
 Dimensional slicing, 7, 63
 Dimensionality, curse of, 28
 Direct marketing (DM), 263*b*
 Direct method, 89–90
 Directed data science, 10
 Discretization, 505
 by binning operator, 505
 by frequency operator, 505
 operator, 506*f*
 by size operator, 505
 by user specification, 505
 Distance-based algorithms, 451, 526
 Distance-based outlier detection,
 453–457. *See also* Density-
 based outlier detection
 implementation, 454–457
 data preparation, 456
 detect outlier operator,
 456–457
 execution and interpretation,
 457
 working principle, 454
 Distribution chart, 52–53, 55*f*
 Distribution model, 225
 Distribution-based clustering.
 See Model-based clustering
 Distribution-based outlier, 452
 DL. *See* Deep learning (DL)
 DM. *See* Direct marketing (DM)
 DMAIC. *See* Define, Measure,
 Analyze, Improve, and
 Control (DMAIC)
 Document, 285
 clustering, 222
 matrix, 106–107
 vector, 106–107, 285–286
 “Dot product” formulation, 141
 Dropout
 layer, 331, 334*f*
 prediction, 149*b*

E

Eigenvalue analysis of covariance
 matrix, 472
 Empty clusters, 233
 Ensemble learners, 15, 148–161,
 149*f*, 156*f*

 achieving conditions for ensemble
 modeling, 151–152
 data mining process, 153*f*
 implementation, 152–160
 AdaBoost, 157–158
 boosting, 156–157
 bootstrap aggregating or
 bagging, 154–155
 ensemble by voting, 152–153
 random forest, 159–160
 wisdom of crowd, 148–149
 works, 150–152
 Ensemble model, 28–29, 33–34,
 523
 Entertainment recommendation, 346
 Entire dataset, 495
 Entropy, 67, 67*f*, 81–83
 Euclidean distance, 103–104,
 228–230
 Evaluation, 350
 framework, 19–20
 of model, 32–33
 Exclusive partitioning clusters, 223
 Execute R operator, 412, 427
 Explanatory modeling, 20–21
 Exploratory data analysis. *See* Data
 exploration
 Exploratory visualization, 7
 Exponential smoothing, 407,
 409–412
 algorithm, 528
 Holt-Winters’ three-parameter, 412
 Holt’s two-parameter, 411–412
 Extract Example Set, 436
 Extracting meaningful patterns, 4–5

F

Fast brute-force approach, 468
 Feature selection method, 165–166,
 174, 467, 468*b*
 Chi-square-based filtering,
 480–483
 classification, 468–470
 information theory-based filtering,
 477–480
 PCA, 470–477
 taxonomy of, 469*f*
 wrapper-type feature selection,
 483–489
 Features, 374
 selection, 11, 18, 28, 529
 Filter bubble, 350–351
 Filter Example operator, 436

Filter Examples Range operator, 172
 Filter model, 469
 Filter-based algorithm, 529
 Filter-type methods, 467–468
 Filtering, 382
 prospect, 77–86
 First order differencing, 422
 Forecast(ing), 17, 407
 errors, 407, 439–441
 Forward selection, 484, 529
 FP. *See* Frequent pattern (FP)
 FP-Growth algorithm, 211–219, 525
 data science process, 216f
 frequent itemset generation,
 214–215
 generating FP-tree, 211–215
 implementation, 215–219
 FP-Growth operator, 217
 FP-Tree, 211, 212f, 213f, 214f
 Frequent itemset
 generation, 214–215
 using Apriori principle, 207f,
 208–211, 208f
 support calculation, 209t
 Frequent pattern (FP), 205
 Fully connected layers, 330, 331f
 Function-fitting approach, 165–166
futureMonths process, 434–436
 Fuzzy clusters, 224

G

Gain ratio, 83, 477–478
 GAN. *See* Generative adversarial
 network (GAN)
 Gaussian distribution. *See* Normal
 distribution
 GDP. *See* Gross domestic product
 (GDP)
 Gender prediction of blog authors,
 294–304
 Generalization, 4–5
 Generate attributes, 178–179
 Generate Data operator, 514
 Generative adversarial network
 (GAN), 335
 Genre attributes, 380–382
Get Pages operator, 291
 Gini index, 67, 83
 Global baseline, 366
 matrix factorization, 366–373
 Golf dataset with modified
 temperature and humidity
 attributes, 114t

GPUs. *See* Graphics processing units
 (GPUs)
 Gradient descent technique,
 169–170, 316–321
 Graphical user interface (GUI),
 492–493
 GUI-driven application, 496
 launch view of RapidMiner 6.0,
 492f
 Graphics processing units (GPUs),
 314
 Greedy methodology, 517–518
 Gross domestic product (GDP),
 248–249
 GUI. *See* Graphical user interface
 (GUI)

H

“Handcrafted knowledge” systems,
 315
 Hidden layers, 321–322, 438
 combining multiple logistic
 regression models, 322f
 Hierarchical clusters, 223, 225
 High-dimensional data visualization,
 60–63
 Andrews curves, 61–63, 62f
 deviation chart, 60–61, 62f
 parallel chart, 60, 61f
 Histogram, 40, 50, 51f
 class-stratified, 52f
 Home price prediction, 166b
 Hybrid recommenders, 389–390,
 390f
 Hyperplane, 135–136, 136f, 140f
 Hypothesis testing, 7

I

ID3. *See* Iterative Dichotomizer 3
 (ID3)
 Identifiers, 24
 IDF. *See* Inverse document frequency
 (IDF)
 Impute Missing Values operator, 512
 Independent variables, 397
 Info gain algorithm, 529
 Information gain, 83, 477–478
 information gain–based filtering
 method, 467–468
 Information theory-based filtering,
 477–480
 Integer, 42
 Internet of things, 90

Interpretation framework, 19–20
 Interpreting results, 40
 Inverse document frequency (IDF),
 284
 Iris dataset, 40–42, 46
 and descriptive statistics, 44t
Iris setosa, 40–41
Iris versicolor, 40–41, 41f
Iris virginica, 40–41
 Item *k*-NN recommender process,
 363–364, 363f, 365f
 Item profile
 building, 374–375, 375t, 377t
 user profile computation,
 375–383, 376f, 378t
 preparation, 379–382
 Item-based collaborative filtering,
 359–361
 normalized ratings and similarity,
 360t
 transposed ratings matrix, 359t
 Item-based neighborhood method,
 351–352
 Itemsets, 202–205, 206f
 Iterative Dichotomizer 3 (ID3), 68
 Iterative process, 22

J

Jaccard similarity, 106–107
 Join operator, 404, 425, 508
 Joint entropy, 73

K

k-means clustering, 16, 226–238.
 See also DBSCAN clustering
 algorithm, 525–526
 evaluation of clusters, 233–234
 implementation, 234–238
 special cases, 232–233
 working principle, 227–234
 calculating new centroids, 231
 data points assignment,
 228–230, 230f
 initiate centroids, 228
 repeating assignment and
 calculating new centroids, 232
 termination, 232
k-medoids clustering process,
 290–291
k-nearest neighbor (*k*-NN), 6,
 98–111, 110f, 454
 algorithm, 26–27, 223, 523
 data mining process for, 109f

- k*-nearest neighbor (*k*-NN)
 - (Continued)
 - implementation, 108–110
 - performance vector for, 110*f*
 - works, 100–107
- Keras extension for RapidMiner, 336
- Keras operator, 338–339, 339*f*
- Keyword clustering, 290–294
 - apply clustering technique, 293–294
 - data preparation, 292
 - gathering unstructured data, 291–292
- Knowledge discovery, 1, 4–5
- Kohonen networks, 248
- L**
- L1-norm regularization, 184
- L2-norm regularization, 183, 184*f*
- Label, 24
- Laplace correction, 118
- Lasso regression, 184
- Latent factors, 354
 - model, 352, 366–367, 367*f*
- Latent matrix factorization
 - algorithm, 527
- Learn-One-Rule technique, 91–93
- Learning. *See also* Deep learning (DL)
 - algorithms, 6, 30–32
 - regression model, 32*f*
 - perceptron, 309–310
 - process, 485
 - rate, 319–320
 - function, 256
- Level, 401
- Lexical substitution process, 286–288
- LibSVM model, 303
- Lift
 - charts, 263
 - curves, 268–270, 271*t*, 272*f*, 277*f*, 278*f*
 - rule, 204
- Linear regression, 9, 16, 185, 186*f*.
 - See also* Logistic regression
 - algorithm, 524
 - checkpoints to ensuring regression model validity, 180–185
 - implementation, 172–179
 - line, 182*f*
 - model, 9, 27, 165–166, 415, 416*f*
 - operator, 173*f*, 174, 487
 - technique, 31
 - works, 167–171
- Linearly non-separable dataset, 144–147
- Linearly separable dataset, 141–144
- Lloyd's algorithm, 227
- Lloyd–Forgy algorithm, 227
- Local outlier factor technique (LOF technique), 460–464
 - algorithm, 526
 - implementation, 462–464
 - data preparation, 462–463
 - detect outlier operator, 463
 - results interpretation, 463–464
- LOF technique. *See* Local outlier factor technique (LOF technique)
- Logistic regression, 16, 185–196.
 - See also* Linear regression
 - algorithm, 524
 - finding sigmoid curve, 188–190
 - growth of logistic regression applications, 186*f*
 - implementation, 193–195
 - model, 316–317
 - points for logistic regression modeling, 196
 - setting up RapidMiner process, 194*f*
 - works, 187–192
- Logit function, 189, 196
- Loop operator, 438
- LSTAT, 177–178
- M**
- m*-lag first order differencing, 422
- MA(*q*) model. *See* Moving Average with *q* lags model (MA(*q*) model)
- Machine breakdowns, predicting and preventing, 90*b*
- Machine learning (ML), 1–4, 3*f*.
 - See also* Deep learning (DL)
 - algorithms, 2–4, 9, 313
 - methods, 429–438
 - lagged inputs and target, 429*f*
 - neural network autoregressive, 436–438
 - windowing, 430–436
 - ML-based prediction model, 350
 - systems, 315
- Macro in RapidMiner, 434–436
- MAE. *See* Mean absolute error (MAE)
- Mahalanobis distance, 450–451
- MAPE. *See* Mean absolute percentage error (MAPE)
- Market basket analysis, 11, 199–200
- Marketing, 222
- MASE. *See* Mean absolute scaled error (MASE)
- Matrix factorization, 354, 366–373, 371*f*, 372*f*
 - decomposition of ratings matrix into latent factor matrices, 368*f*
 - implementation, 370
- Max pooling, 327–328, 328*f*
- Mean, 44
- Mean absolute error (MAE), 350, 441–442
- Mean absolute percentage error (MAPE), 442
- Mean absolute scaled error (MASE), 442–443
- Median, 44
- Median value (MEDV), 171, 485
- Medoid, 290–291
- MEDV. *See* Median value (MEDV)
- Meta learning, 148
- MetaCost operator, 86, 194–195, 195*f*
- Metadata, 494
- Mid-Winter Thaw (1980s), 311–314
- Mining. *See also* Text mining
 - association rules concepts, 201–205
 - itemsets, 202–205
 - rule generation process, 205
 - process, 282–283
- Missing values, 26–27
- Mixture of Gaussians, 225
- ML. *See* Machine learning (ML)
- MLP. *See* Multi-layer perceptron (MLP)
- MLR. *See* Multiple linear regression (MLR)
- Mode, 44
- Model Combiner operator, 390
- Model evaluation
 - confusion matrix, 263–266
 - DM, 263*b*
 - implementation, 271–276
 - data preparation, 271–273
 - evaluation, 273
 - execution and interpretation, 273–276

- modeling operator and
 - parameters, 273
 - lift curves, 268–270, 271*t*, 272*f*
 - ROC curves and AUC, 266–268
- Model-based clustering, 225
- Model-driven forecasting methods, 398–399, 413–429. *See also*
 - Data-driven approaches
- autoregressive integrated moving average, 418–425
- global and local patterns, 415*f*
- regression, 415
 - implementation in RapidMiner, 417–418
 - with seasonality, 415–418
- seasonal ARIMA, 426–429
- seasonal attributes, 417*f*
- Modeling, 5, 29–34, 29*f*, 338–340
 - ensemble, 33–34
 - evaluation of model, 32–33
 - learning algorithms, 30–32
 - process, 20–21
 - training and testing datasets, 30
- Moore's Law, 1
- MovieLens
 - datasets, 362*t*
 - ratings matrix dataset, 378
- Moving average
 - of error, 423
 - smoothing, 408
- Moving Average with q lags model (MA(q) model), 423
- Multi-layer perceptron (MLP), 311
- Multicollinearity, 468
- Multiple linear regression (MLR), 170
- Multiplicative time series, 401–402, 403*f*
- Multivariate exploration, 46–48
 - central data point, 46
 - correlation, 47–48, 47*f*
- Multivariate plots, 503
- Multivariate visualization, 53–59
 - bubble chart, 57–58, 59*f*
 - density chart, 58–59, 59*f*
 - scatter matrix, 56–57, 58*f*
 - scatter multiple, 55–56, 57*f*
 - scatterplot, 54–55, 56*f*

N

- n -grams models, 289
- Naïve Bayesian algorithm, 15, 111–124, 123*f*, 523. *See also*
 - Apriori algorithm; FP-Growth algorithm
- algorithm, 468
 - data mining process, 122*f*
 - distribution table output, 123*f*
 - implementation, 121–122
 - works, 113–120
- Naïve Bayesian operator, 121, 273
- Naïve forecast, 442
- Naïve method, 407
- Natural language processing method (NLP method), 282*b*
- Neighborhood methods, 351–352, 360
- Neighborhood users, deducing rating from, 357–358
- Neighborhood-based method, 354–366
 - dataset, 361–362
 - implementation steps, 362–364
- Nested operator, 81
- Nested process, 513
- Neural net, 130–131
- Neural network autoregressive model (NNAR model), 436–438, 438*f*
- Neural networks, 15. *See also*
 - Artificial neural networks (ANNs)
- models, 26–27
- regression models as, 316–317, 317*f*
- Neuron, 125–126, 248
- NLP method. *See* Natural language processing method (NLP method)
- NNAR model. *See* Neural network autoregressive model (NNAR model)
- Noise, 401, 404
- Nominal
 - to binominal operator, 504
 - data, 43
 - to numerical operator, 504
- Non-domain-focused program, 282
- Non-systematic component, 397–398
- Nonlinear optimization techniques, 190
- Normal distribution, 45, 52–53
- Normalized ratings matrix, 356*t*
- Numeric data, 42
- Numerical
 - to binominal operator, 504
 - to polynomial operator, 505

O

- Observation, 407
- Online advertising, click fraud detection in, 449*b*
- Online analytical processing (OLAP), 7
- Operators, 78, 495–496
- Optical character recognition, 127*b*
- Optimization tools, 512–520, 515*f*
 - configuring grid search optimizer, 516*f*
 - progression
 - of genetic search optimization, 519*f*
 - of grid search optimization, 517*f*
 - of quadratic greedy search optimization, 518*f*
 - searching for optimum within fixed window that slides across, 516*f*
 - simple polynomial function to demonstrate optimization, 514*f*
- Optimize Parameters operator, 424–425
- Outliers, 27, 233, 447
 - causes, 448–449
 - detection using data science, 451–453
 - detection using statistical methods, 450–451
 - watch out for, 64
- Output variable, 10
- Overfitting, 32–33, 71, 84–85, 181
- Overlapping clusters, 223

P

- "Padding" process, 327, 327*f*
- Parallel axis, 60
- Parallel chart, 60, 61*f*
- PCA [Principal component analysis \(PCA\)](#)
- Pearson correlation, 356–357
 - coefficient, 47–48
 - coefficient metric, 359–360
- Pearson similarity measure, 364
- Perceptron, 125, 309–311, 310*f*
 - learning rule, 309–311
- Performance
 - criterion, 232–233
 - evaluation, 382
 - operator, 382
- Pivot data, 63

Pivot operator, 507–508, 508f
 PMML. *See* Predictive Model Markup Language (PMML)
 Polynomial regression, 415
 Polynomial data type, 43
 Post-processing, 233
 Post-pruning, 72–73
 Precision, 265, 266t
 “Predicted Class”, 267
 Predictive analytics, 1
 Predictive Model Markup Language (PMML), 34–35
 Predictive modeling, 20–21
 Predictor variables, 397
 Preprocessing
 DBSCAN clustering, 238–247
 framework, 19–20
 k-Means clustering, 226–238
 Principal component analysis (PCA), 18, 456, 467, 470–477, 471f, 529
 implementation, 472–477
 data preparation, 474
 execution and interpretation, 474–477
 PCA operator, 474
 interpreting RapidMiner output for, 478f
 working principle, 470–472
 breakfast cereals dataset for
 dimension reduction, 473t
 Prior knowledge, 21–24
 causation *vs.* correlation, 24
 data, 23–24, 24t
 objective, 22
 subject area, 22–23
 Prior probability calculation, 115
 Probabilistic clusters. *See* Fuzzy clusters
 Probability mass function, 150
 Product recommendations, 345
 Prototype-based clustering, 224–225
 and boundaries, 229f
 Proximity measure, 102–107
 Pruning process, 72–73

Q

Quartile, 50–51, 53f
 class-stratified quartile plot, 54f

R

R software package
 integration with, 520–521

 script for Holt-Winters’ forecasting, 413
 Random forest, 159–160, 160f
 Random walk
 with drift, 424
 model, 424
 Randomized algorithms, 12
 Range, 45
 RapidMiner, 12–13, 271, 284, 290–291, 404, 454, 475, 485, 491
 data
 importing and exporting tools, 497–501
 transformation tools, 504–509
 visualization tools, 501–503
 implementation in, 417–418
 integration with R, 520–521
 optimization tools, 512–520
 process, 74, 86, 253, 256, 371–373, 382, 427
 for rule induction, 96f
 for tree to rules operator, 97f
 sampling and missing value tools, 509–512
 Studio, 491
 user interface and terminology, 492–497
 wrapper function logic used by, 487f
 Rating prediction, 350
 operator, 364, 372, 382
 Ratings matrix, 347–349, 347t, 355t, 377t
 assemble known ratings, 349
 evaluation, 350
 rating prediction, 350
 Raw German Credit Data view, 78t
 Read CSV operator, 497–498
 Read Excel operator, 78, 79f
 Rebalance sub-process, 511
 Recall, 265–266, 266t
 Receiver operator characteristic curve (ROC curve), 83–84, 263, 266–268
 classifier performance data needed for building, 268t
 Recommendation engines, 11, 13t, 17, 343–348, 347f, 351–353
 applications, 345–346
 balance, 350–351
 building item profile, 374–375, 375t, 377t

 collaborative filtering, 353–373, 354f
 comparison, 392t
 content-based filtering, 373–389, 374f
 content-based recommenders, 378
 global baseline, 366
 hybrid recommenders, 389–390, 390f
 item-based collaborative filtering, 359–361
 matrix factorization, 370, 371f, 372f
 neighborhood based method, 361
 ratings matrix, 348–349
 supervised learning models, 385
 taxonomy, 351f
 user-based collaborative filtering, 355, 360–361
 Recommendation model, 389–390
 Recommenders, 527
 modeling, 382
 Rectified linear unit (RELU), 320–321
 Recurrent neural networks (RNN), 332–334, 333f
 Regression, 13t, 165, 168f, 415, 524
 methods, 11, 485, 512–513
 linear regression, 166–185
 logistic regression, 185–196
 predicting home prices, 166b
 model, 353, 489
 activation function to regression
 “network”, 318f
 as neural networks, 316–317, 317f
 predictive models, 399
 with seasonality, 415–418
 tasks, 29
 trees, 66
 Regularization, 181, 369–370
 Reinforcement learning (RL), 315
 Relevance, 265, 266t
 RELU. *See* Rectified linear unit (RELU)
 Remember operator, 436
 Rename operator, 178–179
 Repeated Incremental Pruning to Produce Error Reduction approach (RIPPER approach), 91
 Replace (Dictionary) operator, 78–79

Replace missing values operator, 512
 Repository, 494
 Representative models, 5
 Retrieve operator, 172
 Ridge regression, 183
 RIPPER approach. *See* Repeated Incremental Pruning to Produce Error Reduction approach (RIPPER approach)
 RL. *See* Reinforcement learning (RL)
 RMSE. *See* Root mean square error (RMSE)
 RNN. *See* Recurrent neural networks (RNN)
 Roadmap for data exploration, 63–64
 ROC curve. *See* Receiver operator characteristic curve (ROC curve)
 Root mean square error (RMSE), 350, 442
 Rule development, 91–92
 Rule generation, 205, 209–211
 Rule induction, 15, 87–98
 algorithm, 523
 approaches to developing rule set, 89–90
 implementation, 94–98
 model, 388, 388f
 modeling operator, 94–95
 works, 91–93
 Rule Model window, 95
 Rule set development, 93

S

Sample, Explore, Modify, Model, and Assess (SEMMA), 19–20
 Sampling, 28, 121
 Sampling and missing value tools, 509–512
 decision trees on well-balanced data, 510f
 rebalanced data and resulting improvement in class recall, 512f
 snapshot of imbalanced dataset, 510f
 unbalanced data and resulting accuracy, 511f
 Scaled error, 442
 Scatter
 charts, 39
 matrix, 56–57, 58f
 multiple, 55–56, 57f
 Scatterplot, 54–55, 56f
 Seasonal and Trend decomposition using Loess (STL), 405
 Seasonal ARIMA, 426–429, 427f
 forecast using, 428f
 implementation, 427–429
 Seasonal differencing, 422
 Seasonal dummy variables, 415
 Seasonal Extraction in ARIMA Time Series (SEATS), 405
 Seasonal index, 412
 Seasonal Naive method, 407–408
 Seasonality, 401, 404
 regression with, 415–418
 seasonally adjusted time series, 406–407
 SEATS. *See* Seasonal Extraction in ARIMA Time Series (SEATS)
 Second order differencing, 422
 Segmenting customer records, 226b
 Select attribute operator, 363–364
 Self-organizing maps (SOMs), 16, 225–226, 247–259, 248f
 algorithm, 525–526
 implementation, 252–259
 data preparation, 255
 execution and interpretation, 256
 location coordinates, 259
 SOM modeling operator and parameters, 255–256
 working principle, 249–252
 assignment of data objects, 250
 centroid update, 250–252
 initialize centroids, 249
 mapping new data object, 252
 termination, 252
 topology specification, 249
 SEMMA. *See* Sample, Explore, Modify, Model, and Assess (SEMMA)
 Sensitivity, 265, 266t
 Sequential covering technique, 89–90
 Session grouping, 222
 Set Role operator, 178–179, 363–364, 371, 382
 SGD. *See* Stochastic Gradient Descent (SGD)
 Shuffle operator, 172
 Sigmoid curve, logistic regression finding, 188–190
 Sigmoid transformation, 320
 Simple forecasting methods, 407–409
 average method, 408
 moving average smoothing, 408
 Naïve method, 407
 seasonal Naive method, 407–408
 weighted moving average smoothing, 408–409
 Simple matching coefficient (SMC), 106
 Softmax, 323–324, 323f
 SOMs. *See* Self-organizing maps (SOMs)
 Spam email, predicting and filtering, 112b
 Specificity, 265, 266t
 Split Data operator, 94, 363–364
 Split Validation operator, 81, 513
 Spread measure, 45
 Spread of each attribute, 63
 Spreadsheet programs, 398–399
 SSE. *See* Sum of squared errors (SSE)
 Stacking model, 153
 Standard deviation, 45
 Standard machine learning techniques, 429–430
 Stationary data, 421–422
 Statistical methods, outlier detection using, 450–451
 Statistics, 39
 machine learning, and computing combination, 6
 Stemming process, 288–289
 STL. *See* Seasonal and Trend decomposition using Loess (STL)
 Stochastic Gradient Descent (SGD), 370
 Stop word filtering, 286
 Stopping behavior, 485–487
 Stratified sampling, 28–29, 81
 Strict partitioning clusters.
 See Exclusive partitioning clusters
 “Stride”, 327, 328f
 Subject matter expertise, 6
 Sum of squared errors (SSE), 181, 231
 Supervised classification, 399, 453
 Supervised data science, 10
 algorithms, 447

Supervised learning models, 353,
383–389, 527
dataset, 385
implementation, 385
classification process for one
user in system, 385*f*
converting item profile to
classification training set,
387*f*
item profile with class label, 383*t*
personalized decision tree, 384*f*
Supervised model (SVM), 281
Supervised techniques, 10
Support count, 208–209
Support rule, 203
Support vector machines (SVMs), 15,
135–147, 137*f*, 144*f*, 195
algorithm, 523
concept and terminology,
135–138
dataset to demonstrating, 142*t*
implementation, 141–147
works, 138–140
Surface plot, 503
SVM. *See* Supervised model (SVM)
SVMs. *See* Support vector machines
(SVMs)
Synonymy, 365–366
Systematic components, 397–398

T

Target variable. *See* Output variable
TDM. *See* Term document matrix
(TDM)
Technical integration, 34–35
Tensorflow Playground (TF
Playground), 311, 312*f*
Term document matrix (TDM),
285–286, 287*t*
Term filtering, 286–288
Term frequency–inverse document
frequency (TF–IDF),
283–285
Test dataset evaluation, 33*t*
Testing datasets, 30, 30*t*, 31*f*
Text analytics, 281
Text mining, 9, 11, 17, 281–283,
380–382
high-level process, 283*f*
implementation, 290–304
keyword clustering, 290–294
predicting gender of blog
authors, 294–304

sequence of preprocessing steps,
289*t*
working principle, 283–290
term frequency–inverse
document frequency,
283–285
terminology, 285–290
TF Playground. *See* Tensorflow
Playground (TF Playground)
TF–IDF. *See* Term frequency–inverse
document frequency
(TF–IDF)
Three-parameter exponential
smoothing, 412
Time periods, 407
Time series, 395
analysis, 395, 400*f*
data-driven approaches, 407–413
decomposition, 397–398,
400–407, 402*f*
classical, 403–404
forecasting using decomposed
data, 406–407
implementation, 404–407
process for, 405*f*
forecasting, 11, 13*t*, 395, 528
demand of product, 399*b*
taxonomy, 397–399, 398*f*
implementation, 412–413
R script for Holt-Winters’
forecasting, 413
machine learning methods,
429–438
model-driven forecasting methods,
413–429
of monthly antidiabetic drug sales,
396*f*
performance evaluation, 439–443
MAE, 441–442
MAPE, 442
MASE, 442–443
RMSE, 442
validation dataset, 439–443
Timeout, 434–436
Token, 285
Tokenization, 285
Topology, 249
Traditional statistical analysis
approaches, 9
Tragic example, 190–192
Trainable parameters, 324
Training
data, 2–4

datasets, 30, 30*t*, 31*f*
incomplete training set, 117–118
perceptron, 309–310
Transformation, 19–20, 27
data transformation tools,
504–509
sigmoid, 320
unit, 320
Tree-to-rules operator, 96–98
Trend, 401, 404
Truth tables. *See* Confusion matrix
Two-parameter exponential
smoothing, 411–412

U

Unbalance, sub-process, 509
Undirected data science, 10
Unit transformation, 320
Univariate exploration, 44–45
central tendency measure, 44–45
spread measure, 45
Univariate plots, 502
Univariate visualization, 50–53
distribution chart, 52–53, 55*f*
histogram, 50, 51*f*
quartile, 50–51, 53*f*
Universal approximator, 126–127
Unseen dataset, applying model to,
195
Unseen test data, application to,
178–179
Unsupervised anomaly detection,
447
Unsupervised data science, 10
Unsupervised techniques, 11
User interface
GUI, 492–493
terminology, 493–497, 493*f*, 494*f*
attributes and examples, 495*f*
operator for building decision
tree, 496*f*
process automatically translated
to XML document, 497*f*
User *k*-NN recommender process,
363–364
User profile
approach, 352–353, 374
computation, 375–383, 376*f*, 378*t*
User-based collaborative filtering,
355, 360–361
deducing rating from
neighborhood users,
357–358

identifying similar users, 355–357
 User-based neighborhood method,
 351–352, 358
 User-item interaction, 348–349, 370
 User-to-user similarity matrix, 357*t*
 Utility matrix, 347

V

Validation
 dataset, 30, 439–443
 techniques, 263
 Variable, 40
 binomial, 196
 independent, 397
 output, 10
 predictor, 397
 seasonal dummy, 415
 Variance, 45
 Visual model of SOM, 257–258

Visualization, 39, 48. *See also* Data
 visualization
 distribution of each attribute, 63
 relationship between attributes, 64
 style, 257
 tools, 502
 Voronoi partitions, 227, 228*f*
 Vote operator, 152–153, 154*f*

W

Web analytics, 222
 Weight adjustment, 129–130
 Weighted moving average
 smoothing, 408–409
 Windowing, 399, 430–436
 forecast generation in loop,
 434–436
 implementation, 432
 model training, 432, 434
 set up, 432–433

windowing-based machine
 learning, 528
 Wizard-style functionality, 492–493
 Wrapper-type feature selection,
 483–489
 backward elimination, 485–489,
 488*f*, 489*f*
 sample view of Boston housing
 dataset, 486*t*
 wrapper function logic used by
 RapidMiner, 487*f*
 Wrapper-type methods, 467–468

X

X11, 405
 XML code, 496–497

Y

Yet Another Learning Environment
 (YALE), 491