Index

A

absolute errors, 332
academic papers, 148
accessing data. *See* data access
accuracy metric, 321, 332–333, 388
AdaBoost (Adaptive Boosting), 293, 406–407
AdaBoostClassifier class, 406–407
add_edge() function, 141, 196, 198
add_edges_ from() function, 196, 198
add_feature() function, 193
add_node() function, 196, 198
add_nodes_from() function, 196, 198
adjacency matrices, 138–139
adjacency_matrix() function, 139
Advances in Neural Information Processing Systems 17, 424
agglomerative (hierarchical) clustering, 274, 283–287
  defined, 283
  dendrograms, 283, 286
  distance metrics, 284
  image data, 284–285
  linkage methods, 283–284
  visualization, 285–287
aggregating data, 129–130, 285–287
AI (artificial intelligence), 14, 417–418, 426
Al-Kindi, 13
Anaconda. *See also* Anaconda Prompt; Jupyter Notebook
  installing
    on Linux, 39–40
    on Mac OS X, 40–41
    on Windows, 36–38
  overview, 34–35
  screenshots in book, 39
Anaconda Prompt
  (base) environment, 35
  finding, 35
  overview, 35
  Windows 10 Start menu, 36
Analysis Of Variance (ANOVA), 236–237, 342
Android devices, 50
annotate() function, 173
annotations
  adding to charts, 173–174
  defined, 172
anonymizing data
  aggregating data, 130
  removing PII, 422
ANOVA (Analysis Of Variance), 236–237, 342
application programming interfaces (APIs). *See also* Keras API
  Keras, 30
  Python and, 11
  Scikit-learn, 100, 202–203
arrays, 153–155
  matrix multiplication, 155
  matrix vector multiplication, 154–155
  NumPy, 29, 219
  scatterplots, 184–185
  Scikit-learn and data science applications, 203–205
  simple arithmetic on vectors and matrices, 154
  slicing and dicing data, 123–125
  unstructured image data, 96–97
  vectorization, 153
artificial intelligence (AI), 14, 417–418, 426
AskSam, 97
average linkage, 284
axes
  accessing, 163–164
  formatting, 164–165
  handles, 163
  labeling, 172–173
  time series, 186–188

B

bag of words model, 132–138, 212–213, 319
bagging technique, 399–401
BaggingClassifier class, 399–401
BaggingRegressor class, 399
bar charts
  comparisons, 178–179
  distributions, 180
BaseEstimator class, 202
Bayes, Thomas, 317
benchmarking, 213–216
BernoulliNB class, 319–321
bias, 145, 336
  fitting data to models, 329–330
  linear models, 357
  mistruths of, 147
  nonlinear transformations, 352
  Random Forest, 399

COPYRIGHTED MATERIAL

big data, 361–365. *See also* dimensionality (complexity) reduction
defined, 251
K-means clustering, 281–283
linking data science with, 14
Stochastic Gradient Descent optimization, 362–365
when data is considered big, 362
bigcode-pii-dataset, 422
bigml, 426
binning
categorical data, 231–234
defined, 151
heterogeneous data, 210
histograms, 179–180
pandas library, 107
black boxes, 30, 401
boosting, 406–411
AdaBoost, 293, 406–407
Gradient Boosting Machines, 293, 407–410
XGBoost, 314, 410–411
Bower, Jonathan, 420
boxplot() function, 181
boxplots
cross-validation, 339–340
EDA, 234–236
groups, 181–182
outlier detection, 296–297
percentiles, 229
performing t-tests after, 236–237
Breiman, Leo, 399
bunches, defined, 27, 133

C

California Housing dataset
fitting data to models, 334
forward feature selection, 343
GBM hyperparameters, 409
gradient boosting regressor, 408–409
grid search, 345
heterogeneous data, 210
interactions, 355
linear regression, 204, 309–310
loading, 18, 28–29, 47–48
Random Forest regressor, 404
regression trees, 397–398
stratification sampling, 338–339
variable transformations, 353
XGBoost, 411
Canadian Institute for Advanced Research (CIFAR) image datasets, 427–428
Cartopy, 190–194
adding to Google Colab environment, 190
adding to Jupyter Notebook environment, 190–192
plotting geographic data, 192–194
cases, defined, 84, 155
categorical variables (enumerations), 113–117
chi-square statistic, 246
combining levels, 116–117
correlation, 241
creating, 114–115
defined, 113
frequencies, 107, 232–233
renaming levels, 115–116
cells
Google Colab, 59–64
clearing output, 62
code execution within, 51, 54
comments, 61
copying, 62
copying links to, 61
creating code cells, 60–62
creating special cells, 63
creating text cells, 62–63
cutting, 62
deleting, 61
editing, 64
forms, 62
mirroring, 61
moving, 60, 64
running, 65
selecting, 62
Jupyter Notebook
creating, 74
deleting, 74
Markdown cells, 44–45, 74–75
running, 48
styles, 74–75
central tendency
EDA, 227
outliers, 292
centroids, 275–284
Chainer, 385
charts. *See* graphs and charts
ChatGPT, 385
Chebyshev's inequality, 298
Chen, Tianqi, 410
chi2_ contingency() function, 246
chi-square
Cramér's V, 246
tables, 245
univariate approach to variable selection, 342
Chollet, Francois, 386
Chrome browser, 18, 50–52
CIFAR (Canadian Institute for Advanced Research) image datasets, 427–428
classification. *See also* clustering; Support Vector Machines
bag of words model, 132, 138
bagging technique, 400
boosting, 406–408
decision trees, 394–397
defined, 203
fitting data to models, 331–333

Gradient Boosting Machine, 407–408
k-nearest neighbors, 323–325, 345–346
Madelon dataset, 424
Naïve Bayes algorithm, 318, 320–321
neural networks, 386–390
Random Forest, 402–403
Stochastic Gradient Descent, 362
Support Vector Classifier, 371–376
support vector machines, 366–369
variable selection, 342
ClassifierMixin class, 202
classifiers, defined, 132
Cleveland, William S., 12
clustering, 273–289
agglomerative, 274, 283–287
defined, 283
dendrograms, 283, 286
distance metrics, 284
example with image data, 284–285
linkage methods, 283–284
visualization, 285–287
cross-validation, 339
data density approaches, 274
DBScan algorithm, 287–289, 302–303
distinguishing between techniques, 274
fuzzy, 274
K-means, 275–283
big data, 281–283
centroid-based algorithms, 275–277
defined, 275
image data, 277–281
outlier detection, 303
partition-clustering techniques, 274
ClusterMixin class, 202
coastlines() method, 193
code repository for Jupyter Notebook, 43–47
adding notebook content, 44–45
create notebooks, 44
defining folder, 43
exporting notebooks, 45–46
importing notebooks, 46–47
coding styles
effect on performance, 26
Python support for, 10, 17
coef_() method, 205
Colab. *See* Google Colab
collaborative filtering, 268
Columbia University, 12
ColumnTransformer() function, 209–212
COM (Component Object Model) applications, 100
comma-separated value (CSV) files, 89–93
commission, mistruths of, 146
Common Crawl, 429–430
compiling Python, 10, 12
complete linkage, 283
complexity reduction. *See* dimensionality reduction
Component Object Model (COM) applications, 100
concat() function, 103, 126
concept drift, 295
Conductrics, 419
contextualization, 144–149
data preparation, 149
evaluating problems, 145
formulating hypotheses, 148–149
importance of, 144–145
researching solutions, 147–148
contingency tables
chi-square statistic, 247
Cramér's V, 248
creating, 233–234
correlation, 241–245
chi-square for tables, 245
covariance and, 242–244
movie recommendations, 270–271
multivariate, 149
nonparametric, 244–246, 248
outlier detection, 292–293
Pearson's correlation, 241, 243–244, 246–248
redundancy of information, 277
scatterplots, 184–186, 240
Spearman correlation, 244–245, 248
variable selection, 342
cosine distance, 284
CountVectorizer() function, 207, 213–215, 321
covariance, 242–244
COVID-related data, 422, 424
Cramér's V, 246
Cross Validated website, 148
cross_val_score() function, 337, 374
cross-tabulation, 245, 279, 285, 289, 318
cross-validation, 336–341
boosting, 406
defined, 220
grid search, 346–347
interactions, 355–356
k folds, 337–338
multiprocessing, 221
pipeline and, 208
stratification sampling, 338–341
SVC, 374–376
CSV (comma-separated value) files, 89–93
curse of dimensionality, 321
cut() function, 231
Cutler, Adele, 399

D

data access, 83–103
  flat files, 83, 89–94
    reading CSV delimited format, 90–93
    reading from text files, 90
    reading Microsoft Excel and Office files, 93–94
  image data, 86–87, 94–97
  memory streams, 83, 86
  relational databases, 83, 97–98
  sampling data, 88–89
  unstructured files, 94–97
  uploading small amounts of data, 84–85
  web-based data, 83, 100–103
data analysis. *See also* Exploratory Data Analysis
  as core competency, 13
  discretization, 210
  EDA, 15, 223–249
    boxplots, 234–236
    categorical data, 231–234
    confirmatory approach, 224
    correlation, 241–245
    Cramér's V, 246
    defined, 223
    distribution modification, 246–249
    goal of, 223
    graphing distributions, 238–239
    Initial Data Analysis vs., 224–225
    nonlinear transformations, 352
    numeric data, 225–231
    parallel coordinates, 238
    scatterplots, 240–241
    t-tests, 236–237
    visualization, 234–241
  effect of algorithm on performance, 26
  feature creation, 149
  pandas library, 29
  prototyping and experimentation process, 25
  Scikit-learn library, 30
  XGBoost, 31
data capture, 13, 418
data density clustering approaches, 274
data lineage, 428–429
data maps, 110–111
data munging. *See* data wrangling
data partitioning, 335–336
data plans, 111–112
data preparation, 15, 107, 149
data presentation. *See also* visualization
  as core competency, 13
  display frameworks, 23
  prototyping and experimentation process, 25
data science, 1, 9, 12, 143–155
  arrays, 153–155
    matrix multiplication, 155
    matrix vector multiplication, 154–155
    simple arithmetic on vectors and matrices, 154
    vectorization, 153
  challenging datasets, 421–430
  contextualization, 144–149
    data preparation, 149
    evaluating problems, 145
    formulating hypotheses, 148–149
    importance of, 144–145
    researching solutions, 147–148
  core competencies of data scientists, 13–14
  emergence of, 12–13
  feature creation (engineering), 149–152
    binning, 151
    combining variables, 150–151
    defining, 149–150
    discretization, 151
    indicator variables, 151–152
    transforming distributions, 152
  libraries, 23
  linking with big data and AI, 14
  pipeline for, 15
  Python libraries and ecosystem, 28–32
  Python setup, 33–48
  Python's role in, 16–17, 22–23
    data access, 23
    data display, 23
    library support, 23
    multipurpose, simple, and efficient language, 17
    parallel processing, 23
    shifting profile of data scientists, 16
  resource collections, 415–420
    Conductrics, 419
    Data Science 101, 418
    Data Science Central, 417–418
    Jonathan Bower, 420
    KDnuggets, 416
    Oracle AI & Data Science Blog, 417
    Quora, 416–417
    Reddit, 416
    Springboard, 420
    Udacity, 418–419
Data Science 101, 418
Data Science Central, 417–418
data scientists
  core competencies of, 13–14
  preparing data, 107
  shifting profile of, 16

data validation, 108–112
  data maps, 110–111
  data plans, 111–112
  finding duplicates, 108–109
  purpose of, 108
  removing duplicates, 110
data wrangling (data munging), 201–222
  clustering, 273–289
  defined, 201
  dimensionality reduction, 251–271
  Exploratory Data Analysis, 223–249
  outlier detection, 291–304
  performance and speed issues, 212–219
  Scikit-learn, 201–212
    applications for data science, 203–206
    classes, 202–203
    documentation, 201
    multicore parallelism, 220–222
    transformative functions, 207–212
Database Management Systems (DBMSes), 97–99
Data.gov, 422
datasets. *See also names of specific datasets*
  challenging, 421–430
  downloading, 42, 44
  functions for importing, 47–48
  high-dimensional sparse, 134
  NoSQL Databases, 99
  providing access to in local environment, 28
  size of and effect on performance, 26
  toy datasets, 84
date- and time-related data, 117–119
  formatting date and time values, 118
  issues with, 117–118
  time transformations, 118–119
DBMSes (Database Management Systems), 97–99
DBScan algorithm, 287–289, 302
decision trees, 392–398
  branches, 394
  classification trees, 395–397
  leaves, 394
  overfitting, 395
  reading nodes of, 394
  regression trees, 397–398
  visualization, 393–394, 397
DecisionTreeClassifier class, 394, 396
DecisionTreeRegressor class, 394, 397
dendrograms, 283, 286
describe() method, 111–112, 226
desired solutions, 145
diabetes dataset, 296
dicing data, 124–125
DiGraph() constructor, 198
dimensionality (complexity) reduction, 251–271
  curse of dimensionality, 321
  factor analysis, 256–258
  principal component analysis, 256–265
    components, 258
    facial recognition, 262–265
    reducing dimensionality, 259
    SVD vs., 258
    topic extraction, 266–268
    t-SNE, 259–261
    unique vs. shared variance, 256
  Singular Value Decomposition, 252–256
    entertainment preferences and recommendations, 256
    formula of, 252
    measuring the invisible, 255–256
    movie recommendations, 268–271
    reducing dimensionality, 253–255
    text hinting at ideas and meaningful categories, 255
dir() function, 27, 78
directed graphs, 195, 197–198
discretization, 151
  defined, 210
  heterogeneous data, 210–211
distributions
  boxplots, 234–236
  defined, 152
  graphing, 238–239
  histograms, 179–181
  modifying, 246–249
  transforming, 152
Domingos, Pedro, 150
dot() function, 254
draw_networkx() function, 198
drop() method, 127
dropna() method, 121
dtreeviz package, 393, 397
dummy variables (indicator variables), 151–152
duplicate data
  finding duplicates, 108–109
  removing duplicates, 110
duplicated() method, 109
duplicates() method, 110
DurusWorks, 23

E

EDA. *See* Exploratory Data Analysis
edges, defined, 138, 195
ElasticNet class, 361–362
ELT (Extract, Load, Transform), 14

Encyclopedia Titanica, 392
ensembles, 221, 391–411
  boosting, 406–411
    AdaBoost, 406–407
    Gradient Boosting Machine classifier, 407–408
    Gradient Boosting Machine hyperparameters, 409–410
    Gradient Boosting Machine regressor, 408–409
    XGBoost, 410–411
  decision trees, 392–398
    branches, 394
    classification trees, 395–397
    leaves, 394
    overfitting, 395
    reading nodes of, 394
    regression trees, 397–398
    visualization, 393–394
  defined, 391
  ensembles of models, 399
  Random Forest, 399–405
    classifier, 402–403
    machine learning, 399–402
    optimizing, 404–405
    regressor, 404
    steps for, 399
entertainment preferences and recommendations, 256, 268–271
enumerate() method, 88
enumerations. *See* categorical variables
error functions, 294, 331, 362
ETL (Extract, Transform, Load), 14, 424
Euclidean distance, 275–277, 284, 324
evaluation metrics, 331
Excel files. *See* Microsoft Excel files
Exploratory Data Analysis (EDA), 15, 223–249
  boxplots, 234–236
  categorical data, 231–234
  confirmatory approach, 224
  correlation, 241–245
    chi-square for tables, 245
    covariance and, 242–244
    nonparametric, 244–245
  Cramér's V, 246
  defined, 223
  distribution modification, 246–249
    converting into uniform or normal distributions, 248–249
    using different statistical distributions, 247
    Z-score standardization, 247–248
  goal of, 223
  graphing distributions, 238–239
  Initial Data Analysis vs., 224–225
  nonlinear transformations, 352
  numeric data, 225–231
    central tendency, 227
    percentiles, 228–229
    skewness and kurtosis, 229–231
    standard deviation, 228
  parallel coordinates, 238
  scatterplots, 240–241
  t-tests, 236–237
  visualization, 234–241
    boxplots, 234–236
    graphing distributions, 238–239
    parallel coordinates, 238
    scatterplots, 240–241
    t-tests, 236–237
Extract, Load, Transform (ELT), 14
Extract, Transform, Load (ETL), 14, 424
eXtreme Gradient Boosting (XGBoost) library, 31, 410–411

F

facial recognition, 262–265
factor analysis, 256–258
  hidden factors, 257–258
  psychometric model, 257
  unique vs. shared variance, 256
FactorAnalysis class, 258
feature creation (engineering), 149–152
  binning, 151
  combining variables, 150–151
  defining, 149–150
  discretization, 151
  indicator variables, 151–152
  nonlinear transformations, 352–354
  transforming distributions, 152
FeatureUnion() function, 209–210
fillna() method, 121
Firefox browser, 51, 53
fit() method, 122, 203–206, 278, 346, 363, 373–374
fit_transform() method, 134
fitting data to models (model training), 205, 208–209, 309–310, 328–336
  bias, 329–330
  boosting, 406
  decision trees, 395
  defined, 328
  evaluation metrics, 331
  optimization process, 331

overfitting, 311, 348–349, 364–365, 370, 376, 388, 390, 392, 395–396, 401, 406, 409, 424
Python workflow, 18–19
strategy for picking models, 330–333
training vs. test sets, 334–336
underfitting, 370
variance, 329–330
flat files, 83, 89–94
reading CSV files, 90–93
reading from text files, 90
reading Microsoft Excel and Office files, 93–94
ForecastWatch, 22–23
forward and backward approach, 341, 343–344
frame-of-reference mistruths, 147
functional coding, 17
fuzzy clustering, 274

G

Gaussian distribution, 230, 297–298
GaussianNB class, 319
GBMs. *See* gradient boosting
General Data Protection Regulation (GDPR), 422
Generalized Linear Models (GLMs), 313
geographical data, 190–195
Basemap, 194–195
Cartopy, 191–194
German Credit Data dataset, 47, 400–402, 405–407, 410
Gershoff, Matt, 419
getroot() method, 102
GitHub, 50, 52, 420
opening notebooks from, 55–56
saving notebooks to, 57–58
saving notebooks with GitHub gists, 58–59
GLMs (Generalized Linear Models), 313
Google Chrome browser, 18, 50–52
Google Colab, 2, 24, 33, 49–69, 386
Android devices, 50
browsers, 51
Cartopy, 190
cells, 59–64
clearing output, 62
code execution within, 51, 54
comments, 61
copying, 62
copying links to, 61
creating code cells, 60–62
creating special cells, 63
creating text cells, 62–63
cutting, 62
deleting, 61
editing, 64
forms, 62
mirroring, 61
moving, 60, 64
running, 65
selecting, 62
code execution
within cells, 51, 54
checking, 67
overview, 65
defined, 49
functions of, 50
GitHub repository, 50
hardware acceleration, 64–65
help resources, 68–69
Jupyter Notebook vs., 50–52
local runtime support, 53
notebooks, 53–59
creating, 54
displaying table of contents, 66
downloading, 59
getting notebook information, 66–67
opening, 54–55
saving, 56–59
sharing, 67–68
storage locations, 55–59
viewing, 66
Google Drive
opening notebooks from, 55
saving notebooks to, 56–57
Google Scholar, 148
Gorman, Kristen, 225
gradient boosting
Gradient Boosting Machines, 293
classifier, 407–408
hyperparameters, 409–410
regressor, 408–409
XGBoost, 31, 410–411
gradient descent, 309, 362–365
GradientBoostingClassifier class, 407–408
GradientBoostingRegressor class, 408–409
graphics. *See* graphs and charts; multimedia and graphics
graphs and charts, 138–141, 159–175
adjacency matrices, 138–139
annotations, 172–174
axes, 163–165
bar charts, 178–179
boxplots, 181–182
directed graphs, 197–198
distributions, 238–239
geographical data, 190–195
graphs, defined, 195
grids, 165–166
hairballs, 139
histograms, 179–181
labels, 171–173
legends, 172, 174–175

graphs and charts *(continued)*
lines
colors, 168–169
drawing multiple, 161–162
line styles, 167–168
markers, 169–171
links (edges), 138
NetworkX, 31–32, 139–141
nodes, 138
plots, 160–162
saving work to disk, 162–163
scatterplots, 182–186
time series, 186–190
undirected graphs, 195–197
grid() function, 166
grid search, 220, 345–348
gridlines() method, 194
GridSearch class, 346
GridSearchCV() function, 350, 375
ground truth, 278–279, 285, 289
groupby() function, 111, 130
Grover, Prince, 393

H

hairballs, 139
HalvingGridSearchCV() function, 348–349
handwriting recognition, 221, 260–261, 277–289, 366, 371–372, 376, 387, 426–428
handwritten number dataset, 221, 260, 277–278, 280, 282, 288, 371–372, 387, 426–427
heterogeneous data, 210–212
hierarchical clustering. *See* agglomerative clustering
high-dimensional sparse datasets, 134
histograms, 179–181, 238–239, 241
Hofmann, Hans, 400
hurdle values, 299
hyperlinkgraph dataset, 429–430
hyperparameters, 204, 220, 344–350, 358
defined, 344
Gradient Boosting Machine, 409–410
grid search, 345–348
randomized search, 348–350
hypotheses, 148–149, 205

I

IDA (Initial Data Analysis), 224–225
IDEs (Integrated Development Environments), 24, 39. *See also* Google Colab; Jupyter Notebook
imperative coding, 17
imshow() function, 87, 95, 371
indentation, 24
indicator variables (dummy variables), 151–152
inductive learning, 327–350
cross-validation, 336–341
fitting models, 328
bias, 329–330
strategy for picking models, 330–333
training vs. test sets, 334–336
variance, 329–330
hyperparameters, 344–350
selecting variables, 338–341
forward and backward approach, 341, 343–344
univariate approach, 341–343
inertia
defined, 280
rate of change in, 280–282
information redundancy, 149, 242
Information Retrieval (IR), 133
Initial Data Analysis (IDA), 224–225
insights, 15, 27
Integrated Development Environments (IDEs), 24, 39. *See also* Google Colab; Jupyter Notebook
International Council for Science, 12
Internet World Stats, 1
interpreted mode, 10–12
interquartile range (IQR), 229, 235, 297, 299
IPython Notebook files, 46, 50
IR (Information Retrieval), 133
isin() method, 117
isnull() method, 115, 120
Isolation Forests, 303–304

J

JavaScript Object Notation (JSON), 103
join() method, 126
jQuery, 100
Jupyter Notebook, 2, 10, 18, 23–24, 33, 42–43, 73–82. *See also* Google Colab
Cartopy, 190–191
cells
creating, 74
deleting, 74
Markdown cells, 44–45, 74–75
running, 48
styles, 74–75
defining new folders, 43
GitHub repository, 50
Google Colab vs., 50–52
help resources, 75–76
identifying version of, 34–35
literate programming, 34
magic functions
axes, 163
backend list, 163
benchmarking, 213–216

displaying status information, 77
embedding graphics, 80, 161
graph display, 163, 165
loading examples from online sites, 80
multiprocessing, 221
obtaining list of, 76–77
percent signs, 77
using with objects, 78
multimedia and graphics, 79–82
notebooks
adding content to, 44–45
creating, 44
exporting, 45–46
importing, 46–47
removing, 46
objects, 77–78
package installation, 190–191
restarting kernel, 79
restoring checkpoints, 79
starting, 42
stopping server, 42–43
styles, 74–75

K

k folds cross-validation, 337–338
Kaggle, 31, 150, 392, 410, 423
KDnuggets, 416
Kelvin, Lord, 145
Keras API, 30–31, 385–388
kernel trick, 368
keys, 27–28
keys() function, 28
KMeans class, 278
K-means clustering, 275–283
big data, 281–283
centroid-based algorithms, 275–277
cross-tabulation, 279
defined, 275
image data, 277–281
inertia, 280–281
memory leak warning, 283
outlier detection, 303
k-nearest neighbors (KNN), 322–325, 334
choosing k parameter, 324–325
hyperparameters, 345
predicting after observing neighbors, 323–324
Knuth, Donald, 34
kurtosis, 229–231

L

labels, 171–173
Lasso (L1) regularization, 358–361
last_valid_index() method, 126
Latent Semantic Indexing (LSI), 256
learning from data, 15, 307–411
algorithms, 307–325
big data, 361–365
ensembles, 391–411
inductive learning, 327–350
neural networks, 385–390
nonlinear transformations, 352–357
regularization, 357–361
support vector machines, 365–384
legend() function, 174–175
legends, 172, 174–175
libraries. *See names of specific libraries*
linear regression, 307–312, 331–332
family of linear models, 308–309
formula type, 308, 329
heterogeneous data, 211
instantiation, 204–206
L2 (Ridge) regularization, 358
limitations and problems, 312
model training, 18–19
multiple variables, 309–312
univariate approach to variable selection, 341
LinearRegression class, 205–206, 355
LinearSVC class, 369, 380–384
lines, in graphs and charts
colors, 168–169
drawing multiple, 161–162
line styles, 167–168
markers, 169–171
links (edges), defined, 138
Linux, installing Anaconda on, 39–40
listwise deletion, 298
literate programming, 11, 34
loading data, effect on performance, 26
local storage, opening notebooks from, 56
logistic regression, 313–317
applying, 313–315
multiclass problems, 315–317
logspace() function, 375
LSI (Latent Semantic Indexing), 256

M

Mac OS X, installing Anaconda on, 40–41
machine capability, effect on performance, 26
machine code, 21
Machine Learning Security Principles (Mueller), 422
Macready, William, 308
Madelon dataset, 424

magic functions, 76–78
axes, 163
backend list, 163
benchmarking, 213–216
displaying status information, 77
embedding graphics, 80, 161
graph display, 163, 165
loading examples from online sites, 80
multiprocessing, 221
obtaining list of, 76–77
percent signs, 77
using with objects, 78
main effects models, 354
make_classification() function, 381–383, 424
make_regression() function, 378
Manhattan distance, 284, 324
manifold learning (nonlinear dimensionality reduction), 259–261
Manuscript on Deciphering Cryptographic Messages (Al-Kindi), 13
markers, adding to graphs, 169–171
MATLAB, 25, 159–160
Matplotlib library, 2, 25, 87, 159–175
agglomerative clustering, 286
annotations, 172–174
axes, 163–165
bar charts, 178–179
boxplots, 181–182
geographical data, 190–195
graphs, 195–198
grids, 165–166
histograms, 179–181
labels, 171–173
legends, 172, 174–175
lines
colors, 168–169
drawing multiple, 161–162
line styles, 167–168
markers, 169–171
MATLAB vs., 159–160
overview, 31
plots, 160–162
saving work to disk, 162–163
scatterplots, 182–186
time series, 186–190
matrices
adjacency matrices, 138–139
matrix multiplication, 155
matrix vector multiplication, 154–155
simple arithmetic on, 154
Singular Value Decomposition, 252–256
maximum likelihood estimation (MLE) method, 259
mebibytes (MiBs), 217
memory_profiler package, 217–218
microservices, 100
Microsoft Academic, 148
Microsoft Excel files
CSV files, 91–92
dates, 117
flat files, 89
reading from, 93–94
Microsoft Office files, reading from, 93–94
Microsoft Windows
Anaconda Prompt and Windows 10, 36
installing Anaconda, 36–38
memory leak warning, 283
MiniBatchKMeans class, 281–282
MinMaxScaler() function, 207, 362
mistruths, 146–147
of bias, 147
of commission, 146
of frame of reference, 147
of omission, 146
of perspective, 146–147
Mixed National Institute of Standards and Technology (MNIST), 426–427
MLE (maximum likelihood estimation) method, 259
model training. *See* fitting data to models
MongoClient class, 99
MongoDB, 99
MovieLens dataset, 47, 268–269, 425
multicore parallelism, 220–222
demonstrating, 221–222
performing, 220–221
MultiDiGraph() constructor, 198
MultiGraph() constructor, 198
multilabel prediction, 220–221
multimedia and graphics. *See also* graphs and charts
facial recognition, 262–265
image data, 86–87, 94–97
Jupyter Notebook, 79–82
embedding plots and images, 80
loading examples from online sites, 80
obtaining online graphics and multimedia, 80–82
MultinomialNB class, 319–321
multiple-data-source problems, 423–424
multivariate approach
data analysis, 234
outlier detection, 300–304
multivariate correlation, 149
Mushroom dataset, 393, 395–398
MXNet, 385
mysqlclient, 23
MySQLdb library, 23

N

Naïve Bayes algorithm, 317–322, 355
  text classification prediction, 320–322
  uses for, 318–319
National Institute of Standards and Technology (NIST), 152, 426
Natural Language Processing (NLP), 133
ndarray structure, 96–97, 153, 267, 296
n-dimensional array manipulation, 29
neighborhoods
  defined, 288
  k-nearest neighbors, 322–325
NetworkX library
  directed graphs, 198
  graph data, 139–141
  overview, 31–32
  undirected graphs, 196
neural networks, 329, 351, 385–390
  classification problems, 386–390
  deep learning, 385
  neurons, 385–386
  numeric values, 386
  overview, 385–386
  regression problems, 386–387
n-grams, 134–135, 138, 320
NIST (National Institute of Standards and Technology), 152, 426
NLP (Natural Language Processing), 133
NMF (Non-Negative Matrix Factorization), 266–268
"no free lunch" theorem, 308, 330
nodes
  adding to graphs, 196
  adjacency matrices, 138–139
  defined, 138, 195
nonlinear dimensionality reduction (manifold learning), 259–261
nonlinear transformations, 352–357
  interactions between variables, 354–357
  variable transformations, 353–354
Non-Negative Matrix Factorization (NMF), 266–268
nonparametric correlation, 244–246, 248
NoSQL databases, 99
notebooks
  Google Colab, 53–59
    creating, 54
    displaying table of contents, 66
    downloading, 59
    getting notebook information, 66–67
    opening, 54–55
    saving, 56–59
    sharing, 67–68
    storage locations, 55–59
    viewing, 66
  Jupyter Notebook, 44–47
    adding content to, 44–45
    creating, 44
    exporting, 45–46
    importing, 46–47
    removing, 46
novelty detection, 294–295
np.max() method, 153
np.min() method, 153
NumPy library
  aggregating data, 128
  arrays, 153–155, 219
  covariance and correlation, 243
  dimensionality reduction, 253–254
  documentation, 148
  identifying version of, 226
  installing, 216
  label count, 372
  ndarray structure, 96–97, 153, 267, 296
  overview, 29
  pandas vs., 106–107
  performance and speed issues, 219
  random generator, 292
  transformers returning features as, 210
  trendlines, 185

O

objectify.parse() method, 101
object-oriented programming, 17
objects, 76–78
  help resources, 77–78
  properties and methods associated with, 78
  using magic functions with, 76–78
Olivetti faces dataset, 262, 428
omission, mistruths of, 146
one-hot encoding, 212, 388
OneHotEncoder() function, 207
OneVsOneClassifier class, 316–317
OneVsRestClassifier class, 316–317
online resources
  Advances in Neural Information Processing Systems, 424
  Anaconda, 36, 39–40
  AskSam, 97
  author's website, 411
  bagging technique, 399
  Basemap, 194
  bigml, 426
  Cartopy, 190, 193–194
  Chainer, 385
  ChatGPT, 385

online resources *(continued)*
cheat sheet (companion to book), 5
Common Crawl, 429–430
Conductrics, 419
COVID data, 424
Cross Validated, 148
data lineage, 428–429
Data Science 101, 418
Data Science Central, 417–418
Data.gov, 422
datasets
bigcode-pii-dataset, 422
California Housing, 18, 47
CIFAR, 427
finding, 426
German Credit Data, 47, 400
handwritten numbers, 426–427
Madelon, 424
MovieLens, 47, 425
mushrooms, 395
Olivetti faces, 262
Palmer Penguins, 47
Single Family Purchase Loan Data, 422
Titanic tragedy, 392, 394
20 Newsgroups, 132
date- and time-related data, 118
distributions, 152
dtreeviz package, 393, 397
DurusWorks, 23
example code (companion to book), 5
Extract, Transform, Load, 424
feature engineering and ML, 150
Feature Extraction, Foundations and Applications, 424
flat files, 90, 92, 94
frame-of-reference mistruths, 147
GDPR, 422
GitHub, 57, 420
Google Colab, 49–50, 53, 64, 68
Google Drive, 56
Google Scholar, 148
Imputer parameters, 121
Jonathan Bower, 420
jQuery, 100
JSON, 103
Jupyter Notebook, 82, 191, 216
Kaggle, 150, 392, 410
KDnuggets, 416
Keras API, 30, 385–386
kernel trick, 368
literate programming techniques, 11, 34
MATLAB, 25
Matplotlib library, 31, 159, 160, 175, 179
measurement and improvement, 145
mebibytes, 217
Microsoft Academic, 148
Microsoft Excel and dates, 117
MNIST, 426–427
MongoDB, 99
multiple-data-source resources, 423–424
MXNet, 385
mysqlclient, 23
MySQLdb library, 23
NetworkX library, 32, 139–140
NIST, 152
“no free lunch” theorem, 308
NumPy library, 29
one-hot encoding, 388
Oracle AI & Data Science Blog, 417
overfit strategies, 424
pandas library, 29, 106, 112
Personally Identifiable Information resources, 422
PyMongo library, 99
Python, 3, 10–12, 22, 135, 148
PyTorch, 385, 423
Quixote, 23
Quora, 148, 416
Reddit, 416
relational databases, 98
Scikit-image library, 94
Scikit-learn library, 30, 148, 201, 207–208, 235, 287, 313
SciPy library, 29
security resources, 423
Singular Value Decomposition, 252
skewness and kurtosis, 231
Spearman correlation, 245
Springboard, 420
Stack Overflow, 148
TensorFlow ML platform, 30, 386
TF-IDF transformations, 136
Tiny Images, 427
t-SNE, 261
Udacity, 418–419
updates to book, 5
U.S. Department of Labor, 422
World Bank, 422
XGBoost library, 31, 410
XML and JSON alternatives, 103
YAML, 103
open() method, 85
Open Source Data Science (OSDS), 420
OpenML, 47, 395
Oracle AI & Data Science Blog, 417
OrdinalEncoder() function, 207
ordinary least squares, 309
outliers
binning and, 151
defined, 292
detecting, 291–304
concept drift vs., 295

DBScan, 302–303
Gaussian distribution, 297–298
Isolation Forests, 303–304
multivariate approach, 300–304
novelty detection vs., 294–295
principal component analysis, 300–302
univariate approach, 295–299
effects of, 292–294
remediating, 294, 298–299
overfitting
bagging technique, 401
boosting, 406, 409
decision trees, 392, 395–396
defined, 311
grid search, 348–349
neural networks, 388, 390
nonlinear kernels, 376
Stochastic Gradient Descent, 364
strategies for, 424
support vector machines, 365, 370

P

Palmer Penguins dataset, 47, 225–248, 257–259, 313–316, 424
pandas library, 89–94, 102, 106–117
categorical variables, 113–117
documentation, 148
finding duplicates, 108–109
identifying version of, 113
NumPy vs., 106–107
outlier detection, 295
overview, 29
performance and speed issues, 219
removing duplicates, 110
unstacked data, 112
parallel coordinates, 238
parallel processing, 17, 23, 31, 220–222
Parr, Terence, 393
parse() method, 94, 101
partial_fit() method, 363
partition-clustering techniques, 274
partitioning data, 335–336
PCA. *See* principal component analysis
pd.concat() function, 103
Pearson's correlation, 241, 243–244, 246–248
percentiles, 228–229, 342
Personally Identifiable Information (PII), 422
perspective, mistruths of, 146–147
pip (preferred installer program), 113, 216
Pipeline() function, 208
plot() function, 161, 167, 169–171, 186, 188–189
plots
defining, 160–161
drawing multiple, 161–162
embedding, 80
plt.figure() function, 164
plt.plot() function, 161, 167, 169–171
plt.savefig() function, 161
plt.show() function, 87, 95, 161, 188
poly1d() function, 186
polyfit() function, 185–186
PolynomialFeatures() function, 207, 356
precision metric, 332–333
predict() method, 206
predict_proba() method, 206
preferred installer program (pip), 113, 216
principal component analysis (PCA), 256–265
components, 258
facial recognition, 262–265
outlier detection, 300–302
reducing dimensionality, 259
SVD vs., 258
topic extraction, 266–268
t-SNE, 259–261
unique vs. shared variance, 256
print() function, 27, 96, 117, 262
probability
Bayes' theorem, 317–318
defined, 317
Generalized Linear Models, 313
procedural language, 17
prototyping and experimentation process, 24–25
psychometric model, 257
PyMongo library, 99
Python, 11, 21–32
capabilities and features of, 10–12, 23–28
compiling, 10, 12
factors affecting performance, 26–27
functional coding, 17
general-purpose language, 10–11
imperative coding, 17
indentation, 24
interpreted mode, 10–12
object-oriented programming, 17
procedural language, 17
prototyping and experimentation process, 24–25
simplicity, 23–24
visualization, 27–28
common uses for, 10–11
companies using, 11
documentation, 148

Python *(continued)*
libraries and ecosystem, 17, 28–32
Keras and TensorFlow, 30–31
Matplotlib, 31
NetworkX, 31–32
NumPy, 29
pandas, 29
Scikit-learn, 30
SciPy, 29
XGBoost, 31
multicore parallelism, 220–222
online tutorials, 3
performance and speed issues, 26–27, 212–219
benchmarking, 213–216
memory_profiler package, 217–218
NumPy and pandas, 219
preferred installer program, 216
role of in data science, 16–17, 22–23
setting up, 33–48
versions of, 33
workflow, 17–19
loading data, 18–19
model training, 18–19
viewing results, 19
PyTorch, 385, 423

Q

qcut() function, 231
QuantileTransformer() function, 248
quartiles, defined, 181
Quixote, 23
Quora, 148, 416–417

R

R^2 score, 206, 212, 310–311, 331–332
radial basis function (rbf) kernel, 376–380
Random Forest, 399–405
classifier, 402–403
machine learning, 399–402
optimizing, 404–405
regressor, 404
steps for, 399
random() method, 88
RandomForestClassifier class, 402–405
RandomForestRegressor class, 404
randomized data
boxplots, 181–182
histograms, 179–181
scatterplots, 182–183
randomized search, 348–350
random.shuffle() method, 129
range() function, 178, 196
rbf (radial basis function) kernel, 376–380
read() method, 85
read_csv() method, 92
read_sql() method, 98
read_sql_query() method, 98
read_sql_table() method, 98
read_table() method, 90
recall metric, 332–333
Receiver Operating Characteristic Area Under Curve (ROC AUC), 332–333, 401–403, 407, 411
Reddit, 416
regression, 203
gradient boosting, 408–409
linear regression, 307–312, 331–332
logistic regression, 313–317
neural networks, 386–387
Random Forest, 404
regression trees, 397–398
SGDRegressor, 362–365, 383
support vector machines, 369
Support Vector Regression, 378–380
RegressorMixin class, 202
regularization, 357–361
ElasticNet class, 361
L1 (Lasso) regularization, 358–360
L2 (Ridge) regularization, 358–359
leveraging, 360
relational databases, 83, 97–98
reset_index() method, 126, 128–129
residuals, 309, 311
Ridge (L2) regularization, 358–361
RobustScaler() function, 299
ROC AUC (Receiver Operating Characteristic Area Under Curve), 332–333, 401–403, 407, 411

S

scaling transformations, 374
scatterplots, 177, 182–186
correlations, 184–186
EDA, 240–241
groups, 184
outlier detection through PCA, 301
randomized data, 182–183
Schlimmer, Jeff, 395
Scikit-image library, 94
Scikit-learn library, 2, 132, 201. *See also names of specific datasets*
applications for data science, 203–206
bagging classes, 399
bunches, 27
classes, 202–203
classification problems, 203, 332, 369, 388
clustering datasets, 287
cross-validation, 337
decision tree classes, 394
documentation, 148, 201
feature selection, 207

hyperparameters, 344, 348–349
linear models, 313
multiprocessing, 220–221
Naïve Bayes classes, 319
object-based interfaces, 203
outlier remediation, 299
overview, 30
regression problems, 203, 331, 369, 378
support vector machines, 369
text processing tools, 207
transformative functions, 207–212
variable selection helper functions, 342–343
SciPy library stack
chi-square statistic, 246
documentation, 148
overview, 29
score() method, 206, 310
secure data environment, 423
select_dtypes() method, 235, 244
SelectKBest class, 342
SelectPercentile class, 342
SequentialFeatureSelector class, 343
set_extent() method, 193
set_xlim() function, 164
set_xticks() function, 164
set_ylim() function, 164
set_yticks() function, 164
SGD. *See* Stochastic Gradient Descent optimization
shaping data, 105–130, 131–141
aggregation, 129–130
binning, 107
categorical variables, 113–117
dataframes, 107
date- and time-related data, 117–119
defined, 105–106
graph data, 138–141
missing data, 119–122
NumPy vs. pandas, 106–107
prototyping and experimentation process, 25
slicing data, 123–125
tokenizing, 132–138
transforming data, 125–129
validation, 108–112
workflow, 107
shared variance, 256–257
psychometric model, 257
unique variance vs., 256
SimpleImputer() function, 207
Single Family Purchase Loan Data dataset, 422
Singular Value Decomposition (SVD), 252–256
entertainment preferences and recommendations, 256
formula of, 252
measuring the invisible, 255–256
movie recommendations, 268–271
reducing dimensionality, 253–255
topic extraction, 255
skewness
defined, 229
EDA, 229–231
slicing data, 123–125
columns, 124
dicing data, 124–125
rows, 123–124
sort_values() method, 128
Spearman correlation, 244–245, 248
Springboard, 420
SQL (Structured Query Language), 97–98
sqlalchemy library, 98
stack() method, 112
Stack Overflow, 148
standard deviation, 228–229, 298, 337
StandardScaler() function, 205, 207, 235, 278, 355, 359, 363
statistics, 12–13
StatLib repository, 18
stemming, 133
Stochastic Gradient Descent (SGD) optimization, 362–365
classifier (SGDClassifier), 362–365, 383–384
regressor (SGDRegressor), 362–365, 383
stop words
defined, 133
removing data, 213
text classification prediction, 321
TF-IDF transformations, 136, 266
str() function, 118
StratifiedKFold class, 340
strftime() function, 118
Structured Query Language (SQL), 97–98
supervised learning, defined, 273
Support Vector Classifier (SVC)
classification, 371–376
multicore parallelism, 221–222
support vector machines (SVMs), 365–384
benefits and drawbacks of, 365–366
classification problems, 366–368
creating stochastic solutions, 380–384
hyperplanes, 368
nonlinear kernels, 376–378
regression problems, 378–380
setting up predictive models, 368–370

support vector machines *(continued)*
Support Vector Classifier, 371–376
Support Vector Regression, 378–380
support vectors, defined, 367
SVC. *See* Support Vector Classifier
SVD. *See* Singular Value Decomposition
SVMs. *See* support vector machines

T

TensorFlow ML platform, 30, 385–387
term frequency-inverse document frequency (TF-IDF), 136–138, 267
text files, reading from, 90
TfidfTransformer() method, 137
TfidfVectorizer() function, 207
TfidVectorizer class, 266
Thucydides, 12–13
time series, 186–190
axes, 186–188
trends, 188–190
timedelta() function, 119
timeit() function, 215
Tiny Images, 427
Titanic tragedy datasets, 392–393
to_frame() function, 103
to_sql() method, 98
tokenizing, 132–138
bag of words model, 132–138
CountVectorizer() function, 213
n-grams, 134
TF-IDF transformations, 136–138
train_test_split() function, 335, 373
transform() method, 122, 130, 137, 206
transformative functions, 206–212
chaining estimators into pipelines, 208
composing features, 209–210
heterogeneous data, 210–212
transforming targets, 209
TransformedTargetRegressor() method, 209
TransformerMixin class, 202
transforming data, 125–129
concatenating data, 126–127
distributions, 152
numeric features, 152
removing data, 127–128
shuffling data, 127–128
sorting data, 127
trendlines, 184–186, 188–190
TruncatedSVD class, 270
t-SNE algorithm, 259–261
t-tests, 236–237
Tukey, John, 224
20 Newsgroups dataset, 132–134, 136–137, 266, 320–322

U

U Climb Higher, 418
Udacity, 418–419
underfitting, 370
undirected graphs, 195–197
unique variance, 256
univariate approach
EDA, 234
outlier detection, 295–299
variable selection, 341–343
unsupervised learning
clustering, 273–274
defined, 273
U.S. Department of Labor, 422

V

validating data. *See* data validation
validation_curve() method, 347
value_counts() function, 233
values, defined, 27
Vanderbilt University School of Medicine, 392
VanderPlas, Jake, 216–217
vanilla models, 346
Vapnik, Vladimir, 365–366
variables, 18
categorical, 113–117
combining, 150–151
composing, 210
defined, 84, 155, 205
frequencies, 225, 232–233
indicator, 151–152
linear regression with multiple, 309–312
selecting, 338–341
variable distribution, 225
variance
EDA, 228
fitting data to models, 329–330
Random Forest, 399
vectorization
arrays, 153–154
count vectorization, 207, 213–215, 321
n-grams, 135
TF-IDF transformations, 207, 266
visualization, 15, 159–175, 177–198

agglomerative clustering, 285–287
bar charts, 178–180
boxplots
  cross-validation, 339–340
  EDA, 234–236
  groups, 181–182
  outlier detection, 296–297
  percentiles, 229
  performing t-tests after, 236–237
decision trees, 392–398
EDA, 234–241
  boxplots, 234–236
  graphing distributions, 238–239
  parallel coordinates, 238
  scatterplots, 240–241
  t-tests, 236–237
geographical data, 190–195
  Basemap, 194–195
  Cartopy, 191–194
graphs and charts, 138–141, 159–175, 195–198
  adjacency matrices, 138–139
  annotations, 172–174
  axes, 163–165
  directed graphs, 197–198
  distributions, 238–239
  graphs, defined, 195
  grids, 165–166
  hairballs, 139
  labels, 171–173
  legends, 172, 174–175
  lines, 161–162, 167–171
  links (edges), 138
  NetworkX, 31–32, 139–141
  nodes, 138
  saving work to disk, 162–163
  undirected graphs, 195–197
histograms, 179–181, 238–239, 241
playing with data, 27–28
plots
  defining, 160–161
  drawing multiple, 161–162
  embedding, 80
saving work to disk, 162–163
scatterplots, 177, 182–186
  correlations, 184–186
  EDA, 240–241
  groups, 184
  outlier detection through PCA, 301
  randomized data, 182–183
time series, 186–190
  axes, 186–188
  trends, 188–190

W

ward linkage, 283–284
web services, defined, 100
whiskers, defined, 181
Windows
  Anaconda Prompt and Windows 10, 36
  installing Anaconda, 36–38
  memory leak warning, 283
winsorizing, 299
Wolpert, David, 308, 330
World Bank, 422

X

XGBoost (eXtreme Gradient Boosting) library, 31, 410–411
xlabel() function, 172
XML, 100–101, 103

Y

YAML, 103
ylabel() function, 172

Z

Z-score standardization, 247–248