

INDEX

A

- abline() function, 169
- abs() (absolute value) function, 8
- acronyms, 221
- activation functions, 163, 186–187, 189–190, 193
- acts argument, 189
- AdaBoost, 95, 101–102
- adjusted R^2 value, 138
- African Soil Property dataset, 99–100, 159–161
- aggregation, 97
- AG News dataset, 219–220
- Airbnb data
 - dimension reduction, 131–133
 - holdout sets, difficulties with, 135
 - linear model, 128–130
 - regularization, 157–159
- All Possible Subsets Method, 66
- All vs. All (AVA) method, 141, 168
- alpha argument, 92
- Anderson Iris dataset
 - kernel, applying, 178–181
 - optimizing criterion, 174–176
 - overview, 172–174
 - soft margin, 181–182
 - support vectors, 177
- Area Under Curve (AUC), 46–48
- as.matrix() function, 226
- Augmentation() function, 208
- autoregressive model, 216

B

- bagging
 - bias vs. variance, 96
 - cross-validation, 109
 - overview, 96
 - qrRF() function, 97–98
 - random forests, 97
 - remote-sensing soil analysis, 99–100
 - vertebrae data, 98–99
- bag-of-words model, 217
- baseball player data
 - blending linear model with other methods, 148–149
 - k-NN and categorical features, 17
 - linear model, 124–127
 - overfitting, 53
 - overview, 16–17
 - scaling, 18–19
- Berra, Yogi, 6
- beta notation, 128
- bias
 - bagging and boosting, 96
 - in boosting, 106
 - in linear model, 142–143
 - in logistic model, 142–143
 - in neural networks, 195
 - overview, 52
 - in time series, 216
- Bias-Variance Trade-off
 - analogy to election polls, 15
 - consolidation, 65
 - convolutional models, 204
 - in decision trees, 91–92
 - in neural networks, 195
 - overview, 15, 52
 - PCs and, 73–74
 - predicting bike ridership, 15–16
 - shrinkage, 154
- bigmemory package, 77
- bike sharing dataset
 - Bias-Variance Trade-off, 15–16
 - dirty data, 26–27
 - linear models, 143
 - loading data, 5
 - long-term time trends, 25–26

- bike sharing dataset (*continued*)
 - missing data, 26–27
 - overview, 4
 - polynomial models, 144–147
 - predicting ridership, 4, 7–9
 - Bonferroni–Dunn intervals, 115–116
 - boosting
 - AdaBoost, 101–102
 - bias vs. variance, 96, 106
 - call network monitoring, 103–105
 - computational speed, 106
 - gradient boosting, 102
 - hyperparameters in, 106
 - learning rate, 106–109
 - overview, 7, 100
 - Vertebral Column Dataset, 105–106
 - bootstrap, 96–97
 - bounded variables, 19
 - Box, George, 136
 - Breiman, Leo, 81, 95, 97, 109
 - broken clock problem, 193
- C**
- call network monitoring, 103–105
 - Call Test Measurements for Mobile Network Monitoring and Optimization dataset, 103–104
 - CART (classification and regression trees), 81
 - categorical variables, 9–10, 17
 - centering, 18
 - channel, 203
 - CIs (confidence intervals), 114–115, 133–134
 - classification and regression trees (CART), 81
 - classification applications. *See also* generalized linear models
 - Area Under Curve, 46–48
 - confusion matrix, 41
 - error rates, 39–41
 - k-NN in, 36–37
 - overview, 10, 31–32
 - Receiver Operating Characteristic, 46–48
 - regression function in, 32–33
 - Telco Customer Churn dataset, 33–37
 - unbalanced data, 41–46
 - Vertebral Column dataset, 38–39
 - CNNs (convolutional neural networks), 200–202, 207, 209. *See also* convolutional models
 - coef() function, 168
 - coefficients, 67, 202
 - combinations of factor levels, number of, 86
 - complete.cases() function, 35
 - computational issues in large datasets, 61–62
 - computational speed in boosting, 106
 - conditional mean, 14
 - confidence intervals (CIs), 114–115, 133–134
 - confusion matrix, 41, 89
 - consolidation, 65, 86
 - conv2d parameter, 202–203
 - conv argument, 189
 - convergence, 108, 183, 193
 - conversions, factor, 226–227
 - convex hulls, 174–175
 - convolutional models
 - convolution operation, 204–205
 - dropout, 206
 - image tiling, 203–204
 - overview, 201–202
 - pooling operation, 205–206
 - recognition of locality, 201–202
 - shape evolution, 206–207
 - translation invariance, 208
 - convolutional neural networks (CNNs), 200–202, 207, 209. *See also* convolutional models
 - Covertypes dataset, 88–90
 - credit card fraud, 44–45
 - cross-validation
 - in decision trees, 91
 - K-fold, 55–56
 - motivation, 22–23
 - overview, 21, 55
 - programmer and engineer data, 56–58
 - random forests, 109

- replicMeans() function, 56
- triple, 58
- ctree() function, 82–84, 101
- cumsum() (cumulative sums) function, 70–71
- Curse of Dimensionality (CoD), 74–75
- cv.glmnet() function, 155

D

- data argument, 97, 112
- data augmentation, 208
- data cleaning, 128–129
- dataset size, overfitting, 53–54
- ?day1 command, 5
- decision trees (DTs). *See also* boosting
 - bagging, 96–100
 - basics of, 81–82
 - Forest Cover Data, 88–90
 - hyperparameters in qeDT() function, 91–92
 - New York City Taxi data, 85–88
 - number of combinations of factor levels, 86
 - overview, 81
 - plot() function, 83–85
 - qeDT() function, 82–83
 - splitting hyperparameters, 90–91
 - tree-based analysis, 86–88
- deep learning, 185
- dense layers, 202
- depth of neural network, 195
- dimension reduction

- All Possible Subsets Method, 66
- computational issues in large datasets, 61–62
- consolidation and embedding, 65
- Curse of Dimensionality, 74–75
- FOCI, 75–77
- general discussion, 62–63
- going further computationally, 77
- hyperparameters in, 70–71
- linear model, 130–135
- Million Song dataset, 63–64
- need for, 64–65
- New York City Taxi data, 86–88
- overview, 61
- PCA, 66–69

- PCs and Bias-Variance Trade-off, 73–74

- qePCA() wrapper, 71–72
- UMAP method, 77

- dirty data, 26–27
- document-term matrix (DTM), 217
- dot products, 171–172
- doughnut-shaped data, 178–180
- downsampling, 42–44
- dropout, 191, 206
- dummy variables, 9–10, 17, 226
- Dunn, Olive Jean, 115

E

- early stopping, 193
- elastic net, 155
- election polls, analogy to, 15
- embedding, 65, 86
- epochs, 188
- error rates, 39–41
- exploding gradient problem, 190

F

- factor conversions, 226–227
- factor data read as non-factor, 34–35
- factor levels, number of combinations of, 86
- factorToDummies() function, 17, 226
- Fall Detection data, 141–142, 191–192
- false positive rate (FPR), 46
- Fashion MNIST data, 200–201
- Feature Ordering by Conditional Independence (FOCI), 75–77, 166

- features

- intuition regarding number of, 53
- notation, 10–11
- overview, 4

- fineTuning() function, 25, 148, 231

- fold, 55

- Forest Cover dataset

- decision trees, 88–90
- motivation, 182
- support vector machines, 166–170

- Fraud: A Guide to Its Prevention, Detection, and Investigation*, 44–45

- fread() function, 88–89

Friedman, Jerry, 81
fully connected layers, 202

G

`gbm()` function, 106, 109
`gbm.perf()` function, 104–105
generalized linear models
 bias and variance in, 142–143
 Fall Detection Data, 141–142
 `glm()` and `qelogit()` functions, 139
 multiclass case, 140–141
 overview, 138
 Telco Churn Data, 139–140
generic functions, 13, 83
geometric view of soft margin, 181
`getPoly()` function, 183
`glm()` function, 139–141
global minimum, 107
going further computationally, 77
“Goldilocks” value, xxii, 16, 54, 59
gradient, 108
gradient boosting, xxiii, 102
grid searching
 calling `qeFT()`, 112–113
 overview, 112
 phoneme dataset, 117–119
 programmer and engineer data,
 113–117

H

hard margins, 177
`head()` function, 5
hidden argument, 189
hidden layers, 186–187
hidden Markov models (HMMs), 217
Ho, Tin Kam, 97
holdout argument, 98
holdout sets, 21–24, 55, 135
hyperbolic tangent, 194
hyperparameters
 Bias-Variance Trade-off, 52
 in boosting, 104–106
 choosing, 19–20
 combinations of, 111–112
 confidence intervals, 114–115
 dataset size and number of
 features, 24
 grid searching with `qeFT()`, 112–116

neural networks, 189
overview, xxii
p-hacking and selection of, 24–25
phoneme dataset, 117–119
Principal Components Analysis,
 70–71
programmer and engineer data,
 113–117
in `qeDT()` function, 91–92
splitting in decision trees, 90–91
hyperplanes, 170

I

image classification. *See also*
 convolutional models
 data augmentation, 208
 Fashion MNIST data, 200–201
 overfitting, 209
 overview, xxiii, 199–200
 pretrained networks, 209
image tiling, 203–204
indicator variables, 9
indices, 8
Ioannidis, John, 229
iterative approach, 107

J

James–Stein theory, 152–153

K

K (number of folds), 55
 k (number of neighbors)
 best values of, 54
 Bias-Variance Trade-off, 15
 choosing number of, 24
 cross-validation, 55
 “Goldilocks” value, xxiii, 16, 54, 59
 overfitting, 53–54
Kaggle Appointments dataset, 42–44
kernel, applying, 177–181
kernel ridge regression, 194
kernel trick, 183
K-fold cross-validation, 55–56
k-nearest neighbors (k-NN)
 categorical features and, 17
 classification applications, 36–37
 development of, 7
 direct access to `regtools` code, 28

- general discussion, 7–9
- overview, xxii–xxiii, 7
- predicting bike ridership with, 7–9

kNN() function, 10, 64

Kozyrkov, Cassie, 230

krsFit() function, 187, 207

L

l_1 and L_2 regularization, 191

label, 4

LASSO, 157–159

- African Soil data, 159–161
- Airbnb data, 155–156
- general discussion, 153–155
- New York City Taxi data, 155–156
- overview, xxiii
- qelASSO() function, 155
- ridge regression vs., 154–155
- sparseness, 161–162

leaf nodes, 82, 91–92

learning rate

- convergence problems in neural networks, 193
- in `gbm()`, 109
- general concepts, 107–109
- overview, 103, 106–107

learnRate argument, 189

Least Absolute Shrinkage and Selection Operator. *See* LASSO

least squares, 135–136, 153

linear model

- baseball player data example, 124–126
- bias and variance in, 142–143
- blending with other methods, 148–149
- dimension reduction, 130–135
- holdout sets, 135
- least squares, 135–136
- `lm()` function, 126–127
- modeling nonlinearity with, 145–147
- NA values and impact on n , 135
- overview, 123–124
- `qeCompare()` function, 149–150
- `qeLin()` function, 127
- R^2 value(s), 137–138
- residuals, 135–136
- significance tests, 133–134

standard errors, 133

statistical significance, 131–132

use of multiple features, 127–130

validity of, 136–137

lines, xxiii, 170

listwise deletion, 27

`lm()` function, 126–127

locality, recognition of, 201–202

local minimum, 107

logistic model

- bias and variance in, 142–143
- Fall Detection data, 141–142
- Fashion MNIST data, 200–201
- Forest Cover dataset, 168
- `glm()` and `qelogit()` functions, 139
- multiclass case, 140–141
- overview, 138
- Telco Churn data, 139–140

log-odds ratio, 138

long short-term memories (LSTMs), 211

long-term time trends, 25–26

loss functions, 21

LPH

- math notation, 170–172
- overview, 170, 183
- separable case, 172–177
- separability, lack of, 177–182

M

machine learning (ML), xix–xx

- prediction and, 6–7
- role of math in, xx
- statistics terminology
 - correspondence, 223

MAPE (Mean Absolute Prediction Error), 21–22, 25, 55–58, 74

margin of error, 15

matrices, 225–226

maxdepth argument, 92

Mean Absolute Prediction Error (MAPE), 21–22, 25, 55–58, 74

Mean Squared Prediction Error (MSPE), 21, 157

Million Song dataset

- All Possible Subsets Method, 66
- overview, 63–64
- Principal Components Analysis, 66–69

- minbucket argument, 92
- minNodeSize hyperparameter, 99, 106
- minsplit argument, 92
- missing data, 27
- mlb dataset
 - blending linear model with other methods, 148–149
 - k-NN and categorical features, 17
 - linear model, 124–127
 - overfitting, 53
 - overview, 16–17
 - scaling, 18–19
- mmscale() function, 19
- momentum, 193
- MSPE (Mean Squared Prediction Error), 21, 157
- mtry argument, 92
- multiclass case, 140–141
- multivariate outliers, 27

N

- n (number of data points)
 - overfitting, 53–54
 - overview, 11
 - square root of, 54
- $n \times p$ data frame, 212
- NA values
 - in classification models, 35–36
 - in large datasets, 61–62
 - in linear models, 135
- nCombs argument, 113, 118
- nEpoch argument, 189
- neural networks (NNs). *See also* image classification
 - activation functions, 189–190
 - bias vs. variance in, 195
 - confidence intervals, 115
 - convergence problems, 193
 - development of, 7
 - Fall Detection data, 191–192
 - hyperparameters, 189
 - overview, xxiii, 185–187
 - polynomial regression and, 194
 - regularization, 190–191
 - Vertebral Column dataset, 188
 - width of, 195
 - working top of complex infrastructure, 187–188

- neurons, 186
- New York City Taxi data
 - combinations of factor levels, number of, 86
 - overview, 85–88
 - regularization, 155–156
 - tree-based analysis, 86–88
- n -fold cross-validation, 55
- nonlinearity, modeling with linear models, 145–147
- nTree hyperparameter, 99, 106
- nTst argument, 113
- numeric applications, 31
- nXval argument, 113, 117

O

- Oakden-Rayner, Lauren, 230–231
- OLS (ordinary least squares) method, 135–136
- Olshen, Richard, 81
- one-hot coding, 8
- one-sided CIs, 114
- One vs. All (OVA) method, 140–141, 168
- optimizing criterion, 174–176
- order() function, 8
- ordinary least squares (OLS) method, 135–136
- outcome variable, 4, 10
- OVA (One vs. All) method, 140–141, 168
- overall misclassification error (OME), 55
- overfitting. *See also* dimension reduction
 - best values of k and p , 54
 - convolutional models, 206–207
 - cross-validation, 55–58
 - due to features with many categories, 37
 - general discussion, 52
 - in image classification, 209
 - intuition regarding number of features and, 53
 - overview, 51
 - in random forests, 109
 - relation to overall dataset size, 53–54
 - retaining useless features, 35
 - shrinkage, 154
 - underfitting, 52

P

p (number of features), 11, 53–54. *See also* dimension reduction

parametric models, 172. *See also* linear model; logistic model; polynomial model

pars argument, 112

pef dataset, 56–57

p-hacking, 24–25, 114, 229–231

phoneme dataset, 117–119

planes, 170

plot() function, 83–85, 157

polynomial kernel, 180

polynomial model

- caution with, 149–150
- modeling nonlinearity with linear models, 145–147
- motivation, 144–145
- overview, 144
- polynomial logistic regression, 147
- programmer and engineer wages, 147–148
- qeCompare() function, 149

polynomial regression, 147, 194

pooling, 205–206

prcomp() function, 67–69

predict() function, 12–13, 69, 72

prediction. *See also* decision trees

- of bike ridership with k-NN, 7–9
- machine learning and, 6–7

pretrained networks, 209

principal component analysis (PCA), 61, 66–69, 71–72, 201

principal components (PCs)

- Bias-Variance Trade-off and, 73–74
- choosing number of, 70–71
- overview, 66
- properties of, 66–67
- qePCA() function, 71–72

print() function, 88

programmer and engineer data

- cross-validation, 56–57
- grid searching, 113–117
- polynomial models, 144, 147–148
- proxies, 65
- p-values, 90–91, 134

Q

qeAdaBoost() function, 102

qeCompare() function, 148–150

qeDT() function, 82–83, 91–92

qeFOCI() function, 75–77

qeFT() argument, 191

qeFT() function

- calling, 112–113
- overview, 112
- phoneme dataset, 117–119
- programmer and engineer data, 113–117

qeftn argument, 112

qeGBoost() function, 102–103, 105–106

qeKNN() function

- classification applications, 36–37
- direct access to regtools k-NN code, 28
- mlb dataset, 16–17
- overview, 10–13
- predicting bike ridership with, 11–13
- scaling, 18–19

qeLASSO() function, 155

qeLin() function, 127

qeLogit() function, 139, 141, 168

qeML package, xxi, 4

qeNeural() function, 187–189, 207

qePCA() function, 69, 71–72

qePolyLin() function, 146–147

qePolyLog() function, 147

qeRF() function, 97–100

qeROC() function, 47, 48

qe*-series functions

- call form, 12
- categorical features, 17
- holdout sets in, 21–22
- overview, xxii, 10
- qeDT() function, 81–82

qeSVM() function, 173–175, 182

qeText() function, 218

qeTS() function, 214

qeUMAP() function, 77

quiz data, 218–219

R

- radial basis function (RBF), 180
 - random forests, xxiii, 92, 97, 109, 213–214
 - Receiver Operating Characteristic (ROC) curve, 46–48
 - rectangular form, converting time series data to, 212–214
 - Rectified Linear Unit (ReLU), 190
 - recurrent neural networks (RNNs), 211, 217
 - recursive partitioning, 82
 - reduced convex hulls, 181
 - regression function, 3, 13–14, 32–33.
See also linear model
 - regression models
 - Bias-Variance Trade-off, 15–16
 - bike sharing dataset, 4–5
 - direct access to `regtools` k-NN code, 28
 - dirty data, 26–27
 - dummy variables and categorical variables, 9–10
 - holdout sets, 21–24
 - hyperparameters, choosing, 19–20
 - k-nearest neighbors method, 7–9
 - k-NN and categorical features, 17
 - long-term time trends, 25–26
 - machine learning and prediction, 6–7
 - missing data, 27
 - `mlb` dataset example, 16–17
 - p-hacking, 24–25
 - `qeKNN()` function, 10–13
 - scaling, 18–19
 - `regtools` package, 4–5, 16, 56, 85, 91, 117, 124, 205–206, 214
 - k-NN code, direct access to, 28
 - regularization
 - African Soil data, 159–161
 - Airbnb data, 157–159
 - LASSO, 153–155, 161–162
 - motivation, 151–152
 - neural networks, 190–191
 - New York City Taxi data, 155–156
 - overview, 151
 - ridge regression, 153–155
 - software, 155
 - vector, size of, 152
 - ReLU (Rectified Linear Unit), 190
 - ReLU() function, 163
 - remote-sensing soil analysis, 99–100
 - `replicMeans()` function, 56, 58, 113, 215–216
 - residuals, 102, 135–136
 - ridge regression, xxiii, 153–155
 - R programming language
 - generic functions, 13
 - subsetting review, 8
 - tutorial, xxi
 - root node, 82
 - R^2 value(s), 137–138
- ## S
- sampling variation, 23
 - `scale()` function, 18–19
 - scaling, 18–19
 - scientific notation, 130
 - sensitivity, 46
 - separability, 172, 177–182
 - `set.seed()` function, 23
 - shape evolution, 206–207
 - shear (twist) operation, 208
 - `showProgress` argument, 113
 - shrinkage, 154
 - shrinkage hyperparameter, 106, 109
 - significance tests, 133–134
 - `smoothingFtn` argument, 148
 - soft margin, 177, 181–182
 - software for regularization, 155
 - specificity, 46
 - splitting hyperparameters, 90–91
 - square root of n , 54
 - standard errors, 58, 133, 143
 - statistical learning, 7
 - Statistical Regression and Classification: From Linear Models to Machine Learning*, 137
 - statistical significance, 91, 131–132
 - statistics terminology correspondence, 223
 - statistics vs. machine learning in prediction, 6–7
 - `stdErrPred()` function, 143
 - Stone, Chuck, 81
 - stop words, 218

- stride, 204
- subscripts, 8
- subsetting in R, 8
- summary() function, 131–132
- support vector machines (SVMs)
 - Anderson Iris dataset, 172–177
 - convergence issues, 183
 - development of, 7
 - dot products, 171–172
 - Forest Cover dataset, 166–170, 182
 - kernel, applying, 178–181
 - kernel trick, 183
 - lack of linear separability in, 177–182
 - lines, planes, and hyperplanes, 170
 - math notation, 170–172
 - margins, 175–177
 - motivation, 166–170
 - optimizing criterion, 174–176
 - overview, xx, xxiii, 165
 - as parametric model, 172
 - separable case, 172–177
 - soft margin, 181–182
 - support vectors, 177
 - text applications, 220
 - vector expressions, 170–171

T

- tabular data, 195, 212
- Telco Customer Churn dataset
 - error rates, 39–40
 - factor data read as non-factor, 34–35
 - logistic model, 139–140
 - NA values, dealing with, 35–36
 - overview, 33–34
 - retaining useless features, 35
 - ROC curve, 47–48
- tensor, 203
- terminal node, 82
- test set, 21
- text applications
 - AG News dataset, 219–220
 - bag-of-words model, 217
 - overview, 211
 - qeText() function, 218
 - quiz data, 218–219
- tiles, 202–204

- time series
 - bias vs. variance in, 216
 - converting data to rectangular form, 212–214
 - long-term time trends, 25
 - overview, 211
 - qeTS() function, 214
 - TStoX() function, 213–214
 - weather data, 214–216
- toSubFactor() function, 166–167
- toy data, 212–213
- training data, 14
- training set, 4
- transfer learning, 194, 209
- translation invariance, 208
- tree-based methods, xxiii
- triple cross-validation, 58
- true positive rate (TPR), 46
- TStoX() function, 213–214
- tuning parameters. *See* hyperparameters

U

- unbalanced data
 - better approach to, 44–46
 - downsampling and upsampling, 42–44
 - Kaggle Appointments dataset, 42–44
 - overview, 41–42
- underfitting, 52
- Uniform Manifold Approximation and Projection (UMAP), 77
- univariate time series, 213
- upsampling, 42–44
- useless features, retaining, 35

V

- validation set, 58
- validity of linear model, 136–137
- vanishing gradient problem, 190
- variance
 - bagging and boosting, 96
 - in boosting, 106
 - in linear model, 142–143
 - in logistic model, 142–143
 - in neural networks, 195
 - overview, 15, 16, 52
 - in time series, 216

- vector, size of, 152
- vector expressions, 170–171
- Vertebral Column dataset
 - boosting, 105–106
 - classification models, 38–39
 - neural networks, 186, 188
 - random forests, 98–99
 - ROC curve, 48
- voting, 97

W

- weather data, 214–216
- weights, 202
- weights argument, 101
- wrapper functions, xxii, 10

X

- xShape argument, 189

Y

- yName argument, 97, 112