

INDEX

Symbols

* (asterisk), 178, 234, 237
\ (backslash), 51, 179
^ (beginning of a line or string), 180
\D, 180
\$ (end of a line or string), 180
/ (forward slash), 7
| (or), 11
% (percent as wildcard), 237
|| (pipe), 198
\s, 180
[] (square brackets), 180
\w, 180
. (wildcard), 180

A

A/A testing, 86
ABBA, 205
abline() command, 244
A/B testing
 applications and advanced
 considerations, 90
 definition, 77, 79
 ethical considerations, 91–93
 framework of, 83–84
 math, 80
 process, 75, 83, 85
accuracy
 of linear probability models,
 104–106
 of supervised learning models,
 134–136
activation function, 133
adding titles and labels
 title(), 14
 xlabel(), 14
 ylabel(), 14
advanced/fluent/efficient
 programming, 247

algebra of straight lines, 38
algorithms. *See also* supervised learning;
 unsupervised learning
 and E-M clustering, 157–158,
 160, 164
 Expectation step, 158–160
 Guessing step, 157–158
 natural language processing
 (NLP), 119, 209–229
 purpose, 119
alias
 definition, 6
 use, 13, 32, 35, 78
alternating least squares, 206
alternative hypothesis, 66, 72–73, 77
Anna Karenina, 69
anomaly detection, 167
application programming interface
 (API), 173, 189
APT, xxiii
apt-get command, xxiii
architectures of neural networks, 133
aspiring data scientists, xx
assignment operator, 241–242
asterisk (*), 178, 234, 237
attribute-based systems (aka content-
 based recommender systems),
 206, 228
attrition risk, 96, 98, 101,
 104–110, 114
ax.plot() method, 17
ax.scatter() method, 17

B

b (intercept), 38
backslash (\), 51, 179
baseline of variables, 38
base Python, xxiv–xxv, 176
Bayesian statistics, 91, 247

- Beautiful Soup library, 184
- beginning of a line or string (^), 180
- bell curve, 64, 155
- Betelguese, 53
- bias node, 133
- bimodal histogram, 148
- binary classification
 - applications, 90, 114–115
 - definition, 95
 - and linear regression, 110, 112
- bivariate bell curve, 154
- box plot, 61–62, 71–72
- boxplot() method, 18, 59
- bs4 package, 185
- business recommendations, making, 103–104
- business-to-consumer (B2C) business model, 91

C

- C++, 246
- central limit theorem, 64
- champion/challenger framework, 75, 83–84, 91, 93
- chatbots, 228
- classification instead of regression, 137–139
- classify() function, 151, 158
- clauses, 236
- clustering
 - in business applications, 152
 - centers, 160, 162–168
 - and collaborative filtering, 206
 - definition, 148–150
 - E-M, 158, 160
 - k-means, 245
 - methods, 164
 - and natural language processing, 226–227
- cmap parameter, 27
- code, interpretation of, 170
- coefficients
 - forecasting, 51
 - intercepts, 37–38, 43, 121
 - linear regressions, 243–244
 - logistic equations, 111
 - Pearson correlation, 21
 - supervised learning, 123

- Cohen's *d*, 88–89
- cold-start problem, 192
- collaborative filtering
 - advanced ideas, 206–207
 - definition, 191–192
 - item-based, 194–195, 199, 201, 203
 - user-based, 201–203, 205
- color map, 27
- columns in user-based collaborative filtering, 201
- combinations of metacharacters, 180
- comma-separated values, 5
- comparisons
 - content, 228
 - experimental, 203
 - group, 57, 74, 86, 88, 219
 - statistical, 77
- concatenation, 34
- confusion matrix, 105, 109, 138
- content-based recommender system (aka attribute-based systems), 206, 228
- Convergence step, 162–164
- convolutional neural networks, 133
- corpus, 211–213
- correlations, 21–27
- corr() method, 21, 26
- cosine, 197–198
- covariance, 145
 - cov method, 145
 - cov/np.cov method, 145, 155
 - matrix, 155, 157
- .csv, 2, 4–5
- cubic curve, 46
- customers, 201
 - attrition, 96
 - segmentation, 153

D

- \D, 180
- data analysis
 - correlation analysis, 21
 - exploratory, 1
 - machine learning, 117, 141
 - making plots, 13
 - of subsets, 10
 - summary statistics, 8

- data cleaning, 32–34
- data engineering, 247
- data scientists
 - aspiring, xx
 - overview/definition of, 32, 75, 105, 153, 231
 - professional, xxi
 - as statisticians, xix, 246
- decision trees, 130–132
- decomposition, 206
- deep neural networks, 133
- density-based spatial clustering of
 - applications with noise (DBSCAN), 166–167
- dependencies, 185
- derived feature, 108, 137
- describe() method, 9
- dev.off() method, 244
- DevOps, 247
- dimension, 154
- distance, 216
 - k-nearest neighbors, 124–125
 - distribution of data, 144
 - dot product, 198
- distplot() method, 63

E

- effect sizes, 86
 - Cohen's *d*, 88–89
- embeddings, 219
- E-M clustering, 155–164
- end of a line or string (\$), 180
- end tag, 172
- engineered feature, 108
- ensemble method, 131
- error measurements, 40–43
- escape sequences, 178–180
- ethical considerations of A/B testing, 91–93
- Euclidean distance, 216
- euclidean() function, 216
- exit probability, 101, 105
- expectation-maximization, 157
- Expectation step, 158
- expected value, $E()$, 62, 64, 80–81, 100
- experiment, 76, 79
- exploration/exploitation trade-off, 91

F

- false positives, 105–106, 138
- feature engineering, 123
- features, 142
- fetchall() method, 236
- filtering, 200, 206
- find_all() method, 185, 188
- find() method, 174–176
- fit() method, 37, 46, 165, 227
- fitting
 - logistic function data, 112
 - regression, 37–38
- forecasting
 - best regression to use with, 50–55
 - customer demand, 31–32
 - with linear regression, 35, 43, 45, 47, 247
 - methods, 51, 54
- for loop, 67, 236
- forward slash (/), 7
- function(s)
 - activation, 133
 - calls, xx
 - classify(), 151, 158
 - collaborative filtering and, 202–203, 205
 - euclidean(), 216
 - generating, 143, 145
 - getcenters(), 160–161
 - KMeans(), 165
 - learned, 142
 - multivariate_normal(), 151
 - norms of, 198
 - np.mean() function, 145
 - plt.scatter() function, 159–160
 - print() function, 6, 11, 47
 - R code and, 242
 - set() function, 203
- future predictions, 102–103

G

- Gaussian distribution, 145, 155
- generating and exploring data, 142–148
- Gensim package, 218
- getcenters() function, 160–161
- goodness of fit, 40–41

- groupby() method, 11–12, 14
- group comparisons, 57
 - hypothesis testing, 66
 - in a marketing context, 70
 - visual, 61
- guessing step, 157–159

H

- Hamlet*, 225
- head() method, 6, 33
- heat maps, 26–28
- hierarchical clustering, 167
- histogram, 18–20, 29, 64, 67, 144–149
- hour.csv*, 4–7
- HTML code
 - elements, 172, 185
 - parsing, 173
- Hyndman, Rob, 32
- hypothesis
 - hypothesized line, 38–39, 41–43
 - null vs. alternative hypothesis, 66, 72–73
 - testing, 65–66, 68, 77

I

- identically distributed, 64
- identity matrix, 157–158
- idx variable, 126
- if...then statements, xx
- independent samples, 67
- input layer, 133
- installation of Python
 - on Linux, xxiii
 - on macOS, xxii–xxiii
 - on Windows, xxii
- interaction matrix, 193–196, 199, 201–206
- interpretable models, 131
- item-based collaborative filtering, 194–201

J

- JavaScript, 171, 188, 246
- Julia, 246
- JupyterLab, xxv
- Jupyter Notebook, xxv

K

- Kiros, Ryan, 223
- k-means clustering, 164–166
- KMeans() function, 165
- k-nearest neighbors (k-NN), 124
 - implementing, 126
 - method, 125
 - with sklearn, 127–128
 - other supervised learning algorithms, 128–129

L

- Last Continent, The*, 210
- latent variable models, 167
- learned functions
 - decision trees and, 132–133
 - definition of, 122–123
 - supervised learning and, 128–129, 135, 142
- Legendre, Adrien-Marie, 43
- linear algebra methods, 246–247
 - alternating least squares, 206
- linear probability model (LPM), 95, 97–109
 - weaknesses of, 109–110
- linear regression, 99
 - classification, 137
 - decision trees, 130
 - definition of, 29
 - and forecasting, 32, 43, 54–55, 247
 - and k-nearest neighbors, 124, 126, 128–129, 131
 - measuring accuracy of, 104
 - methods of, 47–48
 - multivariate, 31, 45–46, 106
 - performing, 35–39, 51, 99, 101–102, 108, 110–112
 - as prediction method, 121
 - with R, 243–244
 - and supervised learning, 117, 122–124
 - univariate, 45
- line continuation characters, 51
- line of best fit, 36, 39, 43, 54, 100
- line plot
 - how to make, 17
 - purpose of, 48

Linux console, xxiii
literal_eval() method, 156
lm() command, 243
loc() method, 10
logistic curve, 110–112
logistic regression, 110–114
LPM. *See* linear probability model
lxml package, 184–187

M

m (slope), 38
machine learning
 clustering, 152
 datasets, 119
 decision trees, 131
 distances, 125
 forecasting, 53
 methods, 36–37, 114–115
Madonna, 205
magnitude of correlations,
 22–25
Mann-Whitney U test, 68
marker style, 16
MATLAB, 246
Matplotlib, 13, 18, 34–35,
 59, 71
matrix laboratory, 246
Maximization (M) step, 157,
 160–163
max() method, 8
mean, calculating, 8
mean absolute error (MAE), 41–42, 51,
 52, 104, 134
mean() method, 8–9, 11
median() method, 8–9, 78
Mengele, Josef, 92
merge() method, 79
metacharacters, 177–182
min() method, 8
Moby Dick, 225
model, 38, 123
modes, 148
monotonic trends, 114
multi-armed bandit problem, 91
multivariate linear regression, 31,
 45–46, 106
multivariate_normal() function, 151

N

NaN (not a number), 33
natural language, 211
natural language processing (NLP)
 applications, 228
 definition, 119
 methods, 207
 sentiment analysis, 119, 228
negatively correlated variables, 24
neighbors, 125
neural networks, 132–134
nighttime data, 10–11
NLP. *See* natural language processing
nodes, 133
nonparametric statistics, 68, 247
normal distribution, 145, 147
norms, 198
not a number (NaN), 33
np.append() method, 43
np.cov() method, 145, 155
np.mean() function, 145
null hypothesis, 66–69
number of observations (nobs), 89–90
NumPy, 41, 78, 82, 126

O

omitted variable, 25
one-dimensional dataset, 154
or (|), 11
overfitting, 53, 134

P

package managers, xxiii
pair plot, 20
pandas, xxiv, 6, 32–33
 dataframes, 32–33, 183
 installation, xxiv–xxv, 6
parametric test, 68
parsing HTML, 173–176, 186–188
Pearson correlation coefficient, 21–22
percent as wildcard (%), 237
pip, xxiv–xxv, 6
pipe (|), 198
.pkg file, xxiii
plagiarism, detecting, 210–211,
 216–218, 222–223, 225
plot() command, 244

- plotting data
 - adding titles and labels, 14
 - appearance
 - box plot, 18
 - histogram, 19
 - line plot, 17
 - pair plot, 20
 - scatterplot, 16–17
 - simple plot, 13–14
 - subsets, 15–16
- `plt.scatter()` function, 159–160
- `png()` command, 244
- popularity-based recommendations, 192, 194
- populations and samples, 58
- positively correlated variables, 22
- postdiction, 134
- practical significance vs. statistical significance, 69–70
- Pratchett, Terry, 210, 225
- precision, 105–106
- `predict()` method, 134
- price elasticity of demand, 90
- `print()` function, 6, 11, 47
- programming languages, 231, 238, 246
- proxy, 120
- proxy servers, 188
- p*-value, 65
- Python
 - base Python, xxiv–xxv, 176
 - installation
 - on Linux, xxiii
 - on macOS, xxii–xxiii
 - on Windows, xxii
 - packages, xxiv–xxv, 6, 217
 - working with, 5

Q

- quantifying similarity between words, 211–213
- quantitative methods, 32, 62, 117
- queries, 234–236

R

- R (programming language), 241–243
 - and linear regression, 243–244
- random forests, 131–132

- randomized controlled trials, 91
- randomness, 23–24, 35–38, 44
- `random.random()` method, 82
- random samples, 59–62
- random seed, 82
- `read_csv()` method, 6–7, 32, 58
- reading data, 32–34
- recall, 105
- recommendation systems, 191
 - advanced approaches to, 206–207
 - case study, 204–205
 - collaborative filtering, 194–195
 - implementing, 199–201
 - using, 201–204
 - cosine similarity, 197, 198–199
 - popularity-based, 192–194
 - vector similarity, 195–201
- recurrent neural networks, 134
- `re.finditer()` method, 183
- regression line, 46–47, 49, 51–52, 100–101, 109–110, 121–122, 127, 245
- regressor, 37
 - decision trees, 130–131
 - k-NN regressors, 127–129
 - with multivariate models, 107, 136
 - random forests, 131–132
- regular expressions, 176–182, 184–185, 238
- related samples, 67
- `re.search()` method, 176
- `reshape()` method, 37
- root mean squared error (RMSE), 42–43, 50–54, 104
- rotating proxies, 188
- `rpy2` package, 242

S

- `\s`, 180
- Salk, Jonas, 92–93
- `sample()` method, 60
- SAS, 246
- Scala, 246
- `scatter()` method, 13, 35
- scatterplot, 14, 20, 35, 48
- scikit-learn (sklearn), 37, 123, 127–128
- SciPy package, 67–68

- scraping, 188–189. *See also* web scraping
 - an email address, 173–174
 - and parsing HTML tables, 186–188
- seaborn package, 18–20, 63
- search() method, 176
- seasonal data, analyzing, 11–12
- sensitivity, 105
- set() function, 203
- significance level, 66, 73, 76–77
- singular value decomposition, 206
- skip-thoughts, 223–225
- sklearn. *See* scikit-learn
- solve_power() method, 90
- sort_values() method, 193
- SPSS, 246
- spurious correlation, 25
- SQL
 - introduction, 234
 - joining tables, 238–241
 - managing soccer with, 232–234
 - running queries, 235–238
 - setting up a database, 234–235
 - SQLite3 package, 235
- square brackets ([]), 180
- standard deviation, 8, 86–88, 145–147
- start tag, 172
- Stata, 246
- statistical power, 88–90
- statistical significance vs. practical significance, 69–70
- statistics packages, 246
- std() method, 8
- subplots() method, 13
- subsets, plotting, 15–16
- summary statistics
 - calculating, 8–9, 72
 - and plots, 58–59
- sum() method, 193
- supervised learning
 - algorithms, 128–129
 - decision trees, 130–131
 - random forests, 131–132
 - neural networks, 132–134
 - comparison to unsupervised learning, 142
 - definition of, 117

- k-NN, 124–127
 - process, 122–124
 - multivariate models, 136
 - sklearn, 127–128
- system of equations, 213–217

T

- terminal (Linux console), xxiii–xxiv
- test set, 53, 136
- titles and labels
 - title(), 14
 - xlabel(), 14
 - ylabel(), 14
- Tolstoy, Leo, 69
- tools to run Python code, xxv
- topic modeling, 225–226
- “trained a model,” 38
- training set, 51
- transpose a matrix, 205
- trigonometric curve, 49
- trigonometry, 47
- true positives, 105, 138
- t-test. *See also* Welch’s t-test
 - calculating, 67–70
 - experimentation, 76–77
 - group comparison, 73–74
 - performing, 79–80, 83
 - statistical power, 88
- Twyman’s law, 84–86, 93

U

- uncorrelated variables, 22, 26
- units of exiting, 100
- univariate
 - bell curve, 154
 - linear regression, 45
- Universal Sentence Encoder (USE), 224
- unsupervised learning
 - clustering, 150
 - comparison to supervised learning, 143
 - definition, 139
 - E-M clustering, 155
 - methods, 143, 167–168
- user-based collaborative filtering, 201, 203, 205

V

Varian, Hal, xix
variance, 145
vector
 degree, 196–197, 199, 225
 similarity, measuring, 195
 space, 219

W

\w, 180
web scraping, 169, 171–172, 189. *See also*
 scraping
 converting results to data, 182–184
 warning, 173
websites, 170–171
 predicting traffic, 118–119

Welch's t-test, 68. *See also* t-test
Wilcoxon rank-sum test, 68
wildcard (.), 180
Williams, Robbie, 205
word2vec, 211, 218, 221–224

X

x-position, 243

Y

y-intercept, 38
Yum, xxiii

Z

zero-based indexing, 10