Preface p. xiii
Natural Language Features p. 1
Language and modeling p. 3
Linguistics for text analysis p. 3
A glimpse into one area: morphology p. 5
Different languages p. 6
Other ways text can vary p. 7
Summary p. 8
In this chapter, you learned: p. 8
Tokenization p. 9
What is a token? p. 9
Types of tokens p. 13
Character tokens p. 16
Word tokens p. 18
Tokenizing by n-grams p. 19
Lines, sentence, and paragraph tokens p. 22
Where does tokenization break down? p. 25
Building your own tokenizer p. 26
Tokenize to characters, only keeping letters p. 27
Allow for hyphenated words p. 29
Wrapping it in a function p. 32
Tokenization for non-Latin alphabets p. 33
Tokenization benchmark p. 34
Summary p. 35
In this chapter, you learned: p. 35
Stop words p. 37
Using premade stop word lists p. 38
Stop word removal in R p. 41
Creating your own stop words list p. 43
All stop word lists are context-specific p. 48
What happens when you remove stop words p. 49
Stop words in languages other than English p. 50
Summary p. 52
In this chapter, you learned: p. 52
Stemming p. 53
How to stem text in R p. 54
Should you use stemming at all? p. 58
Understand a stemming algorithm p. 61
Handling punctuation when stemming p. 63
Compare some stemming options p. 65

Lemmatization and stemming p. 68
Stemming and stop words p. 70
Summary p. 71
In this chapter, you learned: p. 72
Word Embeddings p. 73
Motivating embeddings for sparse, high-dimensional data p. 73
Understand word embeddings by finding them yourself p. 77
Exploring CFPB word embeddings p. 81
Use pre-trained word embeddings p. 88
Fairness and word embeddings p. 93
Using word embeddings in the real world p. 95
Summary p. 96
In this chapter, you learned: p. 97
Machine Learning Methods p. 99
Overview p. 101
Regression p. 105
A first regression model p. 106
Building our first regression model p. 107
Evaluation p. 112
Compare to the null model p. 117
Compare to a random forest model p. 119
Case study: removing stop words p. 122
Case study: varying n-grams p. 126
Case study: lemmatization p. 129
Case study, feature hashing p. 133
Text normalization p. 137
What evaluation metrics are appropriate? p. 139
The full game: regression p. 142
Preprocess the data p. 142
Specify the model p. 143
Tune the model p. 144
Evaluate the modeling p. 146
Summary p. 153
In this chapter, you learned: p. 153
Classification p. 155
A first classification model p. 156
Building our first classification model p. 158
Evaluation p. 161
Compare to the null model p. 166
Compare to a lasso classification model p. 167

Tuning lasso hyperparameters p. 170
Case study: sparse encoding p. 179
Two-class or multiclass? p. 183
Case study: including non-text data p. 191
Case study: data censoring p. 195
Case study: custom features p. 201
Detect credit cards p. 202
Calculate percentage censoring p. 204
Detect monetary amounts p. 205
What evaluation metrics are appropriate? p. 206
The full game: classification p. 208
Feature selection p. 209
Specify the model p. 210
Evaluate the modeling p. 212
Summary p. 220
In this chapter, you learned: p. 221
Deep Learning Methods p. 223
Overview p. 225
Dense neural networks p. 231
Kickstarter data p. 232
A first deep learning model p. 237
Preprocessing for deep learning p. 237
One-hot sequence embedding of text p. 240
Simple flattened dense network p. 244
Evaluation p. 248
Using bag-of-words features p. 253
Using pre-trained word embeddings p. 257
Cross-validation for deep learning models p. 263
Compare and evaluate DNN models p. 267
Limitations of deep learning p. 271
Summary p. 272
In this chapter, you learned: p. 272
Long short-term memory (LSTM) networks p. 273
A first LSTM model p. 273
Building an LSTM p. 275
Evaluation p. 279
Compare to a recurrent neural network p. 283
Case study: bidirectional LSTM p. 286
Case study: stacking LSTM layers p. 288
Case study: padding p. 289

Case study: training a regression model p. 292
Case study: vocabulary size p. 295
The full game: LSTM p. 297
Preprocess the data p. 297
Specify the model p. 298
Summary p. 301
In this chapter, you learned: p. 302
Convolutional neural networks p. 303
What are CNNs? p. 303
Kernel p. 304
Kernel size p. 304
A first CNN model p. 305
Case study: adding more layers p. 309
Case study: byte pair encoding p. 317
Case study: explainability with LIME p. 324
Case study: hyperparameter search p. 330
Cross-validation for evaluation p. 334
The full game: CNN p. 337
Preprocess the data p. 337
Specify the model p. 338
Summary p. 341
In this chapter, you learned: p. 342
Conclusion p. 343
Text models in the real world p. 345
Appendix p. 347
Regular expressions p. 347
Literal characters p. 347
Meta characters p. 349
Full stop, the wildcard p. 349
Character classes p. 350
Shorthand character classes p. 352
Quantifiers p. 353
Anchors p. 355
Additional resources p. 355
Data p. 357
Hans Christian Andersen fairy tales p. 357
Opinions of the Supreme Court of the United States p. 358
Consumer Financial Protection Bureau (CFPB) complaints p. 359
Kickstarter campaign blurbs p. 359
Baseline linear classifier p. 361

Read in the data p. 361
Split into test/train and create resampling folds p. 362
Recipe for data preprocessing p. 363
Lasso regularized classification model p. 363
A model workflow p. 364
Tune the workflow p. 366
References p. 369
Index p. 379

Table of Contents provided by Blackwell's Book Services and R.R. Bowker. Used with permission.