숫자

1×4 행렬 428
2×1 행렬 428
2×2 행렬 430–431
3×2 행렬 428, 433
3×3 행렬 427, 431
3×4 행렬 428
4×1 행렬 428

A

adaboost.loadSimpData( ) 169, 178
adaBoostTrainDS( ) 177–180
adaClassify( ) 177
aggErrors.sum( ) 175
alphas[i].copy( ) 138
APIs(응용 프로그램 인터페이스), 구글 쇼핑 218–224
apriori( ) 294–295, 299, 301
apriori.getActionIds( ) 305
aprioriGen( ) 294–295, 299
argsort( ) 347, 422
ascendTree( ) 327
autoNorm( ) 38–39, 45
AWS(아마존 웹 서비스) 388–395
  계정 만들기 389–390
  제공 서비스 388–389
  하둡 작업 실행하기 390–395
AWS 계정 164–190

B

bayes.loadDataSet( ) 87, 90
bayes.spamTest( ) 99
bayes.testingNB( ) 94
biKmeans( ) 273–275, 280
BillDetail 객체 303
billTitleList.append.line.strip( ) 305
binSplitDataSet( ) 231
buildStump( ) 172, 174–176

C

C++ 라이브러리 18
caclMostFreq( ) 104
calcConf( ) 298–299
calcEk( ) 145, 154
calcMostFreq( ) 102–103
calcShannonEntropy( ) 59
CART(분류와 회귀 트리) 알고리즘 232–237
  코드 실행 236–237
  트리 구축 233–236
chkBtnVar.get( ) 255
chooseBestFeatureToSplit( ) 58
chooseBestSplit( ) 231–235, 238, 245
CircleCollection 객체 198
classCount.keys( ) 61
classifyNB( ) 95
classifyPerson( ) 41
classifyVector( ) 126
clipAlpha( ) 137
clusterClubs( ) 280
colicTest( ) 125–126
configure_options( ) 405
copy( ) 140
corrcoef( ) 246, 364
Create Job Flow 버튼 394
createBranch( ) 50
createDataSet( ) 26, 55, 57–59, 62, 73
createForeCast( ) 248
createInitSet( ) 324
createPlot( ) 65, 69, 70, 75
createTree( ) 230–233, 243–245, 323–325
createVocabList( ) 86
crossValidation( ) 223

D

Data.gov 439
Data=svdRec.loadExData( ) 360
dataMatrix.max( ) 171
dataMatrix.min( ) 171
dataMatrix.transpose( ) 113
dataSet.items( ) 322
datingClassTest( ) 40
det( ) 431
disp( ) 319, 325
distances.argsort( ) 29
distEclud( ) 265–266
distSLC( ) 280
drawNewTree( ) 252, 255
drawTree( ) 254

E

EC2(일래스틱 컴퓨트 클라우드) 389
EMR (일래스틱 맵 리듀스 서비스)
  AWS 390–395
  mrjob 기본 구조 398
extend( ) 57

F

fig.clf( ) 65, 69, 186
findPrefixPath( ) 327–329
float( ) 230, 278
float.tolSentry.get( ) 255
for 반복문 416–418
fpGrowth.loadSimpDat( ) 324

FP-성장 알고리즘 315-337
  FP-트리 318-325
  FP-트리 구축 320-325
  FP-트리 데이터 구조 319-320
  FP-트리 데이터 집합 부호화 316-319
  FP-트리 빈발 아이템 마이닝 325-331
  예제
    뉴스 사이트에서 클릭 스트림 마이닝 336-337
    트위터 피드에서 함께 발생하는 단어 찾기 331-335
FP-트리 부호화 316-318
FP-트리 조건 328-331
FP-트리(빈발 패턴 트리) 318-325
  구축 320-325
  데이터 구조 319-320
  데이터 집합 부호화 316-319
  빈발 아이템 마이닝 325-331
    FP-트리 조건 생성 328-331
    기본 패턴 조건 추출 326-328
fr.readlines( ) 32, 75, 230, 264, 346
frozenset( ) 291
frTest.readlines( ) 125
frTrain.readline( ) 125
fw.close( ) 74, 276

G

generateRules( ) 298, 301
geoGrab( ) 276-277
getActionIds( ) 304, 308
getActionsId( ) 305
getBill( ) 303, 305
getBillActionVotes( ) 304
getInputs( ) 255-256
getLotsOfTweets( ) 333
getMean( ) 241
getNumLeafs( ) 66-70
getTopWords( ) 104
getTransList( ) 307
getTreeDepth( ) 66-70
gradAscent( ) 114
GradientAscent( ) 122
grid( ) 251-253
GUI(Tkinter 사용) 250-257

H

handwritingClassTest( ) 44, 160
headerTable.items( ) 329
headerTable.keys( ) 322

I

ICA(독립적인 구성요소 분석) 342-343
if 문 416
imread( ) 280
imshow( ) 281
innerL( ) 148, 154
input data(입력 데이터) 13-14
inputFile.txt 파일 385-387, 391-392, 395, 398, 401
inputTree.keys( ) 72
IntVar( ) 252
isTree( ) 241

J

json.loads( ) 219
json.loads(c.read( )) 275

K

kernelTrans( ) 153-156
kMeans( ) 266, 272
kNN.classifyPerson( ) 42
kNN.createDataSet( ) 27
kNN.datingClassTest( ) 40
kNN.handwritingClassTest( ) 45
kosarak.dat 파일 336
k-최근접 이웃 알고리즘의 거리 측정 24-31
  검사 30-31
  파이썬으로 데이터 불러오기 26-28
  함수 28-30
k-최근접 이웃 23-47, 396
  거리
    검사 30-31
    측정 24-30
    파이썬으로 데이터 불러오기 26-28
    함수 29-30
  데이트 사이트 예제 31-42
    검사 39-41
    매스플롯라이브러리로 scatter 플롯 생성하기 34-37
    사용 41-42
    수치형 값 정규화 37-39
    텍스트 파일의 데이터 구문 분석 32-34
  필기체 인식 예제 42-46
    검사 44-46
    검사 벡터로 이미지 변환 43-44
k-평균 군집화 알고리즘 261-282
  개요 262-268
  성능 개선 268-270
  양분하는 k-평균 알고리즘 270-273
  지도상의 지역점 군집화 예제 274-281

L

labelCounts.keys( ) 54
labelMat=loadDataSet( ) 115
len(fr.readlines( )) 32
linalg.det( ) 197

linalg.norm( ) 364
line.rstrip( ) 385–387
line.strip( ) 33, 113, 125, 136, 180, 264, 276
linearSolve( ) 245, 249
list 414
loadDataSet( ) 85, 93, 113, 180, 197, 264, 290, 346
loadExData( ) 360
loadImages( ) 160
loadSimpData( ) 169
localD.items( ) 322
localWords( ) 101–103
log( ) 89
logRegres.loadDataSet( ) 115, 118, 121
logRegres.multiTest( ) 126
lowDMat 347
LSI(잠재적 의미 색인) 355–356
LWLR(지역적 가중치가 부여된 선형 회귀) 201–205

M

map( ) 408
map_fin( ) 408
mapper( ) 400
mapper_final( ) 400
massPlaceFind( ) 276–277
mat keyword 420
mat( ) 함수 20
mat(classLabels).transpose( ) 113, 138, 147, 149
mat(labelArr).transpose( ) 156, 160
mineTree( ) 329–331
mineTweets( ) 334
modelErr( ) 244
modelEval( ) 248
modelLeaf( ) 244
modelTreeEval( ) 248
mrjob 기본 구조 397–401
  EMR로 매끄러운 통합 398
  분산처리하는 SVM 예제 404–410
  스크립트 작성 398–401
Mrjob 패키지 423
MRmean.run( ) 399
mrMeanCode 폴더 391–392
mrMeanInput 폴더 391–392
mrMeanMapper.py 파일 385–387, 391–392, 395
MRsvm.run( ) 406
MRsvm 클래스 407
multiTest( ) 126
myFPtree.disp( ) 325
myLabel.grid( ) 251
myMat 행렬 427
mySent.split( ) 95
myTree.keys( ) 66, 68

N

n! elements 431
NaN values 349
nonzero( ) 146
NumPy 라이브러리 19–21, 418–423
NumPy 행렬 386

O

oS.alphas[i].copy( ) 145

P

PathCollection 객체 204, 347
PCA NumPy 모듈 345–348
PCA(주요 구성요소 분석) 341–353
  NumPy 모듈 345–348
  좌표 축 이동 343–345
  차원 축소
    기술 342–343
    반도체 제조 데이터 예제 348–352
pca.replaceNanWithMean( ) 349–350
perasSim( ) 365
plot( ) 254–256
plotMidText( ) 68–70
plotNode( ) 65
plotTree( ) 68–70
plt.figure( ) 35, 115, 185, 198, 204, 209, 279, 347
plt.show( ) 35, 65, 115, 199, 204, 209, 279, 348
predictedVals.copy( ) 171
predStrengths.argsort( ) 185
preFix.copy( ) 329
printMat( ) 375
prune( ) 240–241
PyPy 18
Python 412–425
  Mrjob 패키지 423–424
  NumPy 라이브러리 19–21, 418–423
  SVD 359–362
  데이터 불러오기 26–28
  뷰티플 수프 패키지 423
  사용하는 이유 16–18
    단점 18
    실행 가능한 의사코드 16
    인기 16–17
    장점 17–18
  설치 412–414
    Mac OS X 413
    MS 윈도우 412–413
    리눅스 OS 413–414
  소개 414–418
    구조 제어 416–417
    데이터 유형 414–416

함축 리스트 417–418
스마트 투표 데이터 소스 424–425
파이썬 트위터 모듈 425

R

randCent( ) 264
random.shuffle( ) 222
raw_input( ) 41
readlines( ) 113, 136, 180, 197
README.txt 파일 423
recommend( ) 368–370, 372
ReDraw 명령 253
reDraw( ) 254–256
reDraw.canvas.get_tk_widget( ) 254
reDraw.canvas.show( ) 255
reDraw.f.clf( ) 255
regErr( ) 235, 245
regLeaf( ) 234, 245
replaceNanWithMean( ) 349
retrieveTree( ) 68
ridgeRegres( ) 209
ridgeTest( ) 209, 222
ROC(수신자 조작 특성) 곡선
정확도와 재현율 182–187
root.mainloop( ) 251–253
root=Tk( ) 252
rootNode.disp( ) 320
RSS 피드 불러오기 100–104
rssError( ) 205, 213, 223
rulesFromConseq( ) 298–299

S

S3(심플 스토리지 서비스) 388
scanD( ) 290–291, 292–293
scatter( ) 254, 256
SciPy 16–17
searchForSet( ) 220
secondDict.keys( ) 66, 68, 72
selectJ( ) 145–146
selectJrand( ) 137, 146
self.children.values( ) 319
set( ) 294
setDataCollect( ) 220
sigmoid( ) 114
sign( ) 176, 185–186
sin( ) 280
SMO 알고리즘으로 최적화 134–142
간략한 형태의 SMO 알고리즘으로
적은 양의 데이터 집합 해결하기 135–142
플랫의 SMO 알고리즘 106, 143–151
smoSimple( ) 146–147
sort( ) 290, 293
sorted.classCount.iteritems( ) 29, 61
sorted.freqDict.iteritems( ) 101
sortedIndicies.tolist( ) 186
spamTest( ) 97–101
splitClustAss.copy( ) 271
splitDataSet( ) 57
stageWise( ) 213
standEst( ) 368, 372–373
standRegres( ) 197, 203, 223
stanEst( ) 369
steps( ) 400, 406
stumpClassify( ) 171–172, 178
SVD(특이 값 분해) 354–379
예제
SVD로 이미지 압축하기 375–378
레스토랑 메뉴 추천 엔진 367–375
응용프로그램 355–357
LSI 355–356
추천 시스템 356–357
파이썬 359–362
행렬 인수분해 358–359
협력적 여과 기반 추천 엔진 362–367
유사도 362–366
평가 366
svdEst( ) 372
svdRec.loadExData( ) 365, 369
SVD로 이미지 압축 375–378
svmMLiA.testRbf( ) 157
SVMs(지지 벡터 머신) 128–163
SMO 알고리즘(최적화) 134–142
일반적인 기본 구조로 접근하기 134
최대 마진 찾기 131–134
데이터 분리 129–131
커널 151–158
검사 155–158
고차원의 데이터 매핑 151–152
반지름 성향 함수 152–155
필기체 분류 예제(다시 적용하기) 159–162

T

T 메소드 421
testDigits( ) 161
testingNB( ) 93
testRbf( ) 155, 158, 161
textParse( ) 98, 334
tile( ) 38
time.sleep( ) 219
Tkinter(GUI 사용) 250–257
tolist( ) 186
topNfeat 346–347

transDict.keys( ) 308–309
treeForeCast( ) 248
treeNode( ) 229
treePlotter( ) 68
treePlotter.createPlot( ) 65

U
UCI 데이터베이스 75
updateEk( ) 145
updateHeader( ) 323
updateTree( ) 323
urlencode( ) 276

V
var( ) 235
votesmart.votes.getBillsByStateRecent( ) 302

W
wei.getA( ) 115
ws.copy( ) 212
wsMax.copy( ) 213

X
xCopy=xMat.copy( ) 199

ㄱ
값 분해 396
검사
  k-최근접 이웃 알고리즘 29
  데이트 사이트 예제 40–41
  말의 배앓이 치사율 예제 122–127
  스팸 메일 예제 97–99
  에이다부스트 메타 알고리즘으로 분류 177–179
  커널 155–158
  필기체 인식 예제 44–45
결합 확률 435–436
계수, 데이터 이해를 위한 축소 207–215
  능형 회귀 207–210
  라소 기술 210–211
  전방향 단계별 회귀 211–215
고유 값 345–347, 350–351, 353
구글 쇼핑 API로 데이터 수집 218–224
구조 제어 416–417
국회 투표의 패턴으로 연관 규칙 310–312
  아이템 집합으로 마이닝하기 296–301
군집 11–14, 396
기계 학습
  기술 11
  단계 14–16
  맵 리듀스의 기본 구조 395–397
  미래 7
  센서 6–7
  알고리즘 선택 12–14
  전문용어 8–11
  정의 3–7
기계 학습 단계 14–16
기본 패턴 조건 326–328
기본 패턴 조건 추출 326–328
기술(기계 학습) 11
기울기 상승(로지스틱 회귀 분류 최적화) 110–112

ㄴ
나이브 베이스 78–106, 396, 402
  개요 79–80
  개인 광고 예제에서 지역 특색 도출하기 99–105
    RSS 피드 불러오기 100–104
    지역적으로 사용되는 단어 표현 104–105
  문서 분류 83–85
  스팸 메일 예제 95–99
    검사 97–99
    텍스트 토큰 95–97
  조건부 확률 81–82
  텍스트 분류 85–94
    검사 91–94
    단어 벡터로 확률 계산 88–91
    모델 94
    텍스트로 단어 벡터 만들기 85–87
나이브 베이스 분류기로 문서 분류 83–85
뉴스 사이트에서 클릭 스트림 마이닝 336–337
능형 회귀 207–210

ㄷ
단계별 회귀(전방향) 211–215
단어 벡터
  텍스트로 만들기 85–87
  확률 계산 88–91
데럴 후프 5
데이터
  센서 6–7
  파이썬으로 불러오기 26–28
데이터 구문 분석(텍스트 파일) 32–34
데이터 샘플링(분류 불균형) 188–189
데이터 유형 414–415
데이터 집합 분할(의사결정 트리) 56–59
데이터랭글링 440
데이트 사이트 예제 31–42
  검사 39–41
  매스플롯라이브러리로 scatter 플롯 생성 34–37
  사용 41–42
  수치형 값 정규화 37–39
  텍스트 파일 구문 분석 32–34
독립적인 구성요소 분석 ICA 참조

독버섯 속성 예제 312–313
딕셔너리 415

ㄹ

라소 기술 210–211
레고 가격 예제 217–224
레스토랑 메뉴 추천 엔진 구축하기 367–375
  SVD로 추천 개선하기 370–373
  과제 374–375
  맛보지 못한 음식 추천하기 367–370
로지스틱 회귀 분류 107–127
  말의 배앓이 치사율 예제 122–127
    검사 125–127
    데이터에서 누락된 값 다루기 123–125
  시그모이드 함수 108–110
  최적화 110–122
    가장 좋은 매개변수 찾기 112–115
    기울기 상승 사용 110–112
    의사결정 경계선 플롯 115–116
    확률적인 기울기 상승 사용 117–122
리눅스 OS에서 파이썬 설치 413–414
리눅스 명령 385
리소스 439–440

ㅁ

마이크로소프트 윈도우 OS에서 파이썬 설치 412–413
말의 배앓이 치사율 예제 122–127
  검사 125–127
  데이터에서 누락된 값 다루기 123–125
매스플롯라이브러리 413
  scatter 플롯 34–37
  Tkinter 접속 254–257
  개요 64–66
  플롯하기 66–71
매스플롯라이브러리로 scatter 플롯 34–35
매퍼 382–384, 386, 399
맥 OS X에서 파이썬 설치 413
맵 리듀스 기본 구조 380–411
  mrjob 기본 구조 397–401
    EMR로 매끄러운 통합 398
    스크립트 작성 398–401
  SVM 분산처리
    mrjob 수행 404–410
    예제 402–410
    페가소스 알고리즘 사용 403–404
  개요 381–384
  기계 학습 395–397
  왜 필요한가 410
  하둡 작업 384–388
    AWS 실행 388–395
    평균과 분산 리듀서 분산처리 386–388
    평균과 분산 매퍼 분산처리 385–386
모델 트리 243–246
목적 변수 10
미분행렬 432
밀도 추정 12–13

ㅂ

반도체 제조 데이터 줄이기 예제 348–349
반지름 성향 함수 152–155
배깅 165–166
벡터 머신 380, 396, 402–404, 411
벤 다이어그램 436
변형불가 행렬 430
변화량, 성향/변화량 관계 215–217
부스팅 166
부울형 241
분류
  나이브 데이스 분류기로 문서 분류 83–85
  의사결정 트리 72–73
  텍스트 분류 85–95
분류 불균형 182–189
  데이터 샘플링 188–189
  또 다른 성능 측정 방법 182–187
  비용 함수로 의사결정 다루기 187–188
분류기
  데이터 집합의 다양한 표본 사용 165–167
    배깅 165–166
    부스팅 166–167
  비용 함수로 의사결정 다루기 187–188
  에이다부스트 메타 알고리즘으로 검사 177–179
  오류에 초점을 맞춘 분류기 개선 167–169
  최적화 문제 구성하기 132–133
분류와 회귀 트리 CART 참조
분류 작업 9
분산처리된 SVMs 예제 402–410
  mrjob 수행 404–410
  페가소스 알고리즘 403–404
불러오기
  RSS 피드 100–104
  데이터를 파이썬으로 26–28
뷰티플 수프 패키지 423
비용 함수, 분류기의 의사결정 다루기 187–188
빈발 패턴 성장 FP–성장 참조
빈발 패턴 트리 FP–트리 참조

ㅅ

사용자 기반 유사도와 아이템 기반 유사도 366
사전 가지치기(트리) 238–240
사후 가지치기(트리) 240–243
선형 회귀
  LWLR 201–205

최적선 찾기 194–200
선형대수학 426–433
표준 432
행렬 427–430
미분 432–433
역행렬 430–432
성능(k–평균 군집화 알고리즘) 268–270
성능 측정 182–187
성능 측정(정확도, 재현율, ROC 곡선) 180–187
성향/변화량 관계 215–217
센서 및 데이터 홍수 6–7
수신자 조작 특성 ROC 참조
수치형 값 정규화 37–39
순차 최소 최적화 SMO 참조
스마트 투표 데이터 소스 424–425
스마트폰 7
스칼라 연산 428
스크립트(mrjob 기본 구조) 398–402
스팸 메일 예제 95–99
검사 97–99
텍스트 토큰 만들기 95–97
시그모이드 함수(로지스틱 회귀 분류기) 108–110
실행 가능한 의사코드(파이썬으로) 16
심플 스토리지 서비스 S3 참조
싸이썬 18

ㅇ

아마존 399
아마존 웹 서비스 AWS 참조
아이템 기반 유사도와 사용자 기반 유사도 366
아이템 집합, 어프라이어리 알고리즘으로 찾기 288–296
관련 코드 292–296
연관 규칙 마이닝 296–301
후보 아이템 집합 생성 289–292
알고리즘 10–15, 22
SMO 최적화 134–142
선정 방법 12–13
야후! 위치 찾기 API, 지도상에서
지역점 군집화 예제 275–278
양분하는 k–평균 270–273
어빈 캘리포니아 대학교 439
어프라이어리 알고리즘 283–314
국회 투표 패턴 예제 301–312
국회 투표 데이터 집합 기록 302–310
연관 규칙 310–312
독버섯 속성 예제 312–313
빈발 아이템 집합 찾기 287–296
해당 코드 292–296
후보 아이템 집합 생성 289–292
빈발 아이템 집합으로 연관 규칙 마이닝하기 296–301
연관 분석 284–286
이론 286–288
어프라이어리 알고리즘으로 연관 분석 284–286
어프라이어리 알고리즘 이론 286–288
에이다부스트 메타 알고리즘 389–390
복잡한 데이터 집합 예제 179–182
분류기
데이터 집합의 다양한 표본 사용 165–167
에이다부스트 메타 알고리즘으로 검사 177–179
오류에 초점을 맞춘 개선 167–169
의사결정 스텀프 169–173
전체 구현 173–177
예측(수치형 값) 회귀 참조
유사도
아이템 기반과 사용자 기반 366
측정 362–366
의사결정 스텀프 169–173
의사결정 트리 48–77
구조 50–63
구축 59–63
데이터 집합 분할 56–59
정보 이득 52–56
매스플롯라이브러리 주석으로 플롯 63, 66–71
분류 72–73
콘택트렌즈 예측 예제 74–76
이미지 검사 벡터로 변환 43–44
인포침스 440
일래스틱 맵 리듀스 탭 391, 394

ㅈ

잠재적 의미 색인 (LSI 참조)
잠재적인 변수 342
재현율과 ROC 곡선 182–187
전문가 시스템 8
전문용어(기계 학습) 8–11
전방향 단계별 회귀 211–215
전복 나이 예제 205–206
전자정부기금 439
정규화(수치형 값) 37–39
정보 이득(의사결정 트리) 52–56
정확도와 ROC 곡선 182–187
조건부 확률과 나이브 베이스 분류기 81–83
좌표 축 이동 343–345
주요 구성요소 분석 PCA 참조
지도 지역점 군집화 274–281
지식 노동 7
지식 표현 11
지역 특색(개인 광고 예제) 99–105
지역적으로 사용되는 단어 표현 104–105
RSS 피드 불러오기 100–104
지역적 가중치가 부여된 선형 회귀 LWLR 참조
지역적 데이터 모델링 227–228

지역점 군집화, 지도 예제에서 274–281
지지 벡터 머신 SVMs 참조
직교 342–345

ㅊ

차원 축소 342–345
  기술 342–343
  반도체 제조 데이터 예제 348–352
초평면 344
최대 마진 찾기 131–134
  데이터 분리 129–131
최적선, 선형회귀로 찾기 194–200
최적화
  SMO 알고리즘 134–142
    적은 양의 데이터 집합 해결 135–142
    플랫 135, 143–151
  로지스틱 회귀 분류기 110–122
    가장 좋은 매개변수 찾기 112–115
    기울기 상승 사용 110–112
    의사결정 경계선 플롯 115–116
    확률적인 기울기 상승 사용 117–122
  분류기 관점에서 문제 구성 132–133
추천 시스템 356–357
추천 엔진
  레스토랑 예제 367–375
    SVD로 추천 개선 370–373
    과제 374–375
    맛보지 못한 음식 추천 367–370
  협력적 여과 기반 362–367
    유사도 166
    평가 366–367
축, 좌표 343–345
축소 계수(데이터 이해를 위해) 207–215

ㅋ

커널 151–158
  검사 155–158
  고차원의 데이터 매핑 151–152
  반지름 성향 함수 152–155
커널로 고차원 데이터 매핑 151–152
콘택트렌즈 예측 예제, 의사결정 트리 사용 74–76
콜론 연산자 422
클릭 스트림, 뉴스 사이트 마이닝 336–337

ㅌ

텍스트
  단어 벡터 85–87
  토큰 만들기 95–97
텍스트 분류
  나이브 베이스 분류기 85–94
    검사 91–94
    단어 벡터로 확률 계산 88–91
    텍스트로 단어 벡터 만들기 85–87
텍스트 토큰 만들기 95–97
텍스트 파일(데이터 구문 분석) 32–34
통계학자 8
트리 가지치기 238–243
  사전 가지치기 238–240
  사후 가지치기 240–243
트리 기반 회귀 226–258
  CART 알고리즘 232–237
    코드 실행 236–237
    트리 구축 233–236
  GUI(Tkinter 사용) 250–257
  모델 트리 243–246
  속성 228–232
  지역적 데이터 모델링 227–228
  트리 가지치기 238–243
    사전 가지치기 238–240
    사후 가지치기 240–243
  트리 메소드 비교(일반 회귀 예제) 246–250
트리 기반 회귀의 속성 228–232
트위터 피드(발생 단어 찾기) 331–335
특이 값 분해 SVD 참조
특이행렬 430

ㅍ

파이썬 커맨드 27
파이썬 트위터 모듈 337, 425
파이썬 파일 252, 331
파이썬을 사용하는 이유 16–18
파이썬의 단점 18
파이썬의 장점 17
패턴(국회 투표 예제) 301–312
  국회 투표 기록에 대한 데이터 집합 302–310
  연관 규칙 310–312
페가소스 알고리즘 403–404
표준 432
플랫 SMO 알고리즘 135, 143–151
필기체 인식 예제 42–46
  검사 벡터로 이미지 변환 43–44
  검사 44–46
필기체 인식 예제 다시 적용 159–162

ㅎ

하둡 명령어 394
하둡 작업 384–388
  AWS에서 실행 388–395
  평균과 분산 리듀서 분산처리
    리듀서 386–388
    매퍼 385–386
하위 클래스 17

함께 발생하는 단어, 트위터 피드에서 찾기 331–335
함수
  거리 측정 28–30
  반지름 성향 함수 152–155
함축 리스트 417–418
행렬 427–430
  미분 432–433
  역행렬 430–432
행렬 곱 428–430
행렬 유형 421
행렬 인수분해 358–359
행렬 차 428
협력적 여과 기반 추천 엔진 362–367
  유사도
    아이템 기반과 사용자 기반 366
    측정 362–366
  평가 366–367
확률 434–438
  결합 435–436
  규칙 436–438
  소개 434–435
확률적인 기울기 상승
  (로지스틱 회귀 분류기 최적화) 117–122
회귀 193–225
  데이터를 이해하기 위한 축소 계수 207–215
    regTreeEval( ) 248
    능형 회귀 207–211
    라소 기술 210–211
    전방향 단계별 회귀 211–215
  레고 가격 예제 217–224
  선형
    LWLR 201–205
    최적선 찾기 194–200
  성향/변화량 관계 215–217
  전복 나이 예제 205–206
훈련 예제 10
훈련 집합 10, 21