

# ÍNDICE

<b>Introducción</b>	<b>xvii</b>
<hr/>	
<b>Capítulo 1. Dar a los ordenadores el poder de aprender de los datos</b>	<b>23</b>
<hr/>	
<b>Crear máquinas inteligentes para transformar datos en conocimiento</b>	<b>24</b>
<b>Los tres tipos de aprendizaje automático</b>	<b>24</b>
Hacer predicciones sobre el futuro con el aprendizaje supervisado	25
Clasificación para predecir etiquetas de clase	25
Regresión para predecir resultados continuos	27
Resolver problemas interactivos con aprendizaje reforzado	28
Descubrir estructuras ocultas con el aprendizaje sin supervisión	29
Encontrar subgrupos con el agrupamiento	29
Reducción de dimensionalidad para comprimir datos	30
<b>Introducción a la terminología básica y las notaciones</b>	<b>30</b>
<b>Una hoja de ruta para crear sistemas de aprendizaje automático</b>	<b>33</b>
Preprocesamiento: Dar forma a los datos	34
Entrenar y seleccionar un modelo predictivo	34
Evaluar modelos y predecir instancias de datos no vistos	35
<b>Utilizar Python para el aprendizaje automático</b>	<b>36</b>
Instalar Python y sus paquetes desde Python Package Index	36
Utilizar la distribución y el gestor de paquetes Anaconda de Python	37
Paquetes para cálculo científico, ciencia de datos y aprendizaje automático	37
<b>Resumen</b>	<b>38</b>
<hr/>	
<b>Capítulo 2. Entrenar algoritmos simples de aprendizaje automático para clasificación</b>	<b>39</b>
<hr/>	
<b>Neuronas artificiales: un vistazo a los inicios del aprendizaje automático</b>	<b>40</b>
La definición formal de una neurona artificial	41
La regla de aprendizaje del perceptrón	43
<b>Implementar un algoritmo de aprendizaje de perceptrón en Python</b>	<b>46</b>
Una API perceptrón orientada a objetos	46
Entrenar un modelo de perceptrón en el conjunto de datos Iris	50
<b>Neuronas lineales adaptativas y la convergencia del aprendizaje</b>	<b>56</b>
Minimizar funciones de coste con el descenso de gradiente	57
Implementar Adaline en Python	60
Mejorar el descenso de gradiente mediante el escalado de características	64

## Índice

Aprendizaje automático a gran escala y descenso de gradiente estocástico	66
<b>Resumen</b>	<b>72</b>
<b>Capítulo 3. Un recorrido por los clasificadores de aprendizaje automático con scikit-learn</b>	<b>73</b>
<b>Elegir un algoritmo de clasificación</b>	<b>74</b>
<b>Primeros pasos con scikit-learn: entrenar un perceptrón</b>	<b>74</b>
<b>Modelar probabilidades de clase mediante regresión logística</b>	<b>81</b>
Intuición en regresión logística y probabilidades condicionales	81
Aprender los pesos de la función de coste logística	85
Convertir una implementación Adaline en un algoritmo para regresión logística	88
Entrenar un modelo de regresión logística con scikit-learn	93
Abordar el sobreajuste con la regularización	95
<b>Margen de clasificación máximo con máquinas de vectores de soporte</b>	<b>98</b>
Margen máximo de intuición	99
Tratar un caso separable no lineal con variables flexibles	101
Implementaciones alternativas en scikit-learn	103
<b>Resolver problemas no lineales con una SVM kernelizada</b>	<b>104</b>
Métodos kernel para datos inseparables lineales	104
El truco de kernel para encontrar hiperplanos separados en un espacio de mayor dimensionalidad	106
<b>Aprendizaje basado en árboles de decisión</b>	<b>110</b>
Maximizar la ganancia de información: sacar el mayor partido de tu inversión	112
Crear un árbol de decisión	117
Combinar árboles de decisión múltiples mediante bosques aleatorios	120
<b>K-vecinos más cercanos: un algoritmo de aprendizaje vago</b>	<b>123</b>
<b>Resumen</b>	<b>127</b>
<b>Capítulo 4. Generar buenos modelos de entrenamiento: preprocesamiento de datos</b>	<b>129</b>
<b>Tratar con datos ausentes</b>	<b>129</b>
Identificar datos ausentes en datos tabulares	130
Eliminar muestras o características con valores ausentes	131
Imputar valores ausentes	132
Entender la API de estimador de scikit-learn	133
<b>Trabajar con datos categóricos</b>	<b>134</b>
Características nominales y ordinales	134

Crear un conjunto de datos de ejemplo	135
Mapear características ordinales	135
Codificar etiquetas de clase	136
Realizar una codificación en caliente sobre características nominales	138
<b>Dividir un conjunto de datos en conjuntos de prueba y de entrenamiento individuales</b>	<b>140</b>
<b>Ajustar las características a la misma escala</b>	<b>142</b>
<b>Seleccionar características significativas</b>	<b>145</b>
Regularización L1 y L2 como penalizaciones contra la complejidad del modelo	146
Una interpretación geométrica de la regularización L2	146
Soluciones dispersas con la regularización L1	148
Algoritmos de selección de características secuenciales	152
<b>Evaluar la importancia de las características con bosques aleatorios</b>	<b>158</b>
<b>Resumen</b>	<b>161</b>
<b>Capítulo 5. Comprimir datos mediante la reducción de dimensionalidad</b>	<b>163</b>
<b>Reducción de dimensionalidad sin supervisión mediante el análisis de componentes principales</b>	<b>164</b>
Los pasos esenciales que se esconden detrás del análisis de componentes principales	164
Extraer los componentes principales paso a paso	166
Varianza total y explicada	169
Transformación de características	170
Análisis de componentes principales en scikit-learn	173
<b>Compresión de datos supervisada mediante análisis discriminante lineal</b>	<b>177</b>
Análisis de componentes principales frente a análisis discriminante lineal	177
Cómo funciona internamente el análisis discriminante lineal	178
Calcular las matrices de dispersión	179
<b>Seleccionar discriminantes lineales para el nuevo subespacio de características</b>	<b>182</b>
Proyectar muestras en el nuevo espacio de características	184
ADL con scikit-learn	185
<b>Utilizar el análisis de componentes principales con kernels para mapeos no lineales</b>	<b>187</b>

## Índice

Funciones kernel y el truco del kernel	188
Implementar un análisis de componentes principales con kernels en Python	194
Ejemplo 1: separar formas de media luna	195
Ejemplo 2: separar círculos concéntricos	198
Proyectar nuevos puntos de datos	201
Análisis de componentes principales con kernel en scikit-learn	205
<b>Resumen</b>	<b>206</b>
<b>Capítulo 6. Aprender las buenas prácticas para la evaluación de modelos y el ajuste de hiperparámetros</b>	<b>207</b>
<hr/>	
<b>Simplificar flujos de trabajo con <i>pipelines</i></b>	<b>207</b>
Cargar el conjunto de datos Cancer Wisconsin	208
Combinar transformadores y estimadores en un <i>pipeline</i>	209
<b>Utilizar la validación cruzada de K iteraciones para evaluar el rendimiento de un modelo</b>	<b>211</b>
El método de retención	212
Validación cruzada de k iteraciones	213
<b>Depurar algoritmos con curvas de validación y aprendizaje</b>	<b>217</b>
Diagnosticar problemas de sesgo y varianza con curvas de aprendizaje	218
Resolver el sobreajuste y el subajuste con curvas de validación	221
<b>Ajustar los modelos de aprendizaje automático con la búsqueda de cuadrículas</b>	<b>223</b>
Ajustar hiperparámetros con la búsqueda de cuadrículas	223
Selección de algoritmos con validación cruzada anidada	225
<b>Observar diferentes métricas de evaluación de rendimiento</b>	<b>227</b>
Leer una matriz de confusión	228
Optimizar la exactitud y la exhaustividad de un modelo de clasificación	229
Representar una característica operativa del receptor	232
Métricas de calificación para clasificaciones multiclase	235
<b>Tratar con el desequilibrio de clases</b>	<b>236</b>
<b>Resumen</b>	<b>239</b>
<b>Capítulo 7. Combinar diferentes modelos para el aprendizaje conjunto</b>	<b>241</b>
<hr/>	
<b>Aprender con conjuntos</b>	<b>241</b>
<b>Combinar clasificadores mediante el voto mayoritario</b>	<b>246</b>
Implementar un sencillo clasificador de voto mayoritario	246
Utilizar el principio de voto mayoritario para hacer predicciones	253

Evaluar y ajustar el clasificador conjunto	256
<b>Bagging: construir un conjunto de clasificadores a partir de muestras <i>bootstrap</i></b>	<b>262</b>
El <i>bagging</i> resumido	262
Aplicar el <i>bagging</i> para clasificar muestras en el conjunto de datos Wine	264
<b>Potenciar los clasificadores débiles con el <i>boosting</i> adaptado</b>	<b>268</b>
Cómo trabaja el <i>boosting</i>	268
Aplicar AdaBoost con scikit-learn	273
<b>Resumen</b>	<b>276</b>
<b>Capítulo 8. Aplicar el aprendizaje automático para el análisis de sentimiento</b>	<b>277</b>
<hr/>	
<b>Preparar los datos de críticas de cine de IMDb para el procesamiento de texto</b>	<b>278</b>
Obtener el conjunto de datos de críticas de cine	278
Preprocesar el conjunto de datos de películas en un formato adecuado	279
<b>Introducir el modelo «bolsa de palabras»</b>	<b>281</b>
Transformar palabras en vectores de características	281
Relevancia de las palabras mediante frecuencia de término–frecuencia inversa de documento	283
Limpiar datos textuales	286
Procesar documentos en componentes léxicos	288
<b>Entrenar un modelo de regresión logística para clasificación de documentos</b>	<b>290</b>
<b>Trabajar con datos más grandes: algoritmos <i>online</i> y aprendizaje <i>out-of-core</i></b>	<b>292</b>
<b>Modelado de temas con Latent Dirichlet Allocation</b>	<b>296</b>
Descomponer documentos de textos con LDA	297
LDA con scikit-learn	297
<b>Resumen</b>	<b>301</b>
<b>Capítulo 9. Incrustar un modelo de aprendizaje automático en una aplicación web</b>	<b>303</b>
<hr/>	
<b>Serializar estimadores de scikit-learn ajustados</b>	<b>304</b>
<b>Configurar una base de datos SQLite para el almacenamiento de datos</b>	<b>307</b>
<b>Desarrollar una aplicación web con Flask</b>	<b>309</b>
Nuestra primera aplicación web con Flask	310

## Índice

Validación y renderizado de formularios	312
Configurar la estructura del directorio	313
Implementar una macro mediante el motor de plantillas Jinja2	314
Añadir estilos con CSS	315
Crear la página resultante	316
<b>Convertir el clasificador de críticas de cine en una aplicación web</b>	<b>316</b>
Archivos y carpetas: observar el árbol de directorios	318
Implementar la aplicación principal como app.py	320
Preparar el formulario de críticas	322
Crear una plantilla de página de resultados	324
<b>Desplegar la aplicación web en un servidor público</b>	<b>326</b>
Crear una cuenta de PythonAnywhere	326
Cargar la aplicación del clasificador de películas	327
Actualizar el clasificador de películas	328
<b>Resumen</b>	<b>330</b>
<b>Capítulo 10. Predicción de variables de destino continuas con análisis de regresión</b>	<b>331</b>
<hr/>	
<b>Introducción a la regresión lineal</b>	<b>332</b>
Regresión lineal simple	332
Regresión lineal múltiple	333
<b>Explorar el conjunto de datos Housing</b>	<b>334</b>
Cargar el conjunto Housing en un marco de datos	335
Visualizar las características importantes de un conjunto de datos	336
Observar las relaciones mediante una matriz de correlación	338
<b>Implementar un modelo de regresión lineal de mínimos cuadrados ordinarios</b>	<b>341</b>
Resolver la regresión para parámetros de regresión con el descenso del gradiente	341
Estimar el coeficiente de un modelo de regresión con scikit-learn	346
<b>Ajustar un modelo de regresión robusto con RANSAC</b>	<b>347</b>
<b>Evaluar el rendimiento de los modelos de regresión lineal</b>	<b>350</b>
<b>Utilizar métodos regularizados para regresión</b>	<b>354</b>
<b>Convertir un modelo de regresión lineal en una curva: la regresión polinomial</b>	<b>356</b>
Añadir términos polinomiales con scikit-learn	356
Modelar relaciones no lineales en el conjunto de datos Housing	358
<b>Tratar con relaciones no lineales mediante bosques aleatorios</b>	<b>361</b>
Regresión de árbol de decisión	362
Regresión con bosques aleatorios	364
<b>Resumen</b>	<b>367</b>

<b>Capítulo 11. Trabajar con datos sin etiquetar: análisis de grupos</b>	<b>369</b>
<b>Agrupar objetos por semejanza con k-means</b>	<b>370</b>
Agrupamiento k-means con scikit-learn	370
Una manera más inteligente de colocar los centroides de los grupos iniciales con k-means++	375
Agrupamiento pesado frente a no pesado	376
Utilizar el método <i>elbow</i> para encontrar el número óptimo de grupos	379
Cuantificar la calidad del agrupamiento mediante gráficos de silueta	380
<b>Organizar agrupamientos como un árbol jerárquico</b>	<b>385</b>
Agrupar los grupos de manera ascendente	386
Realizar agrupamientos jerárquicos en una matriz de distancias	387
Adjuntar dendrogramas a un mapa de calor	391
Aplicar un agrupamiento aglomerativo con scikit-learn	393
<b>Ubicar regiones de alta densidad con DBSCAN</b>	<b>394</b>
<b>Resumen</b>	<b>400</b>
<b>Capítulo 12. Implementar una red neuronal artificial multicapa desde cero</b>	<b>401</b>
<b>Modelar funciones complejas con redes neuronales artificiales</b>	<b>402</b>
Resumen de una red neuronal de una capa	404
La arquitectura de red neuronal multicapa	406
Activar una red neuronal mediante la propagación hacia delante	409
<b>Clasificar dígitos manuscritos</b>	<b>411</b>
Obtener el conjunto de datos MNIST	412
Implementar un perceptrón multicapa	418
<b>Entrenar una red neuronal artificial</b>	<b>429</b>
Calcular la función de coste logística	430
Desarrollar tu intuición para la propagación hacia atrás	433
Entrenar redes neuronales mediante la propagación hacia atrás	434
<b>Sobre la convergencia en redes neuronales</b>	<b>439</b>
<b>Unas últimas palabras sobre la implementación de redes neuronales</b>	<b>440</b>
<b>Resumen</b>	<b>441</b>
<b>Capítulo 13. Paralelización de entrenamiento de redes neuronales con TensorFlow</b>	<b>443</b>
<b>TensorFlow y rendimiento de entrenamiento</b>	<b>444</b>
¿Qué es TensorFlow?	445
Cómo aprenderemos TensorFlow	446
Primeros pasos con TensorFlow	446

## Índice

Trabajar con estructuras de matriz	449
Desarrollar un modelo simple con la API de bajo nivel de TensorFlow	450
<b>Entrenar redes neuronales eficazmente con las API de alto nivel de TensorFlow</b>	<b>455</b>
Crear redes neuronales multicapa mediante el API Layers de TensorFlow	456
Desarrollar una red neuronal multicapa con Keras	460
<b>Elegir funciones de activación para redes multicapa</b>	<b>465</b>
Resumen de la función logística	466
Estimar probabilidades de clase en clasificaciones multiclase con softmax	468
Ampliar el espectro de salida con una tangente hiperbólica	469
Activación de la unidad lineal rectificadora	471
<b>Resumen</b>	<b>473</b>
<b>Capítulo 14. Ir más lejos: la mecánica de TensorFlow</b>	<b>475</b>
<b>Características clave de TensorFlow</b>	<b>476</b>
<b>Rango y tensores de TensorFlow</b>	<b>476</b>
Cómo obtener la dimensión y la forma de un tensor	477
<b>Entender los grafos computacionales de TensorFlow</b>	<b>478</b>
<b>Marcadores de posición en TensorFlow</b>	<b>481</b>
Definir marcadores de posición	481
Alimentar marcadores de posición con datos	482
Definir marcadores de posición para matrices de datos con diferentes tamaños de lote	483
<b>Variables en TensorFlow</b>	<b>484</b>
Definir variables	485
Inicializar variable	487
Alcance de la variable	488
Reutilizar variables	490
<b>Crear un modelo de regresión</b>	<b>493</b>
<b>Ejecutar objetos en un grafo de TensorFlow mediante sus nombres</b>	<b>497</b>
<b>Almacenar y restablecer un modelo en TensorFlow</b>	<b>498</b>
<b>Transformar tensores como matrices de datos multidimensionales</b>	<b>501</b>
<b>Utilizar la mecánica de control de flujo para crear grafos</b>	<b>505</b>
<b>Visualizar el grafo con TensorBoard</b>	<b>509</b>
Ampliar tu experiencia en TensorBoard	512
<b>Resumen</b>	<b>513</b>

<b>Capítulo 15. Clasificar imágenes con redes neuronales convolucionales profundas</b>	<b>515</b>
<b>Bloques de construcción de redes neuronales convolucionales</b>	<b>516</b>
Entender las CNN y conocer las jerarquías de características	516
Realizar convoluciones discretas	518
Realizar una convolución discreta en una dimensión	518
El efecto del relleno de ceros en una convolución	521
Determinar el tamaño de la salida de convolución	523
Realizar una convolución discreta en 2D	524
Submuestreo	528
<b>Juntarlo todo para crear una CNN</b>	<b>530</b>
Trabajar con entradas múltiples o canales de color	530
Regularizar una red neuronal con la eliminación	534
<b>Implementar una red neuronal convolucional profunda con TensorFlow</b>	<b>536</b>
La arquitectura de una CNN multicapa	536
Cargar y preprocesar los datos	538
Implementar una CNN en el API de TensorFlow de bajo nivel	539
Implementar una CNN en la API Layers de TensorFlow	552
<b>Resumen</b>	<b>558</b>
<b>Capítulo 16. Modelado de datos secuenciales mediante redes neuronales recurrentes</b>	<b>559</b>
<b>Introducir datos secuenciales</b>	<b>560</b>
Modelar datos secuenciales: el orden sí importa	560
Representar secuencias	561
Las diferentes categorías del modelado de secuencias	562
<b>RNN para modelar secuencias</b>	<b>563</b>
Entender la estructura y el flujo de una RNN	563
Calcular activaciones en una RNN	565
Los retos del aprendizaje de interacciones de largo alcance	568
Unidades de LSTM	570
<b>Implementar una RNN multicapa para modelar secuencias en TensorFlow</b>	<b>572</b>
<b>Proyecto uno: crear un análisis de sentimiento de las críticas de películas IMDb con RNN multicapa</b>	<b>573</b>
Preparar los datos	574
<i>Embedding</i>	578
Construir un modelo de RNN	580
El constructor de la clase SentimentRNN	581

## Índice

El método build	582
Paso 1: definir celdas RNN multicapa	584
Paso 2: definir los estados iniciales para las celdas RNN	584
Paso 3: crear la RNN utilizando las celdas RNN y sus estados	585
El método train	585
El método predict	587
Instanciar la clase SentimentRNN	587
Entrenar y optimizar el análisis de sentimiento de un modelo RNN	588
<b>Proyecto dos: implementar una RNN para el modelado de lenguaje a nivel de carácter en TensorFlow</b>	<b>589</b>
Preparar los datos	590
Construir un modelo RNN a nivel de carácter	594
El constructor	595
El método build	596
El método train	598
El método sample	600
Crear y definir el modelo CharRNN	601
El modelo CharRNN en el modo de muestreo	602
<b>Resumen del capítulo y del libro</b>	<b>602</b>
<b>Índice analítico</b>	<b>605</b>

---