Curso - Python orientado a Machine Learning - Parte 3: Introd. a ML



Erick Castillo / @ecastillo (slack IV Devs)



Indice

- Juegos previos
- Machine Learning, teoría
 - A very (very) brief introduction
 - Tipos de aprendizaje
 - Algoritmos
 - Machine Learning, Minería de Datos y KDD
 - Terminología
- Machine Learning, práctica
 - scikit-learn





Juegos previos

- lápices
- puntos en plano cartesiano
- playtennis





¿Qué es?

En términos muy simples:

- modelos que aprenden datos
- reconocimiento de patrones
- intersección entre estadística, IA y CS





Usos prácticos

- diagnósticos médicos
- detección de fraude en el uso de tarjetas de crédito
- análisis del mercado de valores, predicción de stock
- reconocimiento del habla y del lenguaje escrito
- segmentación de clientes en grupos con gustos similares
- y mucho + (the real "y mucho más")





Tipos de aprendizaje

- supervisado: clasificación, regresión
- no supervisado: agrupamiento
- refuerzo





Algoritmos

Algunos algoritmos populares:

- regresión lineal
- regresión logística
- kNN
- naive bayes
- SVM
- árboles de decisión
- random forest
- K-means
- redes neuronales
- algoritmos genéticos





Machine Learning, Minería de Datos y KDD

- preprocesamiento de los datos
 - OLTP y OLAP
 - Datamart
 - ETL
- procesamiento de los datos
 - la idea básica es entrenar el modelo con una porción de los datos (training dataset) y luego validarlo con el resto de los datos (test dataset).
- post-procesamiento de los datos
 - idealmente, los patrones descubiertos deben tener 3 cualidades:
 - 1 ser precisos
 - 2. comprensibles (inteligibles)
 - 3. interesantes (útiles y novedosos)





Terminología

- dataset
- características/atributos/features
- response/clase
- entrenamiento
- predicción
- evaluación
- subajuste
- sobreajuste
- generalización





scikit-learn

- "Simple and efficient tools for data mining and data analysis"
- Python 2 y 3
- muy bien documentada
- dependencias: numpy, scipy y matplotlib
- última versión estable a la fecha: 0.19.1
- sitio web oficial: http://scikit-learn.org





¿Qué trae?

- datasets
 - cada columna es un atributo
 - cada fila es una instancia
- métodos de preprocesamiento
- modelos de procesamiento
- métodos de postprocesamiento





Esquema de trabajo

Requerimientos para trabajar con datos en scikit-learn:

- features & response son objetos separados (ej: iris.data & iris.target)
- features & response deben ser numéricos
- features & response deben ser arrays Numpy (<class 'numpy.ndarray'>)
- features & response deben tener dimensiones específicas (ej: iris.data.shape = (150, 4), iris.target.shape = (150,))



Esquema de trabajo (cont.)

Una vez definido el problema:

- importar el modelo
 - from sklearn ...
- instanciar el modelo
 - modelo(parametros)
- ajustar el modelo instanciado (entrenar)
 - ocurre in_place
 - uso: modelo_instanciado.fit(X, y)
- aplicar el modelo a nuevos datos
 - retorna un array Numpy
 - supervisado:
 - modelo_instanciado.predict()
 - no supervisado:
 - modelo instanciado predict()
 - modelo_instanciado.transform()
- 🤰 evaluar el modelo (métricas)



Convenciones

- X: matriz de características
- y: vector de respuesta
- \bullet X_train, X_test, y_train, y_test



