

Curso ML con Python

Índice (notebooks)

T01-1-Data Cleaning-Carga de datos

- * Carga de datos con la función `read_csv()`
- * Carga de datos con la función `open()`
- * Lectura y escritura de ficheros
- * Leer datos desde una url externa
- * Generar una función propia
- * Ficheros XLS y XLSX

T01-2-Data Cleaning-Análisis preliminar

- * Resumen de los datos: dimensiones y estructura
- * Resumen de estadísticos básicos
- * Missing Values
- * Gestión de NAs
- * Variables Dummy

T01-3-Data Cleaning-Plots

- * Plots y visualización de datos
- * Scatter plot
- * Histogramas
- * Boxplot

T02-1-Data Cleaning-DataWrangling

- * Data Wrangling o Cirugía de datos
- * Crear un subconjunto de datos (con `set`, bucles, condiciones booleans, filtrado de filas y columnas)
- * Crear una nueva variable
- * Generación de números pseudo aleatorios
- * Shuffling
- * Choice
- * La semilla aleatoria

T02-2-Data Cleaning-Funciones de distribución de probabilidad

- * Distribuciones de probabilidad
- * Distribución uniforme
- * Distribución normal
- * Simulación de Monte Carlo
- * Dummy Data Sets

T02-3-DataCleaning-Agrupación de datos

- * Agregación por categoría
- * Agrupación de datos
- * Operaciones sobre datos agrupados
- * Filtrado de datos
- * Transformación de variables
- * Otras operaciones útiles para el agrupado de datos (`head`, `tail`, `nth`, `sort_values`, `groupby`)
- * Conjunto de entrenamiento y conjunto de testing
- * Dividir usando la distribución normal
- * Dividir con la librería `sklearn`

- * Dividir usando shuffle

T02-4-Data Cleaning-Concatenacion de datos

- * Concatenar y apedazar datos
- * Carga de datos distribuidos
- * Joins de datasets
- * Tipos de joins
- * Inner Join
- * Left Join
- * Right Join
- * Outer Join

T03-1-Statistics-Correlacion

- * Correlación de variables
- * Matriz de correlación y mapa de calor de correlación

T04-0-LinearRegression-Resumen

- * Resumen de la lección de Regresión lineal

T04-1-Linear Regresion-Datos ficticios

- * Modelo con datos simulados
- * Error del modelo
- * Estimación de parámetros mediante la técnica de mínimos cuadrados
- * Otras maneras de verificar la regresión lineal (estadísticos)

T04-2-LinearRegression-DemostracionSST

- * Demostración de $SST = SSR + SSD$

T04-3-LinearRegression-RegresionLinealSimple

- * Regresión lineal simple
- * El paquete statsmodel para regresión lineal
- * Regresión lineal múltiple
- * Regresión lineal múltiple con statsmodel
- * Multicolinealidad

T04-4-LinearRegression-Validación del modelo

- * Validación del modelo de regresión
- * Validación con el conjunto de testing

T04-5-LinearRegression-RegresiónSciKit-Learn

- * Regresión lineal con sklearn
- * Regresión lineal con variables categóricas
- * Eliminar variables dummies redundantes
- * Transformación de variables para conseguir relación lineal
- * Modelo de regresión cuadrático
- * Modelo de regresión lineal y cuadrático (combinado)
- * Problemas al implementar modelos de regresión lineal
- * Aspectos a tener en cuenta al implementar un modelo de regresión lineal (análisis residuos, varianza)

T05-0-LogisticRegression-Summary

- * Resumen de Regresión Logística

T05-1-LogisticRegression-Theory

- * Las matemáticas tras la regresión logística
- * Tablas de contingencia
- * Probabilidad condicional
- * Ratio de probabilidades
- * La regresión logística desde la regresión lineal
- * Regresión logística múltiple
- * Estimación con el modelo de la máxima verosimilitud
- * Método de Newton-Raphson
- * Validación cruzada
- * Matrices de confusión y curvas ROC para validar el modelo de Regresión Logística

T05-2-LogisticRegression-Implementacion

- * Definir la función de entorno
- * Calcular las probabilidades para cada observación
- * Calcular la matriz diagonal W
- * Obtener la solución de la regresión de la función logística
- * Comprobación experimental
- * Regresión logística con la librería statsmodel

T05-3-LogisticRegression-ImplementacionPython

- * Regresión logística para predicciones bancarias
- * EDA
- * Conversión de variables categóricas a variables dummies
- * Selección de rasgos para el modelo
- * Implementación del modelo con statsmodel.api
- * Validación del modelo logístico (con subconjuntos training y testing)
- * Validación cruzada
- * Matrices de confusión y curvas ROC para la validación

T06-0-Clustering-Theory

- * Los objetivos esenciales del clustering
- * Las características principales de una buena clusterización
- * La distancia
- * La matriz de distancias
- * Normalizar las distancias
- * Métodos de enlace
- * Clustering Jerárquico
- * Enlaces simple, completo, promedio, centroide, de Ward
- * Clustering con K-means
- * Ajustar los parámetros del clustering
- * Método del codo
- * Coeficiente de la silueta
- * Resumen del clustering

T06-1-Distancias

- * Profundización en distancias y enlaces
- * Clustering Jerárquico

T06-2-ClusterJerarquico

- * Representación gráfica de un dendrograma
- * Truncar el dendrograma

- * Dendrograma tuneado
- * Corte automático del dendrograma
- * Método del codo
- * Visualización final del clustering

T06-3-Clustering-ClusteringKMeans

- * Teoría

T06-4-ClusteringCompleto

- * Implementación con python
- * Normalización de los datos
- * Clustering jerárquico con sklearn
- * Clustering k-means con sklearn

T06-5-MetodoCodoSilueta

- * Función para iterar silueta
- * Representación del codo
- * Representación del codo normalizado

T06-6-Clustering-PropagacionAfinidad

- * Reporte sobre la propagación de la afinidad

T06-7-Clustering-KMedoids

- * Distribuciones en forma de anillo
- * Algoritmo con K-Means
- * Algoritmo de los K-medoides
- * Algoritmo de clustering espectral

T07-0-Trees-Theory

- * Entropía
- * Ganancia de información
- * Algoritmos para la generación de árboles
- * Algoritmo ID3
- * Otros algoritmos
- * Índice de Gini
- * Reducción de la varianza
- * La poda del árbol
- * Los problemas de los árboles de decisión
- * Variables continuas
- * Valores faltantes
- * Árboles de regresión
- * Random forest – Bosques aleatorios
- * Ventajas
- * Bolseo – bagging o bootstrap
- * Por qué funcionan los bosques aleatorios

T07-1-Tress-ImplementacionPython

- * Árboles de decisión con python
- * Visualización del árbol de decisión
- * Validación cruzada para la poda
- * Bosque aleatorio para clasificación

T07-2-Trees-RegressionTrees

- * Creación del árbol

- * Predicciones
- * Validación cruzada
- * Bosque aleatorio para regresión

T08-0-SVM-Teoría

- * Support Vector Machine
- * SVM para clasificación
- * Las matemáticas
- * Hiperplano que separa clases con margen débil
- * El truco del Kernel

T08-1-SVM-SVCLineal

- * Linear Support Vector Classifier
- * Representación del soporte vectorial en 2D

T08-2-SVMModel

- * Maximización del margen
- * Creación del modelo SVM
- * Representación de los vectores de soporte

T08-3-SVM-Kernels

- * Identificar fronteras no lineales
- * El kernel no lineal
- * Ajustar los parámetros del SVM

T08-5-SVM-RegressionVsClassification.ipynb

- * Clasificación de flores Iris
- * Grid Search Cross Validation
- * Visualizaciones interactivas para jugar con parámetros del modelo

T08-6-SVM-Regresión

- * Implementación simple en python
- * Ventajas e inconvenientes de los modelos de SVM

T09-0-KNN-Teoría

- * Teoría de K Nearest Neighbors

T09-1-KNN-Ejemplo

- * Importación y limpieza del data set
- * Clasificador de los K vecinos
- * Clasificación sin limpieza (de datos)
- * Clasificar nuevos datos

T09-2-KNN-Implementacion

- * Creando nuestro propio KNN
- * Aplicar nuestro KNN al data set de Cancer
- * Conclusiones sobre los algoritmos de ML

T09-3-KNN-Sistemas de recomendación

- * Carga de datos de Movie Lens
- * Análisis exploratorio de los ítems
- * Representación en forma matricial
- * Conjuntos de entrenamiento y validación

- * Filtro colaborativo basado en usuarios
- * Filtro colaborativo basado en los k usuarios más cercanos
- * Filtro colaborativo basado en Ítems
- * Filtro colaborativo basado en los k ítems más cercanos
- * Conclusiones

T10-0-ACP-Teoría

- * Análisis de Componentes Principales
- * La solución de Pearson
- * Cálculo de las Componentes Principales
- * Demostración

T10-1-ACP-PasoAPaso (implementación manual)

- * Análisis de Componentes Principales – Paso a paso
- * Representación de los datos con Plotly
- * Normalización de los datos
- 1.-Cálculo de la descomposición de valores y vectores propios
 - a)Usando la matriz de covarianzas
 - b)Usando la matriz de correlaciones
 - c)Singular Value Decomposition
- 2.-Las componentes principales
- 3.-Proyectar las variables en el nuevo subespacio vectorial

T10-2-ACP-SKlearn

- * ACP con Sklearn

T10-3-ACP-Plotly

- * Dibujar con Plotly
- * Scatter Plots sencillos
- * Gráficos combinados
- * Estilizado de gráficos con Plotly
- * Información al hacer Hover
- * Data Sets muy grandes

T11-0-RN-DeepLearningAprendizajeNoSupervisado

- * Deep Learning y Aprendizaje No Supervisado – Teoría

T11-1-RN-TensorFlow101

- * Introducción a TensorFlow

T11-2-RN-SeñalesTrafico (TensorFlow v. 1.X)

- * Aprendizaje neuronal de las señales de tráfico
- * Carga de datos
- * Análisis Exploratorio de Datos
- * Pre-procesamiento de los datos
- * Modelo de Red Neuronal con TensorFlow
- * Ejecución de la Red Neuronal
- * Evaluación de la Red Neuronal
- * Validación del modelo

T11-3-RNN-ReconocimientoRopa (TensorFlow v. 2.X)

- * El data set de MNIST
- * Importar TensorFlow
- * Carga de datos

- * Exploración de los datos
- * Procesamiento de los datos
- * Normalizar los datos
- * Construir el modelo
- * Preparar las capas
- * Compilar el modelo
- * Entrenar el modelo
- * Evaluar la precisión
- * Predicciones

T16-0-Python y R

- * Combinando R y Python
- * Acceder a python desde R
- * Trabajar de manera conjunta entre R y Python
- * Función mágica para R en notebook
- * Ejemplo complejo de R, Python y Rmagic