Entorno profesional y preparación de datos

Alex Martínez Martínez

valgrai.eu

Tabla de contenidos

0. Enlaces

https://github.com/almtav08/course-python-ml

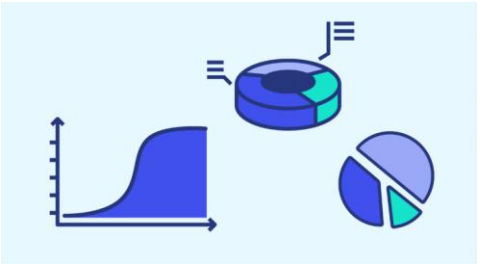
1. Entorno y Stack

Objetivos

Conocer el stack típico de ciencia de datos con Python.

Preparar un entorno reproducible y una estructura de proyecto.

Cargar un dataset y realizar un EDA inicial con gráficos sencillos.



1. Entorno y Stack

Librerías

Python

NumPy

pandas

matplotlib

seaborn

scikit learn

1. Entorno y Stack

Numpy

numpy-basics.py

```python
import numpy as np

# Crear un array y operar con él
arr = np.array([1, 2, 3, 4])

print(arr * 2)
# [2 4 6 8]

print(np.mean(arr))
# 2.5
```

- Fundamental para el cálculo numérico en Python.
- Ofrece estructuras de datos eficientes como los arrays multidimensionales y funciones matemáticas de alto rendimiento.

1. Entorno y Stack

Pandas

pandas-basics.py

```python
import pandas as pd

# Crear un DataFrame
data = {
  'Nombre': ['Ana', 'Luis', 'Marta'],
  'Edad': [23, 30, 27]
}
df = pd.DataFrame(data)

print(df[df['Edad'] > 28])
# Luis, 30
```

- Herramienta clave para la manipulación y análisis de datos.
- Proporciona estructuras como DataFrame y Series que permiten trabajar fácilmente con datos tabulares.

1. Entorno y Stack

Matplotlib

matplotlib-basics.py

```python
import matplotlib.pyplot as plt

# Graficar valores simples
x = [1, 2, 3, 4]
y = [2, 4, 6, 8]

plt.plot(x, y, marker='o')
plt.title("Gráfico simple")
plt.show()
```

- Biblioteca de visualización 2D muy flexible.
- Permite crear gráficos estáticos, animados e interactivos para explorar y comunicar datos.

1. Entorno y Stack

Seaborn

seaborn-basics.py

```python
import seaborn as sns
import matplotlib.pyplot as plt


# Dataset incluido en seaborn
tips = sns.load_dataset("tips")

# Diagrama de dispersión con ajuste de regresión
sns.lmplot(x="total_bill", y="tip", data=tips)
plt.show()
```

- Extensión de Matplotlib enfocada en la visualización estadística.
- Ofrece gráficos más atractivos y funciones simplificadas para análisis exploratorio.

1. Entorno y Stack

Scikit-learn

seaborn-basics.py

```python
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression

# Cargar dataset
X, y = load_iris(return_X_y=True)

# Separar en train/test
X_train, X_test, y_train, y_test

# Entrenar modelo
model = LogisticRegression()
model.fit(X_train, y_train)
```

- Biblioteca de machine learning.
- Incluye algoritmos para clasificación, regresión, clustering y herramientas para preprocesamiento, validación y selección de modelos.

2. Limpieza y Procesado

Objetivos

- Identificar y tratar valores nulos.
- Aplicar diferentes técnicas de imputación.
- Codificar variables categóricas.
- Escalar y normalizar variables numéricas.

2. Limpieza y Procesado

Valores Nulos

seaborn-basics.py

```
# Imports

df = pd.read_csv('data.csv')

# Matriz booleana indicando
# si el valor es nulo o no
df.isna()

# Número de nulos para cada
# columna
df.isna().sum()
```

- Un valor nulo representa la ausencia de un dato. Es decir, indica que no hay ningún valor registrado para esa posición o variable.
- No es lo mismo que un cero o una cadena vacía.

2. Limpieza y Procesado

Imputación

- La imputación es el proceso de rellenar los valores nulos en un conjunto de datos con valores estimados.
- Variables categóricas y numéricas no pueden seguir la misma estrategia.
- Ejemplos de imputación:
  - Media: Se reemplazan los valores nulos por la media de la columna. Es simple y rápido.
  - Mediana: Se usan los valores de la mediana de la columna. Es más robusto.
  - KNNImputer: Utiliza los k vecinos más cercanos de un registro para estimar los valores faltantes, basándose en similitudes entre filas.

Apuntes Stella:

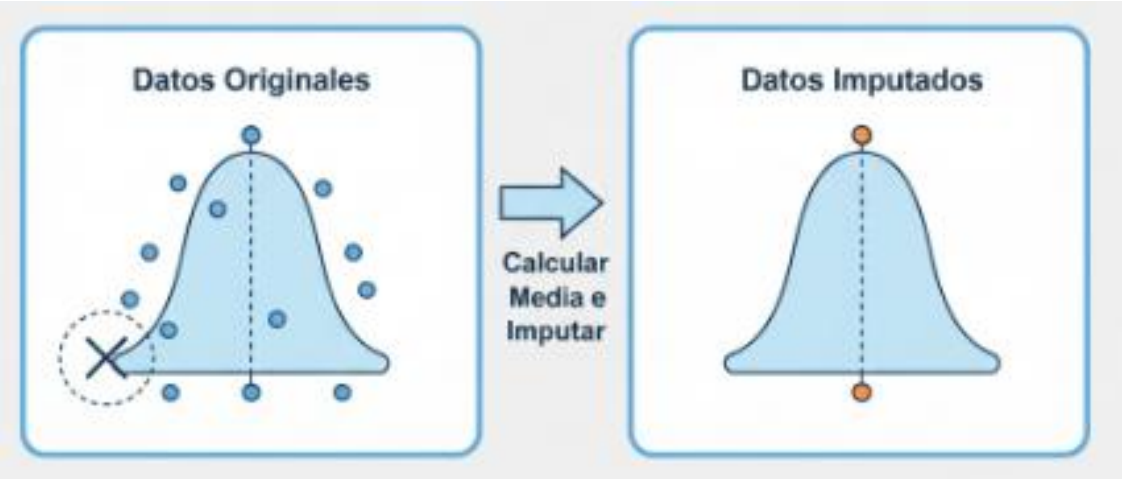
¿Cuál es la mejor estrategia de imputación?

Imputación por Media

Se usa para **variables numéricas** con una distribución aproximadamente normal.

•**Ventajas:** Es simple y rápida de implementar.

•**Desventajas:** Es muy sensible a valores atípicos (outliers) y puede reducir la varianza real de los datos.



***Forma de saber la distribución**

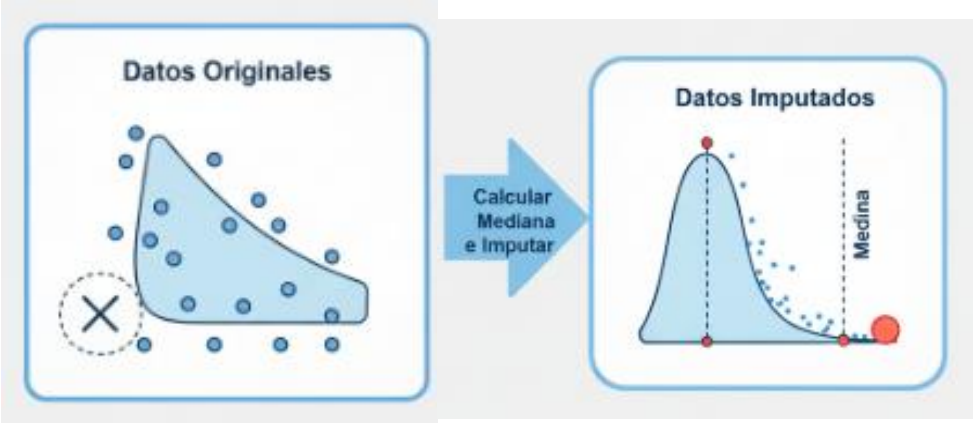
```
# Crea el histograma
sns.histplot(data_normal, kde=True)
plt.title('Histograma de Distribución de Datos’)
plt.xlabel('Valor’)
plt.ylabel('Frecuencia’)
plt.show()
```

Imputación por Mediana

Es ideal para **variables numéricas** que tienen **outliers** o una distribución asimétrica.

•**Ventajas:** Es más robusta que la media frente a valores extremos.

•**Desventajas:** Puede no ser un valor representativo si la distribución de datos es multimodal.

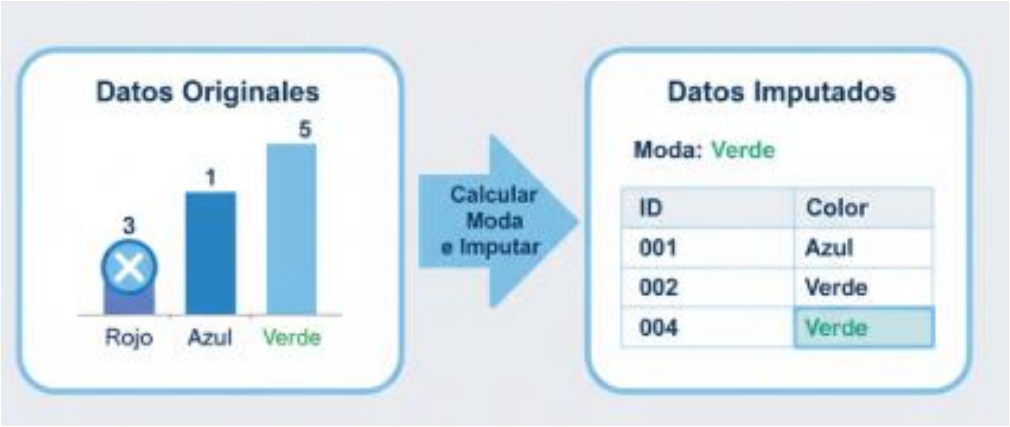


Imputación por Moda

Este método se aplica principalmente a **variables categóricas** o numéricas discretas.

•**Ventajas:** Mantiene la naturaleza de las variables categóricas.

•**Desventajas:** Puede no ser útil si los datos tienen varias modas o si la distribución es uniforme.

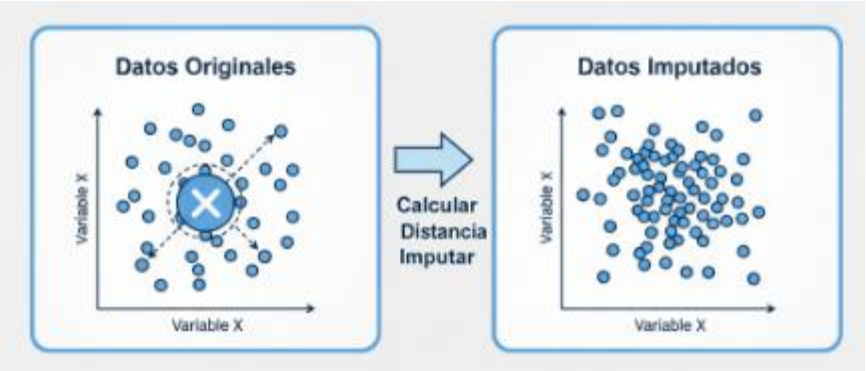


KNNImputer (K-Nearest Neighbors Imputation)

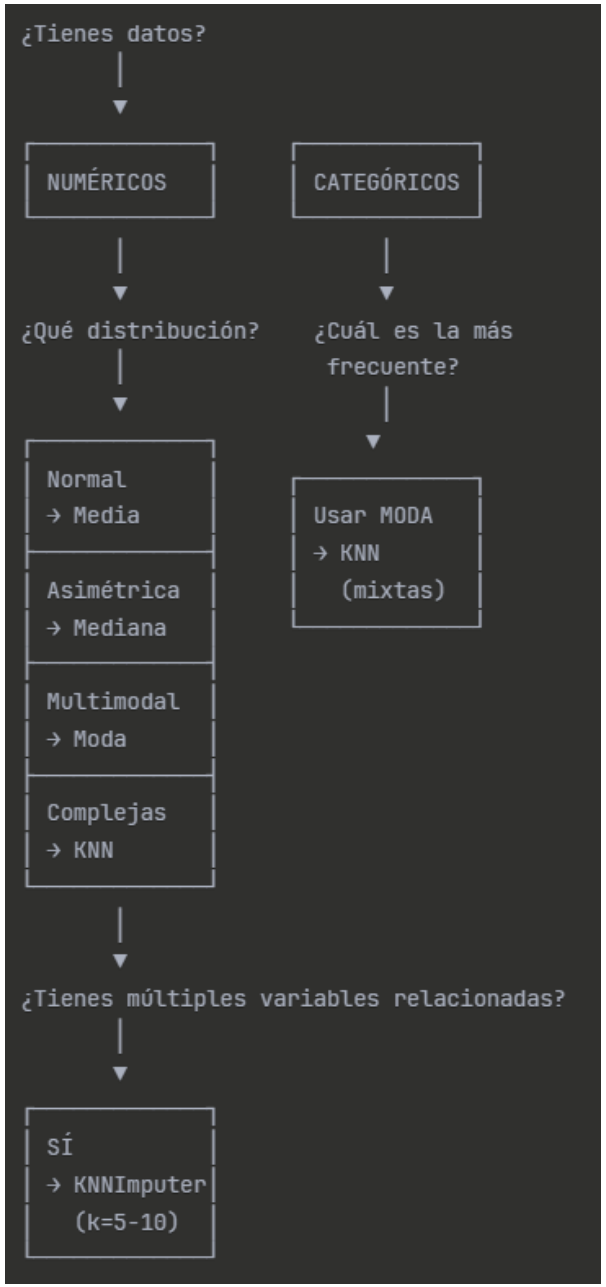
Se utiliza cuando hay **patrones complejos** y relaciones entre las variables.

•**Ventajas:** Es más preciso, ya que considera las relaciones entre las variables correlacionadas para estimar los valores faltantes.

•**Desventajas:** Es un método **computacionalmente costoso** y muy sensible a la escala de los datos, por lo que es necesario normalizar las variables previamente.



Guía de decisión



2. Limpieza y Procesado

Codificación Categórica

- La codificación categórica es el proceso de convertir variables categóricas en valores numéricos que los modelos puedan entender.

color → (rojo, azul, verde)

ONE HOT ENCODING	ORDINAL ENCODING
rojo → [1., 0., 0.]	rojo → 0
azul → [0., 1., 0.]	azul → 1
verde → [0., 0., 1.]	verde → 2

2. Limpieza y Procesado

Escalado y normalización

- El escalado y la normalización son técnicas para transformar los datos numéricos y que todos estén en rangos comparables, lo cual mejora el rendimiento de muchos modelos.

nota media → [0, 10]

suma de notas → [0, 10000] Influirá mucho más que nota media.

- Escalado: Ajusta los datos para que entren en un rango específico, por ejemplo entre 0 y 1.
- Normalización: Ajusta los datos para que la magnitud del vector de características sea 1.

Apuntes Stella:

Codificación de variables categóricas

3. Pipelines

- Una pipeline en scikit-learn es una forma de encadenar varios pasos de procesamiento y un modelo en un solo objeto que se puede entrenar y usar para hacer predicciones de manera ordenada y reproducible.
- Permite automatizar tareas como:
  - Preprocesamiento (imputación de nulos, codificación categórica, escalado, normalización, etc.).
  - Entrenamiento del modelo (regresión, clasificación, clustering...).
- Ventajas:
  - Evita errores al aplicar transformaciones a los datos de entrenamiento y test de manera diferente.
  - Hace que el código sea más limpio y fácil de reproducir.

4. Ingeniería de características

- La ingeniería de características es el proceso de crear, transformar o seleccionar variables de un dataset para que los modelos puedan aprender mejor.
- Su objetivo es extraer la información más relevante de los datos y representarla de forma que facilite la predicción.
- Algunas técnicas comunes:
  - Transformaciones: logaritmos, raíces, escalado, normalización.
  - Combinación de variables: sumar, restar, multiplicar columnas para generar nuevas características.
  - Codificación de variables categóricas: label encoding, one-hot encoding.
  - Extracción de información de fechas o textos: día de la semana, mes, longitud del texto, frecuencia de palabras.
  - Selección de características: elegir solo las más relevantes para el modelo.

5. Taller

¡Para el resto de la clase de hoy es hora de probar todo lo que hemos visto hasta ahora!

https://github.com/almtav08/course-python-ml

valgrai.eu

Valencian Graduate School and Research Network of Artificial Intelligence