guayerd

Fundamentos IA

Análisis con Python

Clase 6

En colaboración con

IBM **SkillsBuild**

¡Bienvenidos!

¿Nos presentamos?

- ¿Qué recuerdan de la clase anterior?
- ¿Qué esperan aprender?
- ¿Tienen alguna pregunta?

guayerd

En colaboración con

IBM **SkillsBuild**

Contenidos

Por temas

05
- Copilot Chat y prompts
- Demo asincrónica

06
- Limpieza y transformación

07
- Estadística aplicada

08
- Visualización

guayerd

En colaboración con

IBM **SkillsBuild**

Objetivos de la clase

- Pandas
- Lectura de archivos
- Estructuras principales
- Inspección y limpieza

guayerd

En colaboración con

IBM **SkillsBuild**

Análisis con Python

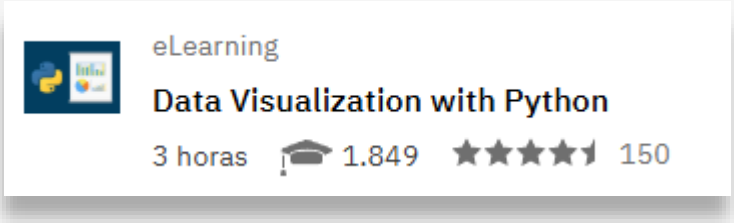
Limpieza y transformación

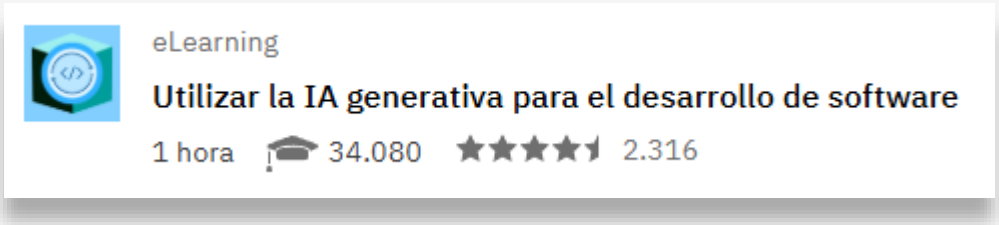
guayerd

En colaboración con

IBM **SkillsBuild**

Plataforma Skill Build: Python





guayerd

En colaboración con

IBM **SkillsBuild**

Limpieza y transformación de datos

Etapa 3 del ciclo de vida del dato

Proceso técnico para **preparar datos antes del análisis**.

- Elimina errores, inconsistencias y valores irrelevantes
- Estandariza formatos y estructuras
- Mejora la calidad y utilidad de los datos

Estandarizar = criterios

Formatos comunes

- **CSV:** datos tabulares separados por comas
- **JSON:** objetos anidados y flexibles
- **Excel:** hojas de cálculo
- **Bases de datos:** estructuras relacionales

Pandas integra el trabajo con todos estos formatos

Pandas
(Panel Data)

Librería basada en NumPy para **análisis y manipulación de datos estructurados.**

- Lectura de archivos CSV, JSON, Excel
- Filtrado, agrupación y ordenamiento
- Transformaciones con DataFrames y Series

Comandos

- **Instalación:** pip install pandas
- **Uso:** import pandas as pd

Estructuras principales

Series (.s)

- Arreglo unidimensional
- Compuesta por índices y valores
- Similar a una columna

DataFrame (.df)

- Tabla bidimensional
- Colección de Series alineadas por índice
- Base para análisis y transformación

Lectura de archivos

CSV

- df_csv = pd.read_csv("nombre_archivo.csv")
- Útil: encoding='utf-8', sep=';'

JSON

- df_json = pd.read_json("nombre_archivo.json")
- Para listas: orient='records'

Excel

- df_xls = pd.read_excel("nombre_archivo.xlsx")
- Especificar hoja: sheet_name="nombre_hoja"

Ejemplos comunes de encoding

Encoding	Características	Uso típico
UTF-8	Universal, soporta todos los idiomas	Web, Python, archivos modernos
latin1	Limitado a caracteres occidentales	Archivos antiguos, Excel en español
ISO-8859-1	Similar a latin1	Sistemas europeos
utf-8-sig	UTF-8 con marca BOM	Archivos exportados desde Excel

Siempre devuelve un DataFrame listo para trabajar

Inspección inicial

Aspecto	Comando	Descripción
Estructura	df.shape	Dimensiones del conjunto
Tipos	df.dtypes	Variables numéricas y categóricas
Completitud	df.isnull().sum()	Valores faltantes
Muestra	df.head()	Primeros registros
Resumen	df.info()	Información general

Desafíos comunes en datos

- **Valores faltantes:** celdas vacías o NaN
- **Duplicados:** registros repetidos exactos
- **Inconsistencias:** formatos diferentes para mismo dato
- **Valores atípicos:** datos extremos que sesgan análisis
- **Tipos incorrectos:** números como texto, fechas mal formateadas

¿Qué harías en cada situación?

- Dataset con 10% de valores faltantes distribuidos aleatoriamente
- Registros de clientes con emails duplicados pero datos diferentes
- Precios con valores negativos en sistema de inventario
- Fechas en formatos: "2024-01-15", "15/01/2024", "Jan 15, 2024"

Tratamiento de valores faltantes

Estrategia	Comando(s)	Descripción
Detección	df.isnull(), df.isna()	Identificar valores faltantes
Eliminación	df.dropna()	Eliminar registros con pocos casos
Valor constante	df.fillna(0)	Rellenar con cero o texto
Valor estadístico	df.fillna(df['columna'].median())	Rellenar con promedio o mediana

Eliminación de duplicados

Estrategia	Comando(s)	Descripción
Detección	df.duplicated()	Marca filas duplicadas
Eliminación completa	df.drop_duplicates()	Remueve todos los duplicados
Por columnas específicas	df.drop_duplicates(subset=['col1', 'col2'])	Detecta duplicados según columnas elegidas
Conservar primera/última	Parámetro keep	Mantiene el primer o último registro

Inconsistencias de formato

Tipo de dato	Comando(s) / Acción	Descripción
Texto	.str.lower(), .str.strip()	Normalizar mayúsculas/minúsculas y espacios
Fechas	pd.to_datetime()	Convertir a formato fecha uniforme
Números	pd.to_numeric()	Ajustar separadores decimales
Categorías	—	Estandarizar variaciones del mismo valor

Manejo de valores atípicos

Valores extremos que se alejan significativamente del resto de los datos.

- **Detección visual:** boxplots muestran valores extremos
- **Filtrado por rango:** remover valores fuera de límites lógicos
- **Criterio de dominio:** usar conocimiento del área
- **Verificación manual:** confirmar si son errores o valores reales

Tipos de datos incorrectos

Estrategia	Comando(s)	Descripción
Verificar tipos	df.dtypes	Identificar tipos actuales
Conversión manual	df.astype()	Cambiar tipo de columna
Fechas	pd.to_datetime()	Convertir a formato fecha
Numéricos	pd.to_numeric()	Forzar conversión numérica

Transformaciones básicas

Operación	Comando(s)	Descripción
Filtrado	df[condición]	Seleccionar subconjuntos específicos
Agrupación	df.groupby()	Calcular estadísticas por categorías
Ordenamiento	df.sort_values()	Organizar registros por columnas
Selección	df[['col1', 'col2']]	Elegir columnas específicas

Café de barrio

1. Calcular correlación entre temperatura y ventas
2. Identificar el mes con mejor retorno publicitario
3. Analizar relación personal vs satisfacción cliente
4. Proponer estrategia basada en datos

Mes	Ventas ($)	Temp (°C)	Publicidad ($)	Personal	Satisfacción
Ene	15,000	18	800	4	4.2
Feb	22,000	25	1,200	5	4.5
Mar	18,000	22	900	4	4.1
Abr	28,000	28	1,500	6	4.8
May	25,000	30	1,300	5	4.6

Proyecto

Tienda Aurelion

- **Documentación:** notebook Markdown
- **Desarrollo técnico:** programa Python
- **Visualización de datos:** dashboard en Power BI
- **Presentación oral:** problema, solución y hallazgos

guayerd

En colaboración con

IBM **SkillsBuild**

Limpieza de datos

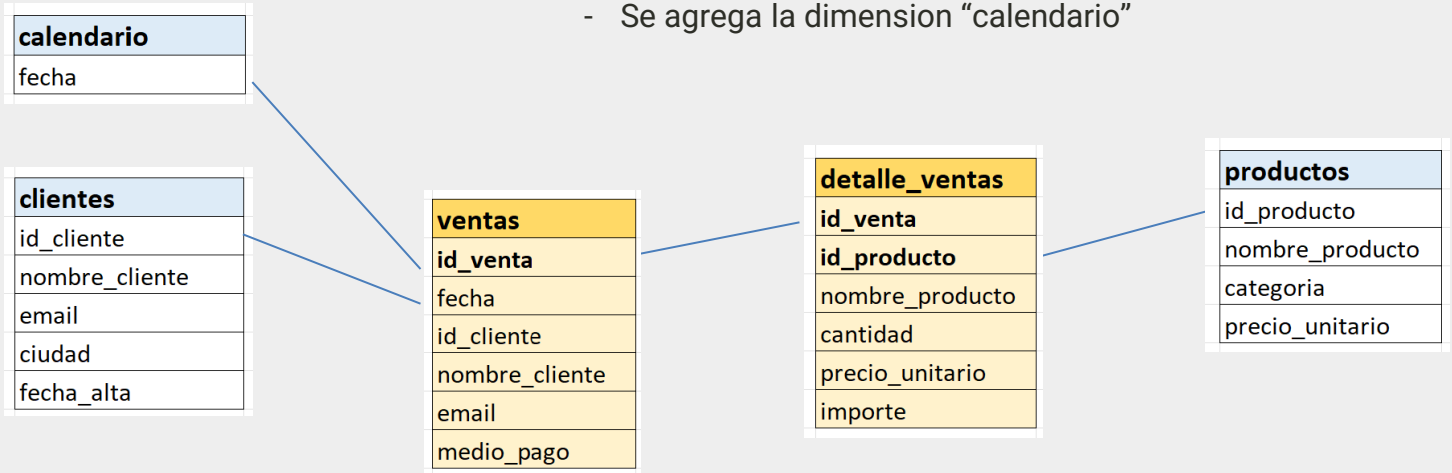
Trabajo en equipo

1. Usar **Copilot para analizar problemas** con dataset
2. **Limpiar la base** de datos
3. **Documentar con Copilot** cada paso aplicado

Modelo de datos

Copo de Nieve

- A partir de los archivos disponibles
- No esta normalizado
- Se agrega la dimension “calendario”



Modelo de datos

Preparacion para ML

Es necesario pasar del modelo relacional inicial a un dataframe adecuado para trabajar con un modelo de ML

DataFrame
id_hecho
id_venta
fecha
año
mes
día
id_cliente
nombre_cliente
ciudad
email
fecha_alta
id_producto
nombre_producto
categoria
cantidad
precio_unitario
importe
medio_pago

Este procedimiento require verificaciones de integridad y Calidad de los datos

Retro

¿Cómo nos vamos?

- ¿Qué fue lo más útil de la clase?
- ¿Qué parte te costó más?
- ¿Qué te gustaría repasar o reforzar?

guayerd

En colaboración con

IBM **SkillsBuild**