Data Scientist Senior Toolkit

Roadmap, Curación y Feature Engineering en Pandas

Curso: Fundamentos en Ciencia de Datos(Maestría) | **Periodo:** 2026-1
Docente: Jorge Iván Padilla-Buriticá | *Universidad EAFIT*

Filosofía de la Excelencia en Datos

Un Consultor Senior entiende que el algoritmo es solo el 10 % del éxito. El 90 % restante reside en la calidad del dato. Documentar cada transformación no es burocracia, es **trazabilidad**. Si un dashboard no es auditable, su recomendación no tiene valor de negocio.

1 Roadmap: Pasos previos a la Visualización

Antes de construir cualquier gráfico o dashboard, el científico de datos debe resolver el *Lineage* y la *Integridad*.

1. **Comprensión del Negocio:** ¿Qué KPI estamos moviendo? (Ej: Churn rate, Rentabilidad por SKU).
2. **Perfilamiento (Profiling):** Auditoría de tipos de datos, nulos y duplicados.
3. **Limpieza (Cleaning):** Estandarización de texto, manejo de nulos y outliers.
4. **Ingeniería (Feature Engineering):** Creación de variables que capturen la esencia del negocio.

2 Pandas Mastery: 20 Funciones Fundamentales

2.1 Auditoría e Identificación

1. isna() & sum()

```
# Identifica nulos por columna. Paso 1 de cualquier EDA.
df.isna().sum()
```

2. info()

```
# Muestra tipos de datos y uso de memoria. Vital para optimizar tipos.
df.info()
```

3. describe()

```
# Estadisticos rapidos. Ayuda a detectar ceros donde no deberia haber.
df.describe()
```

4. nunique()

```python
# Cuenta valores unicos. Sirve para detectar variables constantes.
df.nunique()
```

5. value_counts()

```python
# Frecuencia de categorias. Detecta desbalanceo de clases.
df['categoria'].value_counts(normalize=True)
```

2.2 Limpieza y Corrección

ALERTA TÉCNICA: Efectos del Cero

En campos como 'Presión Arterial' o 'Costo Unitario', el valor 0 es un nulo técnico. No lo dejes así; cámbialo a NaN antes de imputar.

6. replace()

```python
import numpy as np
df['variable'] = df['variable'].replace(0, np.nan)
```

7. dropna()

```python
# Elimina filas con nulos criticos (ej. en la variable objetivo)
df.dropna(subset=['target_y'], inplace=True)
```

8. fillna()

```python
# Imputacion rapida. Se recomienda usar la mediana para evitar
    outliers.
df['edad'].fillna(df['edad'].median(), inplace=True)
```

9. str.strip() & lower()

```python
# Limpieza de strings para evitar duplicados por espacios o mayusculas
    .
df['nombre'] = df['nombre'].str.strip().str.lower()
```

10. to_datetime()

```python
# Conversion de fechas con manejo de errores (coerce).
df['fecha'] = pd.to_datetime(df['fecha'], errors='coerce')
```

2.3 Transformación y Feature Engineering

11. astype()

```python
# Casting de tipos para ahorrar memoria (ej: float64 a float32).
df['id'] = df['id'].astype(str)
```

12. get_dummies()

```python
# One-Hot Encoding: Convierte categorias en variables numericas (0/1).
df_enc = pd.get_dummies(df, columns=['ciudad'], drop_first=True)
```

13. apply()

```python
# Aplicar funciones personalizadas fila a fila o columna a columna.
df['rango_edad'] = df['edad'].apply(lambda x: 'Adulto' if x >= 18 else
    'Menor')
```

14. clip()

```python
# Acota outliers (Winsorizacion) sin eliminar la muestra.
df['salario'] = df['salario'].clip(lower=1000, upper=50000)
```

15. map()

```python
# Sustitucion de valores basada en diccionarios (limpieza de
    categorias).
mapping = {'Med': 'Medellin', 'MDE': 'Medellin'}
df['ciudad'] = df['ciudad'].map(mapping).fillna(df['ciudad'])
```

16. groupby().transform()

```python
# Normalizacion por grupos (ej: restar la media del departamento).
df['rel_precio'] = df['precio'] - df.groupby('cat')['precio'].
    transform('mean')
```

17. drop_duplicates()

```python
# Elimina filas identicas garantizando integridad.
df.drop_duplicates(subset=['id_transaccion'], keep='first', inplace=
    True)
```

18. melt()

```python
# Pasa datos de formato "Ancho" a "Largo" (Tidy Data).
df_long = df.melt(id_vars=['ID'], value_vars=['2024', '2025'])
```

19. merge()

```python
# Une tablas. El 'left' garantiza que no pierdas ventas sin SKU asociado.
df_final = df_ventas.merge(df_productos, on='SKU', how='left')
```

20. nlargest()

```python
# Obtiene los N registros mas altos sin ordenar todo el dataset.
df.nlargest(10, 'ventas')
```

3 Guía Técnica de Visualización Estadística

Seleccionar el gráfico incorrecto puede llevar a interpretaciones erróneas.

Tipo de Variable	Gráfico Sugerido	Propósito Estadístico
Numérica Continua	**Histograma / KDE**	Ver la forma de la distribución (Normal vs Sesgada).
Numérica Continua	**Boxplot**	Identificar Outliers y el Rango Intercuartílico (IQR).
Categórica	**Gráfico de Barras**	Comparar frecuencias o magnitudes entre grupos.
Num vs Num	**Scatter Plot**	Analizar correlación y densidad de puntos.
Num vs Tiempo	**Gráfico de Líneas**	Identificar tendencias, ciclos y estacionalidad.
Categórica vs Num	**Violin Plot**	Ver la distribución numérica dentro de cada categoría.

ALERTA TÉCNICA: Sobreajuste Visual

Evite los gráficos de torta (Pie charts) cuando tenga más de 3 categorías. El ojo humano tiene dificultades para comparar áreas; prefiera siempre gráficos de barras horizontales.

4 Resumen de Imputación Senior

Método	Escenario	Efecto Secuencial
Media	Distribución Normal	Mantiene el promedio, reduce la varianza.
Mediana	Distribución con Outliers	Robusta ante valores extremos.
Moda	Variables Categóricas	Mantiene la consistencia de clases.
Forward Fill	Series de Tiempo	Asume que el valor actual es igual al anterior.

"Un dashboard es la punta del iceberg; la ingeniería de datos es lo que lo mantiene a flote."