Examen Técnico: Data Scientist

Duración: 4 horas
Instrucciones:

- Responde cada sección siguiendo las instrucciones proporcionadas.
- Crea un repositorio **Git público** y haz commit regularmente del progreso.
- Usa **Python 3**, **SQL**, **Pandas**, y herramientas relacionadas según corresponda.
- Documenta claramente tus pasos, decisiones y supuestos en cada ejercicio.
- Incluye **pruebas unitarias** y scripts modularizados en Python.

Ejercicio 0: Generación de Datos Sintéticos

Tareas

1. Escribe un script en Python que genere un conjunto de datos con al menos **50,000 filas** siguiendo este esquema:

```
{
  "order_id": "uuid",
  "customer_id": "random_int(1, 10_000)",
  "product_id": "random_int(1, 1_000)",
  "quantity": "random_int(1, 20)",
  "price": "random_float(1.0, 500.0)",
  "discount": "random_float(0.0, 0.3)",
  "order_date": "random_date(2023-01-01, 2024-12-31)",
  "shipping_priority": "random_choice(['Low', 'Medium',
'High'])",
  "region": "random_choice(['North', 'South', 'East', 'West'])"
}
```

- Asegúrate de que:
  - `order_id` sea único.
  - `order_date` esté distribuido con un patrón de estacionalidad y tendencia creciente (más órdenes en 2024).
  - `discount` esté correlacionado inversamente con `price`.
  - `shipping_priority` sea proporcional a `region` (por ejemplo, más alta prioridad en "North").

2. Introduce ruido y valores faltantes en un **5% de las filas** siguiendo estos criterios:
    - En al menos **tres columnas aleatorias por fila**.
    - Opciones para ruido: eliminar valores, introducir cadenas como `"NULL"`, o números extremos (ej. `-9999`).
3. Guarda el dataset generado en `raw_sales_data.csv`.

Ejercicio 1: Procesamiento de Datos

Usando el archivo `raw_sales_data.csv`:

Tareas

1. Propón y describe brevemente una estrategia para manejar los datos faltantes, considerando tanto la imputación como la eliminación.
2. Limpia y procesa los datos:
    - Identifica valores inválidos o fuera de rango.
    - Realiza imputaciones dinámicas según patrones detectados (por ejemplo, completar `price` basado en el promedio de productos similares).
3. Calcula:
    - **Ingreso total por cliente** considerando descuentos.
    - **Producto más vendido por región** y **ingreso total generado por cada región**.
    - **Distribución de prioridad de envío por región**.
4. Guarda el dataset limpio y procesado en `cleaned_sales_data.csv`.

Ejercicio 2: Análisis con SQL

Dado el esquema de base de datos (puedes usar `sqlite3`):

- **Tabla: customers**
    - `customer_id` (PK)
    - `name`
    - `email`
    - `region`
- **Tabla: orders**
    - `order_id` (PK)
    - `customer_id` (FK)
    - `order_date`
    - `shipping_priority`

- **Tabla: order_details**
  - `order_detail_id` (PK)
  - `order_id` (FK)
  - `product_id`
  - `quantity`
  - `price`
  - `discount`

Tareas

1. Escribe consultas para:
   - Calcular el **ingreso total por cliente**, ordenado de mayor a menor.
   - Encontrar el **producto más vendido en cada región** considerando el volumen total (`quantity * price`).
   - Calcular el **ingreso promedio por cliente y región** para cada mes.
   - Identificar a los **top 5 clientes con más ingresos generados** en el último año, junto con el número de órdenes realizadas.
2. Documenta cualquier optimización aplicada a las consultas (uso de índices, subconsultas, etc.).

Ejercicio 3: Visualización y Reportes

Usando los datos limpios de `cleaned_sales_data.csv`:

1. Genera un reporte con:
   - **Ingresos mensuales totales por región**.
   - **Top 10 productos con mayores ingresos** (por precio y cantidad).
   - **Relación entre prioridad de envío y descuento aplicado**.
2. Visualiza:
   - **Gráfica de barras** para ingresos mensuales por región.
   - **Mapa de calor** que muestre la correlación entre `quantity`, `price`, y `discount`.
3. Describe brevemente cualquier patrón detectado (ejemplo: estacionalidad, diferencias regionales).

Ejercicio 4: Modelado Predictivo

Usando `cleaned_sales_data.csv`:

1. Realiza un **análisis exploratorio avanzado**:
    - Identifica estacionalidad y tendencias.
    - Detecta outliers y evalúa su impacto en las ventas.
2. Define una estrategia para predecir el **ingreso diario total**:
    - Divide los datos en entrenamiento (80%) y prueba (20%).
    - Implementa modelos:
        - **Regresión lineal múltiple** considerando `quantity`, `price`, `discount`, y `region`.
        - **Modelo basado en Random Forest o Gradient Boosting** para capturar relaciones no lineales.
    - Opcional: Usa técnicas de feature engineering, como generación de variables temporales.
3. Evalúa los modelos usando:
    - **MAE**, **R^2**, y análisis de residuales.
    - Describe el rendimiento y si el modelo es aplicable en un entorno real.