REPORTE DE ANÁLISIS EXPLORATORIO

Alumno: Magaña Celis David

1. INTRODUCCIÓN

El presente reporte documenta el análisis exploratorio realizado sobre el dataset CTG.csv, utilizando una librería personalizada desarrollada en Python. El análisis incluye:

- Preprocesamiento completo del dataset
- Categorización automática de variables
- Generación de visualizaciones
- Interpretación y recomendaciones analíticas

La librería implementa funciones reutilizables con buenas prácticas de desarrollo, documentación y estructura modular.

2. DESCRIPCIÓN TÉCNICA DE MÓDULOS Y FUNCIONES

2.1. Módulo preprocessing.py

Este módulo contiene el pipeline completo de limpieza de datos.

FUNCIONES:

1. cargar_archivo(path)
    a. Carga archivos CSV o Excel.
    b. Devuelve un DataFrame
    c. Maneja errores controlados.
2. estandarizar_columnas(df)
    a. Limpia y normaliza nombres de columnas:
    b. minúsculas
    c. reemplazo de espacios y símbolos
    d. elimina inconsistencias
3. convertir_tipos(df)

    a. Convierte tipos automáticamente:
        i. intenta convertir numéricos
        ii. convierte fechas cuando aplica
        iii. reemplaza pd.NA → np.nan
4. eliminar_col_nulos(df, threshold)
    a. Calcula porcentaje de nulos por columna.
    b. Elimina columnas con proporción mayor al umbral.
    c. Mejora calidad del dataset antes de imputar.
5. imputar_valores(df, metodo)
    a. Imputa valores faltantes según método:
        i. mean
        ii. median
        iii. knn (si se usa)
    b. Maneja numéricos y categóricos por separado.
        i. Evita errores con pd.NA y dtype object.
6. outliers_iqr(df)
    a. Identifica outliers usando rango intercuartílico.
    b. Regresa un diccionario con máscaras por variable.
7. outliers_z(df, umbral)
        i. - Calcula Z-score y devuelve outliers.
        ii. - Permite ajustar sensibilidad.
8. tratar_outliers(df, metodo, accion)
    a. Aplica:
        i. winsorize
        ii. drop
    b. Puede usar IQR o Z-score.
        i. Previene distorsión extrema en variables contínuas.
9. preprocessing(path, …)
    a. Pipeline completo:
        i. cargar datos
        ii. 2. estandarizar columnas
        iii. convertir tipos
        iv. eliminar columnas con nulos excesivos
        v. imputar valores faltantes
        vi. tratar outliers
    b. Regresa un DataFrame listo para análisis

2.2. Módulo categorization.py

Este módulo clasifica variables automáticamente y resume su estructura.

FUNCIONES:

1. check_data_completeness(df)
    a. Evalúa por columna:
    b. total de nulos
    c. porcentaje de completitud
    d. tipo de dato
    e. métricas estadísticas (mínimos, máximos, medianas, etc.)
    f. Regresa un DataFrame resumen.
2. clasificacion_variables(df)
    a. Clasifica automáticamente columnas en:
        i. continuas
        ii. discretas
        iii. categóricas
    b. Útil para decidir tratamiento por tipo de variable.
3. categorization(df)
    a. Une las funciones anteriores.
    b. Regresa un reporte resumido con análisis por tipo de variable.

2.3. Módulo plots (barplots.py, histograms.py)

FUNCIONES PRINCIPALES:

1. plot_bar_counts(df, columna)
    a. Gráfico de barras con frecuencias.
    b. Ideal para columnas categóricas.
2. plot_histogram(df, columnas, kde, hue, facet)
    a. Histograma multivariable.
    b. KDE opcional.
    c. División por clase (hue).
    d. Soporta facetado en varias subfiguras.
3. Funciones auxiliares
    a. Configuración estética
    b. Control de bins
    c. Manejo de errores cuando columnas no existen

3. PIPELINE APLICADO AL DATASET CTG.csv

1. Se cargó el archivo original CTG.csv.
2. Se estandarizaron los nombres de las columnas.
3. Se detectaron y eliminaron columnas con más del 20% de nulos.
4. Se imputaron:
    a. variables numéricas → media
    b. variables categóricas → moda
5. Se analizaron y trataron outliers con:
    a. método: IQR
    b. acción: drop
6. Se clasificaron variables en continuas, categóricas y constantes.
7. Se generaron visualizaciones.

3. CONCLUSIONES

1. El pipeline del módulo preprocessing.py fue ejecutado y validado correctamente. Se comprobó que todas las etapas funcionan sin errores: carga de datos, estandarización de nombres de columnas, conversión de tipos, imputación de valores faltantes y tratamiento de outliers. El pipeline es funcional y puede aplicarse a otros datasets similares.
2. Se probaron individualmente las funciones de los módulos preprocessing, categorization y plots para confirmar que operan conforme a lo esperado. Esto incluyó la imputación con media y mediana, eliminación de columnas con alto porcentaje de nulos, clasificación automática de variables y la ejecución de los métodos de visualización.
3. Se generaron visualizaciones básicas (histogramas y gráficos de barras) únicamente para validar que las funciones de los módulos de gráficas producen resultados sin errores. Las visualizaciones se utilizaron únicamente como prueba de funcionamiento y no con fines analíticos.
4. Las funciones del módulo categorization.py clasificaron correctamente las columnas del dataset en continuas, categóricas y constantes, lo cual confirma la correcta implementación de la lógica de categorización.
5. El enfoque del trabajo se centró en validar el funcionamiento de los módulos desarrollados y no en extraer conclusiones estadísticas o clínicas del dataset. El objetivo fue comprobar la robustez y operatividad de la librería personalizada.

4. RECOMENDACIONES

1. Ampliar el análisis estadístico en futuras versiones, integrando boxplots, correlaciones y comparaciones entre variables del dataset.

2. Extender el conjunto de pruebas unitarias para cubrir más casos y excepciones, fortaleciendo la estabilidad y confiabilidad de la librería.
3. Aprovechar el dataset preprocesado para tareas posteriores de modelado supervisado, ya que el pipeline deja los datos en un formato adecuado para entrenamiento.
4. Documentar interpretaciones básicas de las visualizaciones en caso de usarse para análisis en lugar de simple validación técnica.

5. ESTRUCTURA DEL REPOSITORIO

```
ctg_viz_project/
│
├── ctg_viz/
│   ├── preprocessing.py
│   ├── categorization.py
│   ├── plots/
│   │   ├── histograms.py
│   │   ├── barplots.py
│
├── data/
│   └── CTG.csv
│
├── tests/
│   ├── test_preprocessing.py
│   ├── test_categorization.py
│   ├── test_plots.py
│
├── README.md
├── requirements.txt
```