

Manual de Usuario para el Script de Procesamiento de Eventos Agua Abajo

Grupo de Control y Procesamiento Digital de Señales
Universidad Nacional de Colombia
CHEC

April 28, 2025

Contents

1	Descripción General	2
2	Funcionalidades Principales	2
3	Bibliotecas Necesarias	2
4	Archivos Necesarios	2
5	Instrucciones de Uso	2
5.1	Preparación del Entorno	2
5.2	Ejecución del Script	3
5.3	Resultados del Procesamiento	3
6	Consideraciones Adicionales	3

1 Descripción General

El script de Python `procesamiento_eventos_aguas_abajo.py` está diseñado para procesar eventos relacionados con tramos de agua abajo. Este script es parte de un sistema más grande que maneja datos de eventos críticos en una red de transporte o infraestructura. El objetivo principal es identificar y asignar códigos específicos a los tramos de agua abajo basados en la ubicación geográfica y el tiempo.

El script utiliza datos de eventos y una red de transporte (representada por el DataFrame `redmt`) para determinar qué tramos están afectados por cada evento. Los resultados se guardan en un nuevo archivo de pickle, facilitando el análisis posterior.

2 Funcionalidades Principales

- **Carga de Datos:** El script carga datos de eventos y una red de transporte desde archivos pickle.
- **Procesamiento de Datos:** Para cada evento, identifica los tramos de agua abajo afectados y asigna los códigos correspondientes.
- **Progreso en Tiempo Real:** Muestra una barra de progreso que indica el progreso del procesamiento y el tiempo transcurrido.
- **Guardado de Resultados:** Los resultados del procesamiento se guardan en un nuevo archivo de pickle.

3 Bibliotecas Necesarias

Para ejecutar este script, es necesario tener instaladas las siguientes bibliotecas de Python:

- `os`: Para manejo de archivos y directorios.
- `time`: Para medir el tiempo de ejecución del script.
- `datetime`: Para manipulación de fechas y horas.
- `numpy`: Para operaciones numéricas.
- `pandas`: Para manipulación y análisis de datos.

4 Archivos Necesarios

Para que el script funcione correctamente, deben existir los siguientes archivos:

- `/kaggle/input/SuperEventos_Criticidad_AguasAbajo.pkl`: Archivo pickle que contiene los datos de eventos.
- `/kaggle/input/REDMT_1.pkl`: Archivo pickle que contiene los datos de la red de transporte.

5 Instrucciones de Uso

5.1 Preparación del Entorno

1. **Instalar Bibliotecas:** Asegúrese de tener instaladas todas las bibliotecas necesarias. Puede instalarlas utilizando pip:

```
pip install numpy pandas
```

2. **Descargar Archivos:** Asegúrese de tener los archivos `SuperEventos_Criticidad_AguasAbajo.pkl` y `REDMT_1.pkl` en la ubicación especificada.

5.2 Ejecución del Script

1. **Configurar Variables:** Asegúrese de que las variables de ruta en el script coincidan con la ubicación de sus archivos pickle.

2. **Ejecutar el Script:** Ejecute el script en su entorno de Python. Puede hacerlo desde la línea de comandos:

```
python procesamiento_eventos_aguas_abajo.py
```

3. **Monitorear el Progreso:** El script mostrará una barra de progreso en la terminal, indicando el progreso del procesamiento y el tiempo transcurrido.

5.3 Resultados del Procesamiento

1. **Archivo de Salida:** Una vez que el script haya terminado de ejecutarse, los resultados se guardarán en un nuevo archivo de pickle llamado `Eventos_AguasAbajo.CODEs_e.pkl`, donde `e` es un número que indica el segmento de datos procesado.

2. **Revisar Resultados:** Puede cargar el archivo de salida en un entorno de Python para revisar los resultados del procesamiento.

6 Consideraciones Adicionales

- **Particionamiento de Datos:** El script procesa los datos en segmentos (en este caso, en cinco partes). Esto puede ser útil para manejar grandes volúmenes de datos y evitar problemas de memoria.
- **Optimización de Rendimiento:** El script puede tardar en ejecutarse dependiendo del tamaño de los datos. Considere optimizar el rendimiento si es necesario.
- **Dependencia de Archivos:** El script depende de archivos específicos en ubicaciones específicas. Asegúrese de que los archivos estén disponibles y accesibles.
- **Error Handling:** Asegúrese de manejar posibles errores durante la ejecución del script, como archivos no encontrados o datos corruptos.