UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA DEPARTAMENTO DE MECÁNICA SEDE VIÑA DEL MAR

Aplicación

El contenido generado por IA puede ser incorrecto.

TRABAJO COMPUTACIÓN APLICADA

CERTAMEN REBOLIER

PROFESOR RAYMI ANTONIO VASQUEZ MORENO

INTEGRANTES: MANUEL MOLINA

RAY PALOMINO

JORGE MUÑOZ

SERGIO VERGARA

**Introducción**

En este informe se presenta el **preprocesamiento de una base de datos del Reboiler** de una planta de generación de dióxido de cloro, utilizando la herramienta **DataBruin**   
El objetivo principal del trabajo fue **limpiar y preparar los datos**, abordando las siguientes tareas:

* Eliminar valores atípicos del sensor de conductividad.
* Filtrar los datos para considerar únicamente el periodo más reciente de operación (desde el año 2016).
* Exportar los datos limpios en formato **CSV** (plano).
* Generar un archivo en formato **NPZ** con ventanas de tiempo, pensado para el desarrollo posterior de modelos de detección de anomalías.

Este preprocesamiento permite dejar la información en un formato adecuado y optimizado para análisis avanzados con técnicas de Machine Learning.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Paso 0: Configuración del entorno y ejecución de DataBruin**

**Antes de comenzar con el preprocesamiento de los datos del Reboiler, fue necesario configurar correctamente el entorno de trabajo e instalar DataBruin.**

**Proceso de instalación y acceso:**

1. **Descargar e instalar Anaconda desde su sitio oficial.**
2. **Una vez instalado, abrir Anaconda Prompt.**
3. **Seguir las instrucciones del sitio web oficial de DataBruin, copiando y pegando en la consola los comandos necesarios para descargar e instalar todos los recursos del entorno.**
4. **Con la instalación completada, se ejecutó el comando correspondiente para iniciar DataBruin desde Anaconda Prompt.**
5. **Finalmente, se accedió a la interfaz gráfica de DataBruin desde el navegador, donde se realizó todo el flujo de preprocesamiento de datos descrito en los pasos siguientes.**

**Este paso fue esencial para disponer de una plataforma funcional que permitiera realizar limpieza de datos, manipulación de columnas, análisis de series temporales y exportaciones en múltiples formatos (CSV, NPZ).**

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Paso 1: Carga del archivo Excel en DataBruin**

Una vez iniciado el entorno de trabajo, se procedió a **cargar el archivo Excel** con los datos crudos del Reboiler en la plataforma **DataBruin**.

El archivo contenía un total de **42.206 filas y 10 columnas**, con variables como **fecha, conductividad, temperatura, presión**, entre otras.

**Nota técnica:** Si el archivo Excel no es reconocido por DataBruin, es posible que falte instalar la librería **openpyxl**. Para solucionarlo, se puede ejecutar el siguiente comando en Anaconda Prompt

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

**Paso 2: Eliminación de columna con datos completamente faltantes**

En esta etapa se realizó una limpieza inicial del conjunto de datos eliminando columnas que no aportan información útil para el análisis. En particular, se identificó que la columna:

contenía únicamente valores faltantes (NaN) a lo largo de todas las filas del archivo.

Para depurar correctamente la base de datos, se utilizó el nodo Drop Columns en DataBruin para eliminar esta columna vacía, dejando un conjunto de datos más limpio y manejable.

Cantidad de datos después de este paso:  
42206 rows × 9 columns

**Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.**

**Imagen que contiene interior, computadora, computer, escritorio

El contenido generado por IA puede ser incorrecto.**

**Paso 3: Renombrar columnas de sensores**

Para facilitar el análisis y hacer más intuitiva la interpretación de los datos, se realizó un renombramiento de las columnas que representan sensores. Esto permite identificar rápidamente las variables de interés sin depender de nombres técnicos extensos o poco claros.

🔧 Esta acción se ejecutó mediante el nodo Rename Columns en DataBruin, asignando nombres más amigables y significativos, como por ejemplo:

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

**Paso 4: Ajuste de tipos de datos (Set Dtypes)**

En esta etapa se definieron correctamente los tipos de datos de cada columna para asegurar un procesamiento adecuado durante las etapas siguientes.

Se utilizó el bloque Set Dtypes en DataBruin, configurando:

* La columna Fecha como tipo datetime, ya que representa la marca temporal de cada medición.
* Las columnas correspondientes a sensores (como temperatura, nivel, presión, etc.) se establecieron como tipo numeric, para permitir su análisis y modelado posterior.

Este paso es esencial para que las herramientas de análisis temporal y los filtros numéricos funcionen correctamente dentro de la plataforma.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Paso 5: Rellenar datos faltantes (Handle Missing)**

En esta etapa, se abordó el problema de los valores faltantes (NaN) en los sensores del Reboiler. Para asegurar la continuidad de los datos en los análisis futuros, se utilizó la técnica de relleno hacia adelante (forward fill).

🔧 Se utilizó el nodo Handle Missing, con la siguiente configuración:

* Function: fillna
* Type: method
* Method: ffill (Forward Fill → rellena con el último valor válido conocido)
* Axis: none
* Limit: (no se modifica)

Esto permite mantener la consistencia temporal de las señales sin introducir valores artificiales arbitrarios. Cada valor faltante fue reemplazado por el último valor disponible inmediatamente anterior.

Resultado:  
Ahora los datos no tienen valores vacíos, lo que facilita el entrenamiento de modelos y evita errores en etapas futuras.

Escala de tiempo, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 6: Filtrado por fecha desde 2016**

El segundo filtro aplicado a la base de datos fue **por fecha**, ya que el enunciado solicita trabajar únicamente con la **etapa más reciente de operación del Reboiler**, es decir, **a partir del 1 de enero de 2016**.

Para esto, se utilizó nuevamente el nodo **"Filter Rows"**, configurando la condición para conservar solo las filas cuya fecha fuese **mayor o igual a 2016-01-01 00:00:00**.

Este paso redujo aún más el volumen de datos, quedando en un total de **11.530 registros y 10 columnas**, que corresponden exclusivamente al periodo requerido.

Gráfico

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto. Tabla

El contenido generado por IA puede ser incorrecto.

**Paso 7: Indexación temporal de los datos**

**Para asegurar un orden adecuado y una correcta manipulación de los datos en análisis temporales, se estableció la columna Fecha como índice del dataset.**

**Esto se realizó utilizando el nodo Set Index en DataBruin:**

* **Columna seleccionada: Fecha**
* **Configuración: Se deja activada la opción de mantener la columna como índice principal.**

**Al convertir la columna de fecha en índice, se garantiza que todos los procesos futuros (ventaneo, análisis de series de tiempo, exportaciones, etc.) se ejecuten respetando el orden cronológico y sin duplicidades.**

**Resultado:  
La tabla queda indexada por tiempo, ideal para cualquier análisis en series de tiempo o modelos predictivos que dependan de la secuencia de eventos.**

**Escala de tiempo

El contenido generado por IA puede ser incorrecto.** **Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.**

**Paso 8: Filtrado de valores atípicos del sensor de conductividad**

Una de las condiciones establecidas en la consigna fue eliminar mediciones no posibles del sensor de conductividad, específicamente aquellas que fueran mayores a 12.

Para ello se utilizó nuevamente el nodo Filter Rows en DataBruin con la siguiente configuración:

* Columna seleccionada: Conductividad (o el nombre correspondiente al sensor en tu dataset)
* Condición: Conductividad ≤ 12

Esto elimina automáticamente todas las filas en las que la conductividad sea superior a 12, reduciendo aún más la cantidad de datos y dejando solo los registros válidos y dentro del rango esperado.

Resultado:  
La base de datos resultante es más confiable y precisa para su análisis, ya que se eliminaron lecturas anómalas o erróneas del sensor.

Cantidad de datos: 11166 rows × 8 columns

Escala de tiempo

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto. Tabla

El contenido generado por IA puede ser incorrecto.Imagen que contiene grande, pequeño, entero, llenado

El contenido generado por IA puede ser incorrecto.

**Paso 9: Re-muestreo de la base de datos para ajustar los intervalos temporales**

En este paso, utilizamos la función resample para trabajar con los intervalos de tiempo de la base de datos.

La base de datos original registra datos cada 2 horas, lo que equivale a intervalos de 120 minutos. Para manejar estos intervalos en minutos dentro de nuestro análisis, usamos el parámetro '120T' (donde T representa minutos) para indicar la frecuencia deseada.

Con resample('120T') estamos agrupando o reajustando los datos según su frecuencia original de 2 horas, pero expresada en minutos. Esto facilita la manipulación y análisis temporal, asegurando que el muestreo sea consistente y acorde a la resolución real de la base de datos.

Resultado:  
Se logra un manejo adecuado de los intervalos temporales, permitiendo realizar análisis o transformaciones que dependan de un muestreo con una frecuencia precisa y homogénea.

Escala de tiempo

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Paso 10: Aplicación de resample con el bloque Aggregate usando la mediana**

En este paso aplicamos el re-muestreo de los datos utilizando el bloque **Aggregate** para resumir la información.

En la configuración del bloque Aggregate, seleccionamos la opción **“all columns”** y elegimos la función **mediana (median)** para cada intervalo de tiempo definido en el paso anterior.

Esto significa que, para cada intervalo de 120 minutos, se calculará la mediana de todas las columnas numéricas del dataset, lo que ayuda a reducir el impacto de valores extremos o ruidos en las mediciones.

Resultado:  
Obtenemos una versión suavizada y más representativa de los datos para cada intervalo, facilitando el análisis posterior con datos menos dispersos y más confiables.

Escala de tiempo

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto. Tabla

El contenido generado por IA puede ser incorrecto.

**Injertación de la data pronosticada tras el resample y aggregate**

En este paso, integramos la data pronosticada, obtenida tras aplicar el re-muestreo (resample) y la agregación por mediana (aggregate), con la tabla original del Reboiler para su análisis conjunto.

La tabla final contiene un total de **11.166 datos**, de los cuales:

* **8.686 datos (77,7%)** corresponden a condiciones normales de operación, etiquetados con valor **“0”**.
* **2.480 datos (22,21%)** corresponden a condiciones anómalas o fallas, etiquetados con valor **“1”**.

Dado que el porcentaje de datos en condición de falla (22,21%) es significativamente mayor al umbral del 5%, se concluye que la base de datos está **balanceada** en términos de distribución entre clases normales y anómalas. Esto es beneficioso para el entrenamiento y validación de modelos predictivos, ya que evita el sesgo hacia una sola clase.

Resultado:  
Con la injertación realizada, se cuenta con un dataset consolidado y balanceado, listo para análisis más precisos y desarrollo de modelos de detección de fallas.

Tabla

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Paso 11: Guardado del archivo limpio reboiler\_limpio.csv**

En este paso guardamos la tabla procesada y limpia en un archivo llamado **reboiler\_limpio.csv**.

Este archivo será utilizado en pasos posteriores, permitiendo cargar fácilmente los datos ya procesados en otros bloques o flujos de trabajo sin necesidad de repetir todo el proceso.

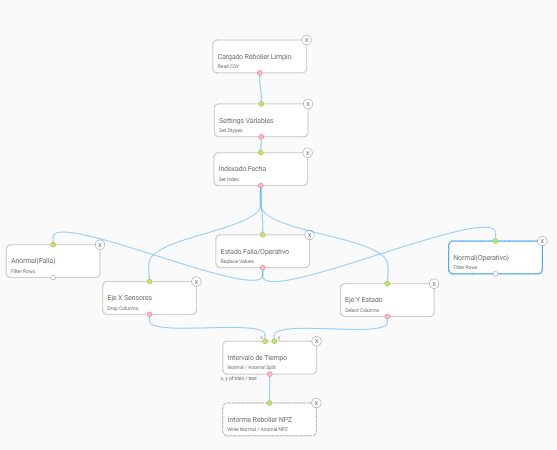
Guardar el archivo facilita la organización y el manejo de la información para análisis futuros o etapas adicionales del proyecto.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Siguiente Ramal**



**Paso 12: Carga del archivo reboiler\_limpio.csv usando el bloque Read CSV**

En este paso, procedemos a **cargar el archivo reboiler\_limpio.csv** que fue guardado previamente, utilizando el bloque **Read CSV**.

Este bloque permite leer archivos en formato .csv e incorporarlos nuevamente al flujo de trabajo para continuar con el análisis o aplicar nuevos procesos.

Configuración del bloque:

* **Archivo seleccionado**: reboiler\_limpio.csv
* **Encabezado activado** (si corresponde)
* **Separador**: Coma (,), si el archivo sigue el formato estándar CSV

Resultado:  
El archivo limpio y balanceado se carga correctamente, quedando disponible para ser utilizado en nuevos análisis, modelado o visualización.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.

**Paso 13: Ajuste de tipos de datos con el bloque Set DTypes**

En este paso, utilizamos nuevamente el bloque **Set DTypes** para asegurarnos de que los tipos de datos estén correctamente definidos para su análisis.

Configuración del bloque:

* **Columna de fecha**: Se ajusta al tipo **datetime**, lo que permite trabajar con funciones temporales como resampleo, ordenamiento cronológico y filtrado por fechas.
* **Columnas de sensores**: Se ajustan al tipo **numeric**, lo que es esencial para aplicar operaciones matemáticas, estadísticas y técnicas de modelado posteriores.

Este paso es fundamental para evitar errores en bloques posteriores que requieren formatos específicos para operar correctamente.

Resultado:  
La tabla queda con todos sus datos correctamente tipificados, lista para continuar con el análisis sin conflictos de formato.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Paso 14: Indexación de la columna de fecha con el bloque Set Index**

En este paso utilizamos el bloque **Set Index** para **establecer la columna de fecha como índice principal** del dataset.

Al indexar por fecha, se facilita el manejo temporal de los datos, permitiendo realizar operaciones como filtrado por rangos de tiempo, resampleo y visualización cronológica de manera más eficiente.

Configuración del bloque:

* **Columna seleccionada como índice**: fecha (o el nombre correspondiente en tu dataset)

Este paso es clave para trabajar correctamente con series temporales y asegurar que las operaciones posteriores estén alineadas cronológicamente.

Resultado:  
La columna de fecha se convierte en el índice del dataframe, permitiendo una navegación y manipulación temporal más estructurada.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Paso 15: Reemplazo de valores en la columna de condición con el bloque Replace Value**

En este paso utilizamos el bloque **Replace Value** para hacer más intuitiva la interpretación de los estados operativos y las fallas en la base de datos.

Configuración del bloque:

* En la columna de condición:
  + Se reemplaza el valor **1** por **Falla**
  + Se reemplaza el valor **0** por **Operativo**

Esta transformación convierte los valores numéricos en etiquetas descriptivas, facilitando la lectura y comprensión del dataset por parte de cualquier persona, incluso sin conocimientos técnicos.

Resultado:  
La columna de condición ahora muestra claramente los estados como **“Falla”** u **“Operativo”**, mejorando la presentación y facilitando análisis visuales o reportes.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.

**Paso 16: Filtrado de registros con fallas utilizando el bloque Filter Rows**

En este paso se utiliza el bloque **Filter Rows** para **extraer únicamente los registros en los que el Reboiler presentó una condición anómala**, es decir, cuando ocurrió una **falla**.

Configuración del bloque:

* **Columna**: Condición (o el nombre que hayas dado a la columna)
* **Condición aplicada**: Condición == "Falla"

Este filtro permite aislar y analizar exclusivamente los casos en que el sistema no estuvo operativo, lo cual es fundamental para el estudio de patrones de falla, mantenimiento predictivo o generación de alertas.

Resultado:  
Se obtuvieron **2.480 filas** correspondientes a momentos en que el Reboiler se encontraba en estado de **falla**, distribuidas en **8 columnas**.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.

**Paso 17: Filtrado de registros en estado operativo utilizando el bloque Filter Rows**

En este paso se utiliza nuevamente el bloque **Filter Rows**, esta vez para **extraer únicamente los registros en los que el Reboiler estuvo funcionando con normalidad**, es decir, en estado **operativo**.

Configuración del bloque:

* **Columna**: Condición (o el nombre correspondiente en tu dataset)
* **Condición aplicada**: Condición == "Operativo"

Este filtrado permite enfocarse en los periodos de funcionamiento normal del sistema, lo cual puede ser útil para establecer comparaciones, referencias o análisis de estabilidad.

Resultado:  
Se obtuvieron **8.686 filas** correspondientes a momentos en que el Reboiler se encontraba en estado **operativo**, con un total de **8 columnas**.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

**Paso 18: Eliminación de columnas innecesarias y conexión al bloque de separación “Anormal / Normal Split”**

Previo a la separación de datos entre condiciones normales y anormales, se realizó una limpieza adicional del dataset eliminando columnas que no aportaban valor directo al análisis, tales como aquellas que no entregaban datos o cuyo contenido era irrelevante para la clasificación de eventos.

Para esto se utilizó el nodo Drop Columns, donde se seleccionaron explícitamente los nombres de las columnas a eliminar, conservando únicamente los sensores relevantes para el análisis de fallas del Reboiler.

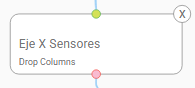
**Detalles técnicos:**

* Se eliminaron columnas vacías (como VAL356M003-Carga Motor, si no tenía datos).
* Se conservaron columnas como Temperatura, Nivel, Presion, Conductividad, Estado, entre otras.

**Conexión del flujo:**

* El bloque Drop Columns se conectó directamente al eje X del nodo “Anormal / Normal Split”, lo que asegura que las variables de entrada utilizadas para el análisis correspondan únicamente a sensores útiles y filtrados.

Esta conexión garantiza que el bloque de separación opere sobre variables limpias y adecuadas, evitando errores o datos residuales que afecten la calidad del análisis de anomalías.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.

**Paso 19: Selección de columna clave para clasificación del estado del Reboiler**

Para preparar correctamente los datos que alimentarán los modelos de detección de anomalías, fue necesario identificar y seleccionar explícitamente la columna que indica el **estado de funcionamiento del Reboiler** (por ejemplo, si está operativo o en falla).

Para ello, se utilizó el nodo **Select Columns**, el cual permite extraer columnas específicas del conjunto de datos.

**Configuración realizada:**

* Se seleccionó la columna **Estado** o su equivalente, que indica si el sistema se encuentra en **modo operativo (0) o falla (1)**.
* Este nodo se conectó al **eje Y del bloque “Anormal / Normal Split”**, que necesita una variable de referencia para separar las ventanas etiquetadas como normales o anormales.

Esta configuración es crucial para que los datos se dividan correctamente y así poder entrenar modelos que detecten comportamientos fuera de lo normal basados en esta etiqueta.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

**Paso 20: Unión de variables filtradas y estado del sistema para exportación NPZ**

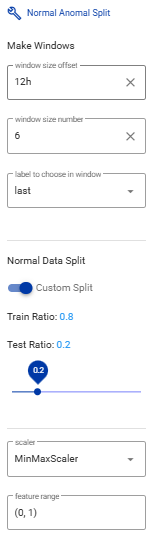
Una vez realizados los pasos de limpieza y separación de columnas clave, se procedió a unir ambas salidas —las variables de sensores relevantes y la etiqueta de estado— utilizando el nodo **Anormal / Normal Split**.

Este bloque permite separar automáticamente los datos en conjuntos etiquetados como “normal” y “anormal”, lo que es esencial para el desarrollo posterior de modelos de detección de anomalías.

**Configuración del nodo:**

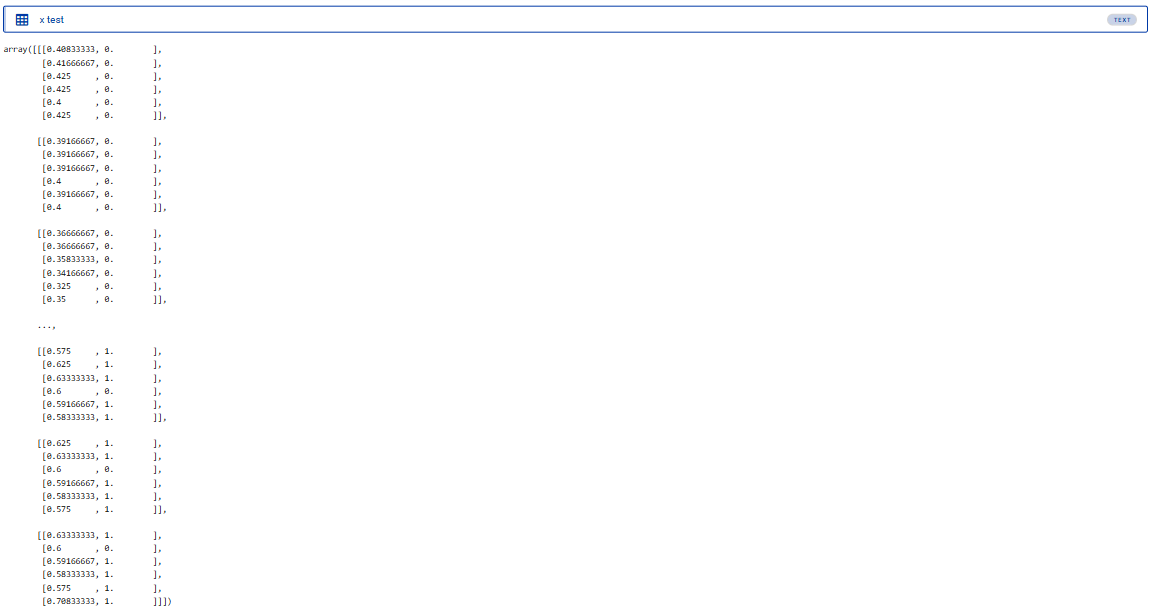
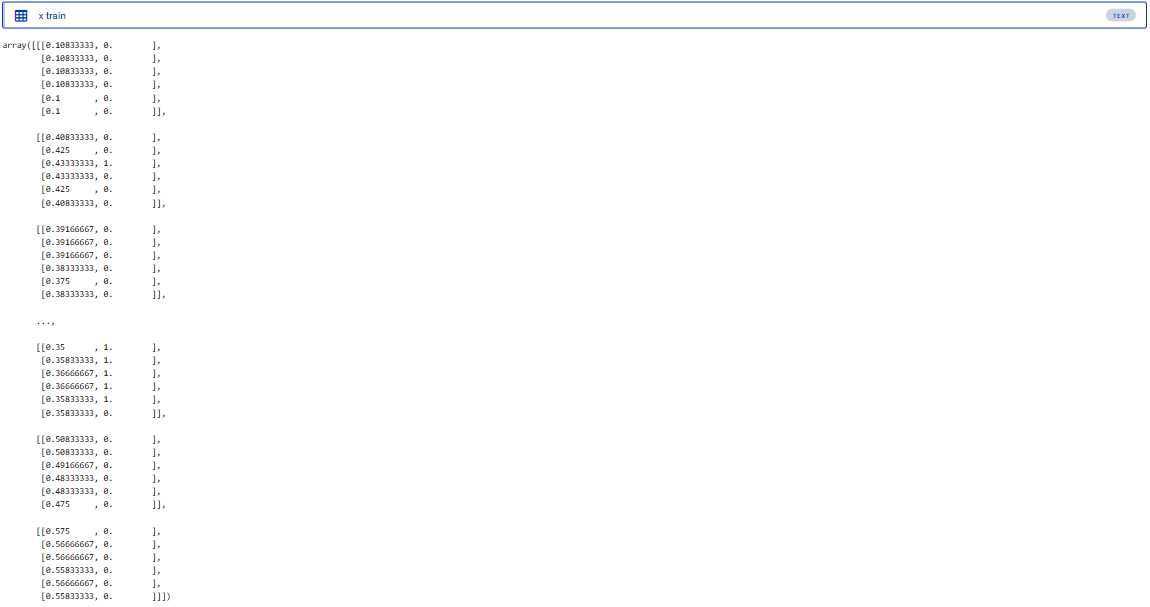
* El nodo **Drop Columns**, que contiene las variables limpias, se conectó al **eje X** del nodo **Anormal / Normal Split**.
* El nodo **Select Columns**, que contiene la columna Estado, se conectó al **eje Y del nodo Anormal / Normal Split**.
* Esta configuración permite crear subconjuntos de datos ordenados y listos para ser usados en ventaneo y exportación.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Se ha logrado crear exitosamente el archivo .NPZ correspondiente al Reboiler.**  
A continuación, se presentan las imágenes que contienen los datos utilizados para su elaboración.

**EJE X**



**EJE Y**



**Paso 21: Exportación del archivo .npz desde los datos separados “Anormal / Normal Split”**

Con los datos ya separados correctamente en condiciones normales y anormales gracias al nodo **Anormal / Normal Split**, el siguiente objetivo fue **guardar y exportar los datos ventaneados en un archivo .npz**, el cual puede ser utilizado posteriormente para entrenar modelos de machine learning orientados a la detección de anomalías.

Para ello se utilizó el nodo **Write Normal / Anormal Split**, que permite almacenar ambos conjuntos de datos (normal y anormal) directamente en un archivo comprimido .npz.

**Configuración realizada:**

* Se conectó directamente la salida del nodo **Anormal / Normal Split** al bloque **Write Normal / Anormal Split**.
* Se definió el nombre del archivo de salida como:  
  **Reboiler\_Ventanas.npz**
* Este archivo almacena internamente los arrays de datos con las etiquetas necesarias para el entrenamiento de algoritmos supervisados o semisupervisados.

**Importancia de este paso:**

Exportar el .npz permite que el dataset quede listo para su uso en modelos de aprendizaje automático, facilitando así futuras tareas de análisis predictivo, mantenimiento preventivo y monitoreo inteligente del Reboiler.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

**Conclusión**

A lo largo de este trabajo se logró cumplir satisfactoriamente con los objetivos planteados en la consigna, realizando un preprocesamiento completo y estructurado de los datos operativos del **Reboiler** mediante la plataforma **DataBruin PreProcessing Studio**.

Entre los logros más destacados se encuentran:

* **Carga y limpieza** del dataset original, compuesto por más de 42.000 registros.
* **Eliminación de columnas vacías** y tratamiento de **valores atípicos**, en particular en la variable de **conductividad**.
* **Filtrado temporal**, trabajando exclusivamente con datos desde el año 2016 en adelante, según lo requerido.
* **Relleno de valores faltantes**, renombramiento de columnas y **tipificación adecuada** de los datos.
* **Separación de condiciones normales y anómalas**, identificando con claridad los diferentes estados de operación del Reboiler.
* Aplicación de **técnicas de resampleo y agregación temporal** para reducir el ruido y estandarizar los intervalos de medición.
* Generación de un archivo limpio en formato **.csv**, junto con la **creación exitosa del archivo. npz**, que contiene los conjuntos de datos preprocesados para su uso directo en modelos de machine learning.

Además, se aplicó una estructura de ventanas móviles para la generación de secuencias temporales, con un **tamaño de ventana de 6 registros** y un **offset de 12 horas**, lo que dio lugar a un conjunto de 10.791 muestras (x\_mix, y\_mix). El análisis de la variable y\_mix —que representa el estado del sistema (0: normal, 1: anómalo)— permitió detectar claramente los momentos de transición entre funcionamiento estable y condiciones de falla.

A partir de este análisis, se identificó el **punto medio de la zona de falla**, definido como el valor central entre el inicio y fin del comportamiento anómalo. Este punto crítico es especialmente valioso porque:

* Marca el momento de mayor severidad de la anomalía.
* Sirve como referencia clave para la calibración de modelos de detección temprana.
* Permite planificar estrategias de mantenimiento predictivo basadas en evidencia temporal.