Predicción del frío a utilizar el día siguiente en una planta productora de cerveza

Asignatura:
-Laboratorio de Datos II

Docentes:
-Borzone, Eugenio
-Duarte, Sofía
-Gerard, Matias

-

Integrantes:
-Marzioni, Agustín
-Scalzo, Santiago

Fecha límite de entrega: 17/11/2025

Introducción

Nuestro trabajo consistió en predecir los kilowatts(Kw) de frío que utilizara una planta de cerveza para enfriar una parte puntual de su planta.
Para esto, comenzamos preparando los datos, concatenando los dataset y seleccionando únicamente algunas de las hojas con conocimiento experto.
Posteriormente continuamos realizando el análisis exploratorio de datos para luego generar un pipeline de preprocesamiento que entrene diferentes modelos y así buscar la mejor métrica de rendimiento

Conformación del dataset

El conjunto de datos se veía conformado por un total de tres datasets, compuesto cada uno en su mayoría por hojas de tipo “Consolidado” y “Totalizadores”.
A su vez, la separación en estos dos “tipos de hojas” se volvía a separar en diferentes valores a medir como son la energía, el agua consumida, los “KPI” (métricas que una empresa utiliza para medir su rendimiento), entre otros.

Preparación de los datos

Para comenzar con la preparación previa a la realización del análisis exploratorio de datos(EDA) lo primero que se realizó fue la concatenación de los datasets. A partir de esto nos encontramos con fechas superpuestas entre las diferentes hojas, por lo que tuvimos que manualmente cortar las hojas para posteriormente volver a concatenarlas.

Luego, por conocimiento experto, decidimos quedarnos únicamente con aquellas hojas de “Consolidado” para realizar nuestras predicciones

Análisis exploratorio de datos

El objetivo de esta fase es preparar, limpiar y analizar el conjunto de datos de entrenamiento para identificar patrones, anomalías y relaciones relevantes que informan la construcción del modelo de predicción

Carga y Preparación Inicial

Se inició cargando el conjunto de datos foundational_dataset.csv en un DataFrame de pandas. Este se consigue quitando, con conocimiento experto, todas las columnas que comenzaban distinto de “Consolidado”.

La variable objetivo (y) se definió como el consumo de “Frío (Kw)" del día siguiente, lo cual se logró desplazando la columna Frio (Kw) un período hacia atrás (.shift(-1)) y se eliminó la última fila dado que quedaba como Nan debido al “shifteo”

Ingeniería de Características

Para capturar la dependencia temporal y la estacionalidad del consumo de frío, se crearon nuevas características basadas en valores pasados:

- **Características de Diferencia (Lag):** Se calcularon las diferencias diarias del consumo de frío, desplazadas un día (Frio_diff1_lag1, Frio_diff7_lag1), para capturar el cambio reciente en el consumo.
- **Características de Ventana Móvil:** Se generaron medias (Frio_roll_mean_...) y desviaciones estándar (Frio_roll_std_...) móviles del consumo de frío, también desplazadas un día. Se utilizaron ventanas de 3, 7, 14 y 28 días para capturar tendencias a corto y medio plazo.

Partición de Datos

Dado que se trata de datos de series temporales, se aplicó una **partición temporal** estricta para evitar la fuga de datos (data leakage) del futuro al pasado.

- El conjunto de datos se dividió en entrenamiento y prueba basándose en el tiempo.
- El **conjunto de prueba** se definió como el **30% final** de los datos (del 2022-10-29 al 2023-10-25).
- El **conjunto de entrenamiento** constituyó el **70% inicial** (del 2020-07-01 al 2022-10-28).

Todo el preprocesamiento posterior se ajustó (entrenó) únicamente sobre el conjunto de x_train y y_train. La columna FECHA_HORA se eliminó después de realizar la partición.

Limpieza y Preprocesamiento de Datos

El preprocesamiento se centró en el conjunto de entrenamiento (x_train) para manejar valores atípicos y faltantes.

- **Manejo de Outliers:**
    1. Se utilizó la **Mediana de la Desviación Absoluta (MAD)** con un umbral de Z-score de 3.5 para definir los límites de valores atípicos.
    2. Como método alternativo si el MAD era cero, se usaron los cuantiles 0.001 y 0.999.
    3. Los valores identificados como atípicos fueron convertidos a NaN para ser tratados en el siguiente paso.
- **Imputación de Valores Faltantes (NaNs):**
    1. **Columnas con Alto % de NaN:** Primero, se identificaron y apartaron las columnas que tenían más del 95% de valores NaN.
    2. **Escalado:** Los datos numéricos restantes se escalaron utilizando RobustScaler, que es resistente a los outliers.
    3. **Imputación KNN:** Los valores NaN (tanto los originales como los generados por outliers) se imputaron utilizando KNNImputer (con 5 vecinos y ponderación por distancia) sobre los datos escalados.
    4. **Escalado Inverso:** Los datos se transformaron de nuevo a su escala original.

5. **Relleno Final:** Las columnas con alto porcentaje de NaN (apartadas en el paso 1) se rellenaron con el valor 0.

Selección de Características

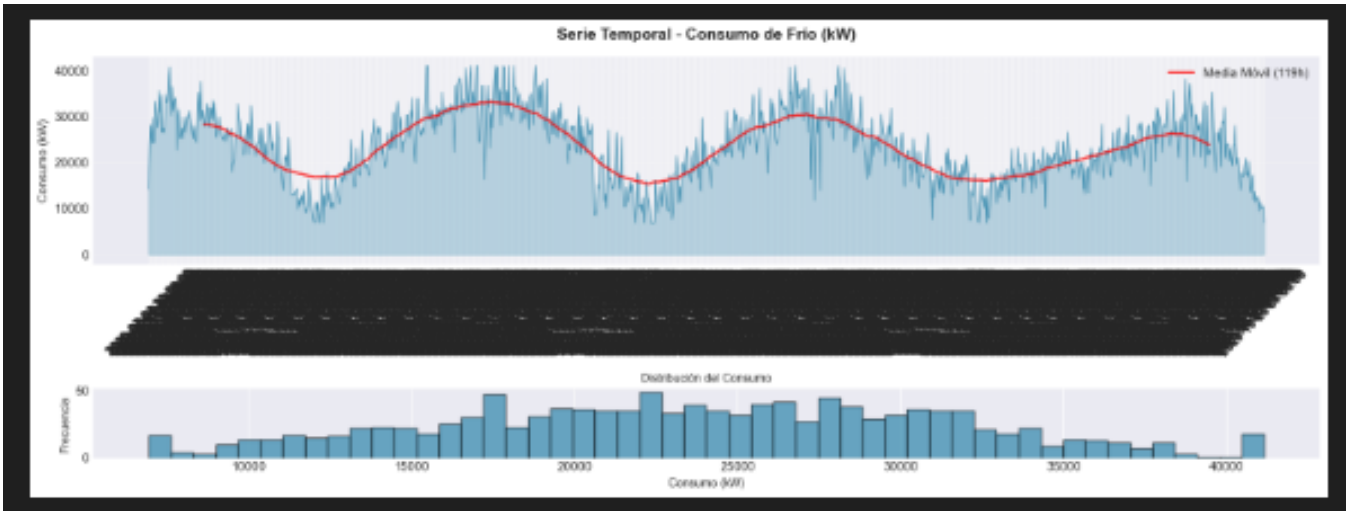
Se aplicaron varias técnicas para reducir la dimensionalidad del conjunto de datos y eliminar características irrelevantes o redundantes.

- **Eliminación por Baja Varianza:**
  - Se calculó el **Coeficiente de Variación (CV)** para todas las características numéricas y se eliminaron aquellas con un índice menor a 0.1
- **Eliminación por Correlación con el Target:**
  - Se evaluó una lista predefinida de características (analizar_corr). Estas se obtuvieron por conocimiento experto.
  - Se calcularon las correlaciones de **Pearson** (lineal) y **Spearman** (monotónica) de estas características con la variable objetivo (y_train) y se eliminaron aquellas cuyo valor absoluto de correlación era inferior a 0.3
- **Análisis de Multicolinealidad (Visual):**
  - Para identificar la redundancia entre las propias características, se seleccionaron las 20 variables más correlacionadas con el target.
  - Se generó un **heatmap triangular** de la matriz de correlación de Pearson entre estas 20 variables. Esto permite visualizar gráficamente qué características predictoras están fuertemente correlacionadas entre sí.

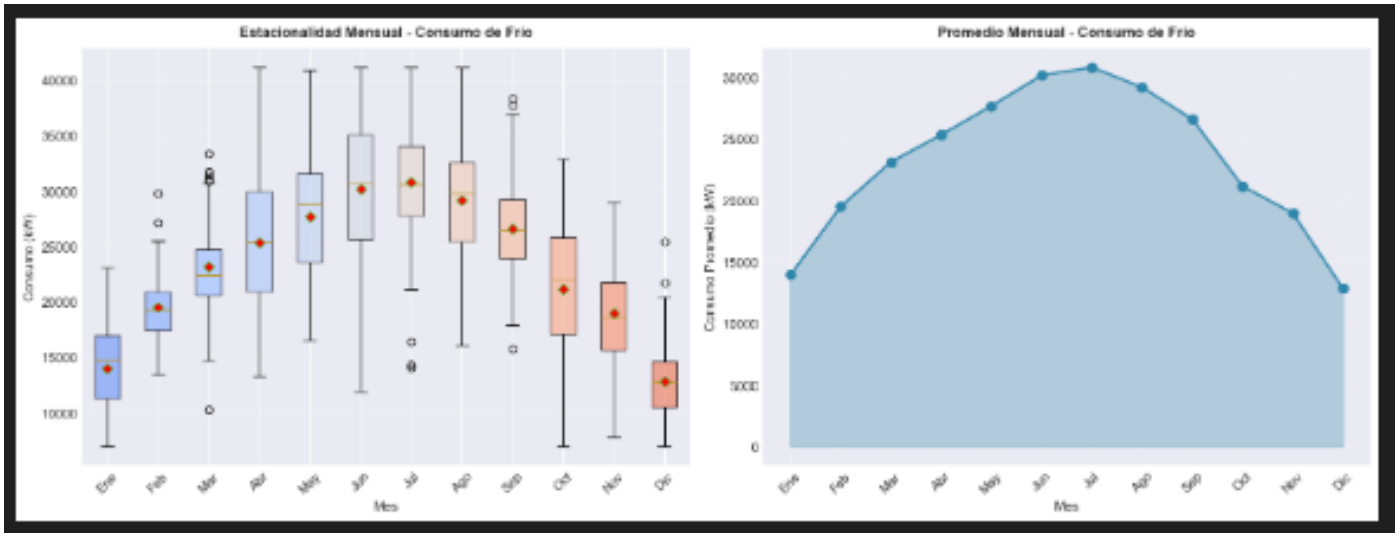
Visualizaciones

Finalmente, se generó un conjunto de funciones para generar un dashboard del EDA. Estas son:

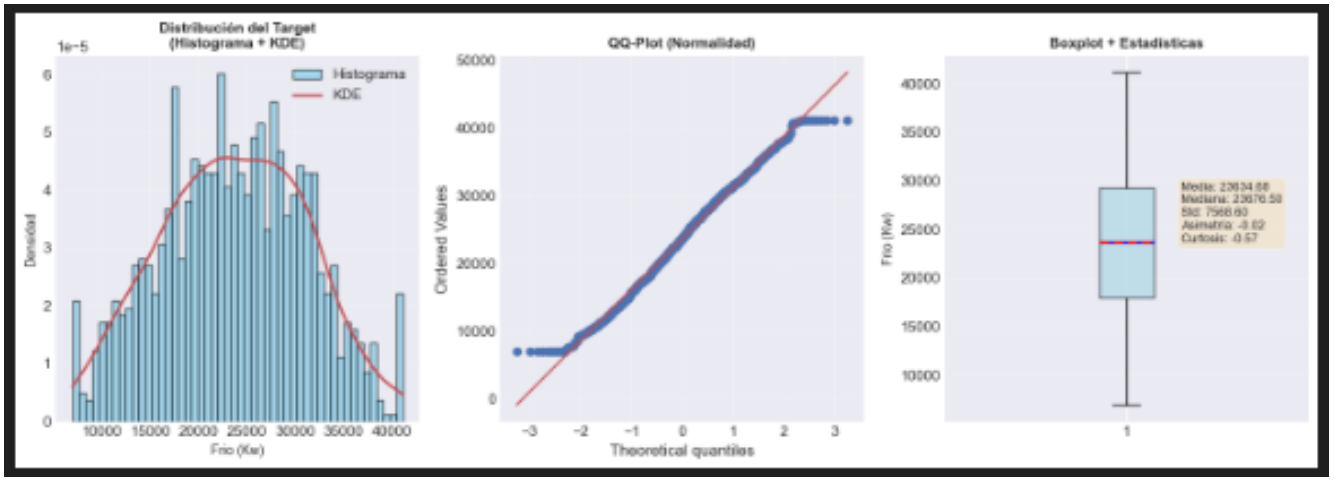
- **Gráfico de Serie Temporal:** Para visualizar el consumo de "Frio (Kw)" a lo largo del tiempo.



- **Gráfico de Distribución:** Un histograma y gráfico KDE para entender la distribución del consumo.



- **Gráficos de Estacionalidad:** Boxplots para analizar el consumo por mes y por día de la semana.



- **Gráficos de Dispersión (Scatter):** Para visualizar la relación entre el target y las 5 variables más correlacionadas.
-
- **Análisis por Área:** Un conjunto de gráficos (Boxplots, Gráfico de Torta y Barras) para comparar el consumo promedio y total entre las diferentes áreas de la planta.

Desarrollo del Pipeline de Procesamiento y Modelado

Para abordar el desafío de predecir el consumo de "Frío (Kw)" del día siguiente, se desarrolló un pipeline de datos robusto. Este pipeline está diseñado para ingresar nuevos datos desde un archivo Excel, procesarlos con los datos históricos de entrenamiento y generar una predicción utilizando un modelo de RandomForestRegressor previamente optimizado.

A continuación, se detallan los pasos secuenciales del pipeline:

Carga de Datos

El proceso inicia con la ingesta de datos desde un archivo.

1. **Lectura y Filtrado:** El sistema lee todas las hojas del archivo. Se aplica un filtro para seleccionar únicamente aquellas hojas cuyo nombre comienza con "Consolidado"
2. **Preparación de Hojas (preparar_hoja):** Cada hoja seleccionada pasa por una función de preparación, que realiza dos tareas principales:
    - **Normalización de Timestamp:** Construye una columna FECHA_HORA unificada a partir de las columnas DIA y HORA existentes. Maneja los errores de parseo que podrían generar valores nulos (NaT).
    - **Resolución de Duplicados Intra-Hoja:** Agrupa los datos por FECHA_HORA. Para asegurar una única fila por marca de tiempo, aplica una agregación: calcula la media (mean) para todas las columnas numéricas y toma el primer valor (first) para las no numéricas.
3. **Fusión de Hojas:** Una vez que todas las hojas están limpias y con una frecuencia coherente, se fusionan en un único DataFrame. Se utiliza la columna FECHA_HORA para no perder ninguna medición y el resultado se ordena cronológicamente.

Transformación de Frecuencia e Ingeniería de Características

Dado que el objetivo es una predicción diaria, la frecuencia de los datos se transforma.

1. **Remuestreo a Frecuencia Diaria:** El pipeline selecciona la **última medición registrada de cada día**. Esto se logra agrupando por fecha y seleccionando el índice de la FECHA_HORA más reciente para esa fecha.
2. **Creación de la Variable Objetivo:** La variable a predecir se define como el valor de "Frío (Kw)" del día siguiente. Esto se implementa desplazando la columna mencionada un período hacia atrás (shift(-1)).
3. **Ingeniería de Características:** Se generan nuevas características basadas en el historial de "Frío (Kw)" para capturar tendencias y estacionalidad:
    - **Diferencias Rezagadas:** Se calcula la diferencia del valor respecto al día anterior, rezagada un día (Frio_diff1_lag1).
    - **Estadísticas Móviles:** Se calculan medias (Frio_roll_mean_..._lag1) y desviaciones estándar (Frio_roll_std_..._lag1) móviles. Estas se generan para ventanas de 3, 7, 14 y 28 días, todas rezagadas un día (shift(1)) para evitar fuga de datos (data leakage).

Carga y Alineación de Datasets

El pipeline opera cargando un conjunto de datos de entrenamiento (x_train.csv, y_train.csv) y validación (x_test.csv, y_test.csv) previamente procesados.

1. **Consolidación del Training Set:** Los datos de entrenamiento y validación se concatenan para formar un único y completo conjunto de entrenamiento.
2. **Selección de Características:** Se aplica una lista predefinida de 20 características importantes tanto al conjunto de entrenamiento como al nuevo conjunto de test

Preprocesamiento Robusto de Características

1. **Detección de Outliers (MAD):** Se utiliza un método robusto para detectar valores atípicos basado en la **Desviación Absoluta Mediana (MAD)**.
    - Los límites se calculan *únicamente* sobre los datos de entrenamiento.
    - Se utiliza un umbral Z-score robusto de 3.5.
    - En caso de que MAD sea cero (sin variabilidad), el sistema utiliza un *fallback* a percentiles amplios (0.001 y 0.999) para definir los límites.
2. **Enmascaramiento de Outliers:** Todos los valores (tanto en train como en test) que caen fuera de los límites calculados en el paso anterior se reemplazan por valores nulos.
3. **Escalado e Imputación (KNN):**
    - **Escalado:** Los datos numéricos se escalan utilizando RobustScaler. Este escalador es insensible a los outliers (que ahora están enmascarados como NaN) y se ajusta (fit) solo con los datos de entrenamiento.
    - **Imputación:** Los valores nulos (tanto los originales como los generados por outliers) se imputan utilizando KNNImputer. Este método rellena los faltantes basándose en los 5 vecinos más cercanos (n_neighbors=5) en el *espacio escalado*, ponderando por distancia.
    - **Desescalado:** Finalmente, los datos imputados se transforman de nuevo a su escala original

Transformación Final y Modelado

1. **Transformación Yeo-Johnson:** El conjunto de datos limpio e imputado se somete a una PowerTransformer con el método yeo-johnson. Esta transformación ajusta la distribución de cada característica para que sea más Gaussiana (normal), estabilizando la varianza y mejorando el rendimiento del modelo.
2. **Entrenamiento y Predicción:**
    - Se instancia el modelo RandomForestRegressor con los hiper parámetros optimizados
    - El modelo se entrena (fit) sobre el conjunto completo de entrenamiento, ya transformado.
    - El modelo entrenado se utiliza para predecir sobre el conjunto de test, también transformado.

Evaluación

El pipeline concluye calculando y reportando las métricas de rendimiento del modelo sobre los datos de test. La métrica principal es el Error Absoluto Medio (MAE), complementada por MSE, RMSE y R^2.

Entrenamiento y Optimización

Búsqueda de Hiper Parámetros con Optuna

La primera fase de optimización consistió en una búsqueda automática de hiper parámetros para un modelo RandomForestRegressor utilizando el conjunto de datos completo

1. **Partición de Validación Temporal:** El conjunto de entrenamiento se dividió en dos subconjuntos: un 70% inicial para entrenamiento y el 30% final para validación, respetando el orden cronológico
2. **Búsqueda (Optuna):** Se configuró un estudio de Optuna para minimizar el Error Cuadrático Medio (RMSE) en el conjunto de validación.
3. **Espacio de Búsqueda:** Se exploraron los siguientes hiper parámetros:
    - max_depth: [None, 6, 10, 16, 24, 32]
    - max_features: ["sqrt", "log2", 0.5, 0.7, 1.0]
    - bootstrap: [True, False]
    - n_estimators: (200 a 2000)
    - min_samples_split: (2 a 20)
    - min_samples_leaf: (1 a 20)
    - max_samples: (0.5 a 1.0, solo si bootstrap=True)

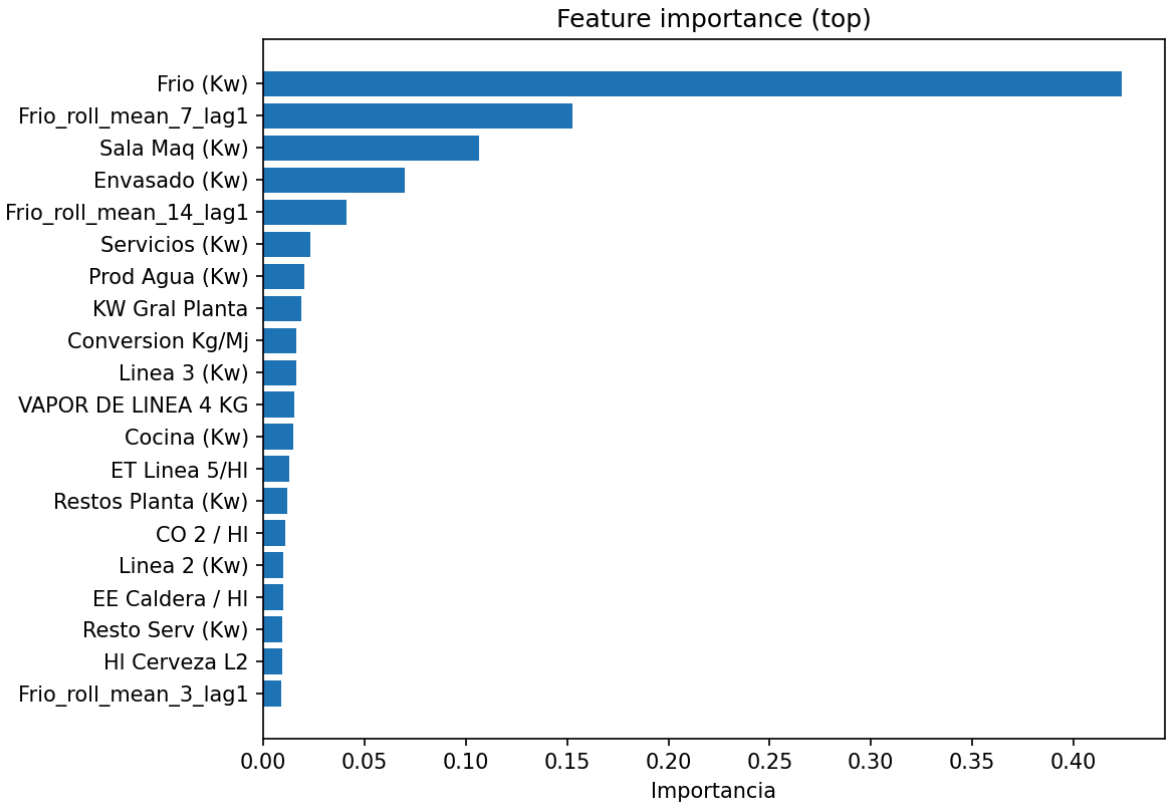
Desarrollo del Modelo Final

1. **Transformación de Potencia:** Se aplicó un PowerTransformer (método yeo-johnson) a los conjuntos de train y test para normalizar la distribución de las características
2. **Configuración del Modelo:** Se instanció un RandomForestRegressor con los siguientes hiper parámetros específicos:
    - n_estimators: 1183
    - max_depth: 16
    - max_features: 1.0
    - bootstrap: True
    - min_samples_split: 2
    - min_samples_leaf: 2
    - max_samples: 0.8654

Selección de Características

Utilizando el modelo de random forest, se extrajo la importancia de cada característica para identificar a los predictores más influyentes.

Se seleccionaron las **20 características principales**. Las más relevantes fueron:

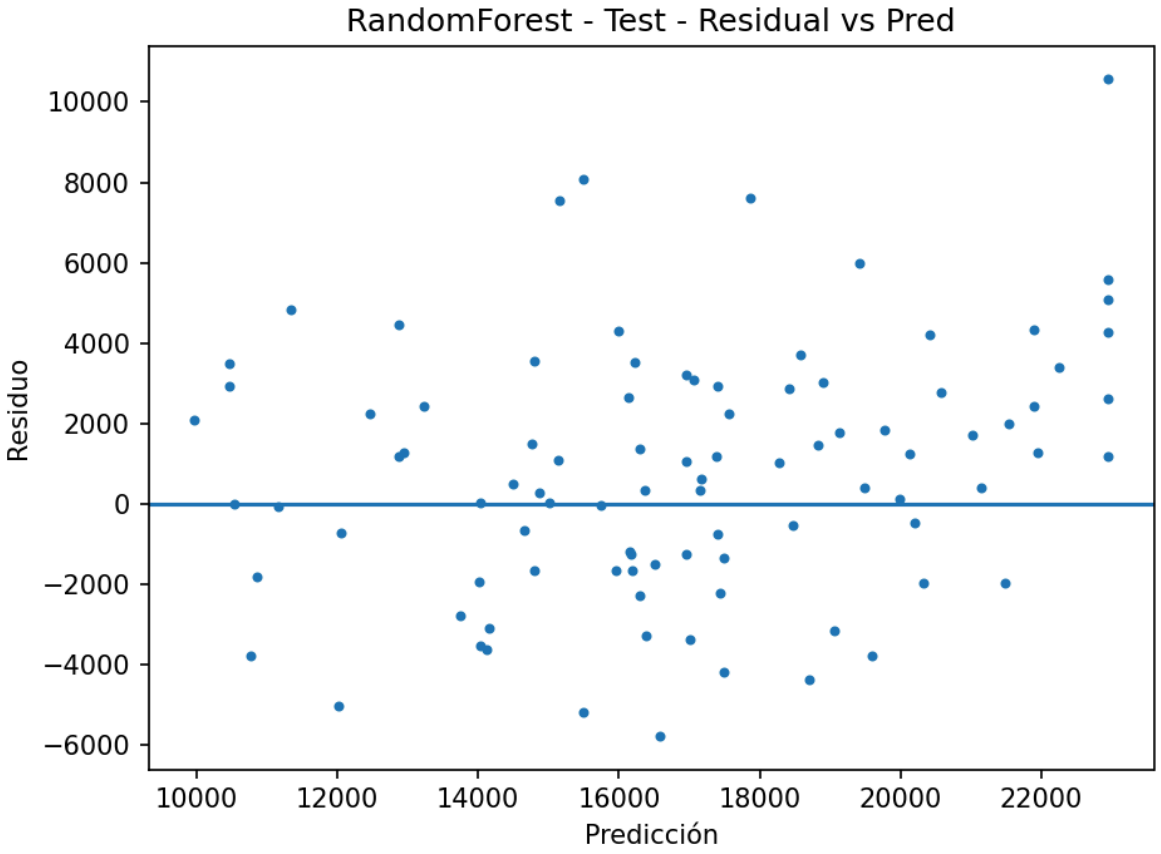


Teniendo en cuenta que la variable Frio(Kw) representa el consumo de frío del día de hoy.

Experimentación con mejores features

Para validar la selección de características y comparar arquitecturas de modelos, se ejecutó el modelo usando sólo el subconjunto de 20 características.

1. **Transformación y Partición:** Los datos fueron nuevamente transformados y divididos en conjuntos de entrenamiento (80%) y validación (20%).
2. **Modelos Comparados:**
    - **RandomForest**



- **XG Boost:** Configuración fija con early_stopping.
- **LightGBM:** Configuración fija con early_stopping
- **Ridge (Lineal):** Hiper Parámetro alpha optimizado con Optuna sobre CV temporal.
- **Lasso (Lineal):** Hiper Parámetro alpha optimizado con Optuna sobre CV temporal.

Resultados

Este proceso de optimización múltiple y selección de características culminó en la identificación del RandomForestRegressor como el modelo de mejor rendimiento.

El modelo fue primero testeado con datos extraídos del dataset provisto por la cátedra, arrojando resultados favorables:

MAE: 2,629

R2=0.66

Luego fue testeado “en vivo” por la cátedra por un nuevo conjunto de testeo jamás visto por nosotros. Aquí, el pipeline se encargó de realizar el preprocesamiento tal como fue diseñado, arrojando resultados (MAE y R2) similares a los del test anterior

Conclusiones

A partir de los resultados obtenidos podemos concluir que el RandomForestRegressor, para los datos tomados y preprocesados de la manera realizada, es el mejor modelo para predecir la cantidad de Kilowatts utilizados para enfriar la sección seleccionada de la planta. Con lo mencionado, podemos afirmar que a medida que la escala de datos con los que se trabaja (tanto features como observaciones) aumenta se debe tener cuidado y ser minucioso con que métodos se elige para escalar datos, imputar valores nulos, tratar outliers, entre otras cosas tanto así como en la comparación y elección del modelo predictor. Esto, ya que puede influir directamente en el desempeño del modelo, llevándonos a tener mejores o peores resultados y por ende pudiendo dar mayor escalabilidad e importancia a nuestro producto.